

Inaugural-Dissertation

zur

Erlangung der Doktorwürde

der

**Gesamtfakultät für Mathematik, Ingenieur- und
Naturwissenschaften**

der

Ruprecht-Karls-Universität Heidelberg

vorgelegt von

Michael Anton Baumgartner, M. Sc.

aus Euskirchen, Deutschland

Tag der mündlichen Prüfung: _____

Generalised Medical Object Detection via Self-Configuring Method Design

Erstbetreuer: Prof. Dr. Klaus H. Maier-Hein
Zweitbetreuer: Prof. Dr. Philipp Vollmuth

Abstract

The rich information in medical images fuels the need for an increasing number of acquisitions, ultimately resulting in a high workload for clinicians. Finding relevant structures in these images resembles a needle in the haystack problem, and when conducted manually, it is an error-prone and time-consuming process. Computer Aided Diagnosis (CAD) tools offer an alternative and can help to alleviate the current burden by speeding up clinical workflows and assisting with a second opinion. Diagnostic tasks, building the backbone of daily clinical routines, require the fast and accurate identification of critical structures like (1) vessel occlusions, which can potentially cause an Acute Ischemic Stroke, the second leading cause of death worldwide or (2) lung cancer manifesting as spherical structures, the leading cause for cancer-related death. Current work on medical image analysis predominantly focuses on semantic segmentation, which has shown great success for precise voxel-wise delineation of targets. However, diagnostic decision-making requires the correct localisation and classification of objects rather than voxels, which can not be accurately captured by semantic segmentation. Object detection methods can learn to identify objects in an end-to-end fashion, providing great utility by directly solving diagnostic tasks. Adoption in the domain is hampered by limited experience with these methods and complex configuration of application-specific parameters. This thesis explores the wide range of medical detection tasks and builds the foundation for future work in this important domain.

Three studies are presented to gain insights into important configuration choices of detection methods and highlight their versatility and competitiveness. The first case study revolves around an international challenge to tackle the detection of mediastinal lesions in Computed Tomography (CT) images. Despite its clinical relevance, no public benchmark was available to develop suitable methods for this anatomical region. Our

solution based on a one-stage anchor-based detection model achieves an excellent Free-response Receiver Operating Characteristic (FROC) score of 0.9897, resulting in near-optimal sensitivity for this task. The submitted method achieved the third rank in the challenge, underlining the competitiveness of detection methods for diagnostic tasks.

The second study explores the quick and reliable identification of vessel occlusions in Computed Tomography Angiography (CTA) images. Instead of relying on hand-crafted heuristics and extensive pre-processing schemes that limit the applicability of current solutions, our method can detect an arbitrary number of occlusions without restrictions on certain vessels. Our study includes three cohorts, two of which were collected in a pseudo-prospective manner from external hospitals to evaluate the real-world impact of our method. The proposed method achieves high sensitivity ($\geq 81\%$) and negative predictive value ($\geq 93\%$) on these cohorts, highlighting its clinical utility for identifying patients at risk. Qualitative inspection revealed the ability to find High-grade Stenosis (HGS), which were not labelled within the training cohort but constitute clinically relevant findings. We compared our method against two commercially available CE-marked and FDA-approved software solutions and demonstrated significant improvements over these, especially for the difficult to detect Medium Vessel Occlusions (MeVOs). Our solution is publicly available via a web platform: <https://stroke.ccibonn.ai>.

Thirdly, the feasibility of different detection models for the medical domain is explored. Detection Transformer models do not rely on additional proxy formulations with prior anchor boxes and offer direct set prediction capabilities, bypassing the requirement for manual heuristics during training and inference. Our study explores the utility of these models for diagnostic tasks by comparing the performance of three direct set prediction models with varying complexity against a strong anchor-based detection baseline. Two simpler designs, using single-scale information, are not able to compete with anchor-based approaches while the more complex model, using multi-scale deformable attention, performs on par with or better than the baseline.

Based on a newly established development pool consisting of ten data sets and equipped with the experiences from the initial three case studies, we developed the first generalising detection method, nnDetection. Following nnU-Net’s design principles, we systematise the configuration process of medical detection methods by identifying rule-based, fixed and empirical design choices. It distils the knowledge from hundreds of experiments and several years of experience into a self-configuring method design. We build a unified framework to incorporate a heterogeneous set of object detection models based on single-stage, two-stage and direct-set prediction designs. To offer the best possible utility of our method, models with box-level and a combination of box- and voxel-level supervision are incorporated to handle diverse annotation types. We evaluate our method on nine previously unseen detection tasks, introducing new modalities, anatomical regions and object structures. nnDetection outperforms two baselines and five ablation models on this diverse pool of tasks. Additionally, we compare the generalising design of our method on

three benchmarking data sets against current task-specific detection solutions and show that nnDetection achieves state-of-the-art results. Our work establishes a standardised baseline and easy entry point in the detection domain to catalyse future research. It democratises the availability of volumetric detection methods by offering out-of-the-box applicability to new data sets without requiring expert knowledge.

In summary, the work in this thesis has the potential to revolutionise the field of medical object detection by establishing a new development paradigm aimed at designing a generalising method. Our experiments on manually configuring detection methods demonstrate the utility and superiority of the proposed approaches over existing solutions. By distilling our findings into a self-configuring method, we make our knowledge available to the entire community and build the foundation for the next generation of medical detection methods. We have already leveraged the capabilities of nnDetection to compete in several international challenges with great success: ADAM 2020 (first rank detection track), MELA 2022 (third rank), TDSC-ABUS 2023 (second rank detection track) and INSTED 2024 (first rank). The code release of a preliminary version of nnDetection has already attracted a lot of interest in the community and can be found under <https://github.com/MIC-DKFZ/nnDetection>.

Zusammenfassung

Die vielfältigen Informationen in medizinischen Bildern führen zu einem steigenden Bedarf an Aufnahmen und dadurch zu einer hohen Arbeitsbelastung für Kliniker:innen. Das Auffinden relevanter Strukturen in diesen Bildern ähnelt einem Nadel-im-Heuhaufen-Problem und ist bei manueller Durchführung fehleranfällig und zeitaufwändig. Computergestützte Verfahren bieten eine Alternative und können dazu beitragen, die aktuelle Belastung zu verringern, indem sie klinische Arbeitsabläufe beschleunigen und eine Zweitmeinung unterstützen. Diagnostische Aufgaben, die das Rückgrat des täglichen klinischen Arbeitsablaufs bilden, erfordern die schnelle und genaue Identifikation kritischer Strukturen wie (1) Gefäßverschlüsse, die potenziell einen akuten ischämischen Schlaganfall verursachen können, die weltweit zweithäufigste Todesursache oder (2) Lungenkrebs, der sich durch kugelförmige Strukturen manifestiert und die häufigste Todesursache im Zusammenhang mit Krebs darstellt. Aktuelle Arbeiten zur medizinischen Bildanalyse konzentrieren sich hauptsächlich auf die semantische Segmentierung, die sich als sehr erfolgreich für die präzise Voxel-weise Erfassung von Zielen erwiesen hat. Diagnostische Entscheidungen erfordern jedoch die korrekte Lokalisierung und Klassifizierung von Objekten anstelle von Voxeln, was von der semantischen Segmentierung nicht genau erfasst werden kann. Objekterkennungsmethoden können lernen, Objekte in einem ununterbrochenen (End-to-End)-Verfahren zu identifizieren und bieten einen großen Nutzen, indem sie diagnostische Aufgaben direkt lösen. Die Akzeptanz im Bereich wird durch begrenzte Erfahrungen mit diesen Methoden und die komplexe Konfiguration von anwendungsspezifischen Parametern verringert. Diese Arbeit untersucht das breite Spektrum medizinischer Erkennungsaufgaben und legt den Grundstein für zukünftige Arbeiten in diesem wichtigen Bereich.

Drei Studien werden vorgestellt, um Einblicke in wichtige Konfigurationsentscheidungen von Erkennungsmethoden zu gewinnen und ihre Vielseitigkeit und Wettbewerbsfähigkeit hervorzuheben. Die erste Fallstudie dreht sich um einen internationalen Wettbewerb zur Erkennung von mediastinalen Läsionen in Computertomographie (CT)-Bildern. Obwohl diese anatomische Region klinisch höchst relevant ist, stand kein öffentlicher Datensatz zur Verfügung, um geeignete Methoden für diese Region zu entwickeln. Unsere Lösung, die auf einem einstufigen, ankerbasierten Erkennungsmodell basiert, erzielt einen ausgezeichneten Free-response Receiver Operating Characteristic (FROC)-Wert von 0.9897, was zu einer nahezu optimalen Sensitivität für diese Aufgabe führt. Die eingereichte Methode erreichte den dritten Platz in dem Wettbewerb und unterstreicht die Wettbewerbsfähigkeit von Erkennungsmethoden für diagnostische Aufgaben.

Die zweite Studie untersucht die schnelle und zuverlässige Identifizierung von Gefäßverschlüssen in CT Angiographie Bildern (CTA). Anstatt auf handgefertigte Heuristiken und umfangreiche Vorverarbeitungsschemata zurückzugreifen, die die Anwendbarkeit aktueller Lösungen einschränken, kann unsere Methode eine beliebige Anzahl von Verschlüssen ohne Einschränkungen bestimmter Gefäße erkennen. Unsere Studie umfasst drei Kohorten, von denen zwei in einem pseudo-prospektiven Verfahren aus externen Krankenhäusern gesammelt wurden, um den realen Einfluss unserer Methode zu bewerten. Die vorgeschlagene Methode erreicht auf diesen Kohorten eine hohe Sensitivität ($\geq 81\%$) und einen hohen negativen prädiktiven Wert ($\geq 93\%$), was ihre klinische Nützlichkeit bei der Identifizierung gefährdeter Patienten unterstreicht. Die qualitative Inspektion zeigte die Fähigkeit, hochgradige Stenosen (HGS) zu finden, die nicht in der Trainingskohorte gekennzeichnet waren, aber klinisch relevante Befunde darstellen. Wir verglichen unsere Methode mit zwei kommerziell erhältlichen CE-gekennzeichneten und FDA-zugelassenen Softwarelösungen und zeigen signifikante Verbesserungen gegenüber diesen, insbesondere bei den schwer zu erkennenden Verschlüssen eines mittleren hirnversorgenden Gefäßes (MeVO). Unsere Lösung ist öffentlich über eine Webplattform verfügbar: <https://stroke.ccibonn.ai>.

Drittens wird die Machbarkeit verschiedener Erkennungsmodelle für den medizinischen Bereich untersucht. Detektions-Transformer-Modelle (DETR) stützen sich nicht auf zusätzliche indirekte Aufgabenformulierungen mit vorherigen Ankerboxen und können Mengen direkt erstellen, um die Anforderung an manuelle Heuristiken während des Trainings und der Inferenz zu umgehen. Unsere Studie untersucht die Nützlichkeit dieser Modelle für diagnostische Aufgaben, indem sie die Leistung von drei DETR-Modellen mit unterschiedlicher Komplexität mit einer starken ankerbasierten Methode vergleicht. Zwei simple Modelle, die Informationen einer einzelnen Auflösung verwenden, können nicht mit ankerbasierten Ansätzen konkurrieren, während das komplexere Modell, das mehrere Auflösungen verwendet, auf Augenhöhe oder besser als die Vergleichsmethode abschneidet.

Basierend auf einer neu etablierten Sammlung von 10 Entwicklungsdatensätzen und mit den Erfahrungen aus den ersten drei Fallstudien, haben wir die erste generalisierende Erkennungsmethode, nnDetection, entwickelt. Nach den Prinzipien von nnU-Net systematisieren wir den Konfigurationsprozess von medizinischen Erkennungsmethoden, indem wir regelbasierte, feste und empirische Entwicklungsentscheidungen identifizieren. Es destilliert das Wissen aus Hunderten von Experimenten und mehreren Jahren Erfahrung in eine selbstkonfigurierende Methode. Wir bauen ein einheitliches Framework auf, um mehrere heterogene Objekterkennungsmodellen aufzunehmen, die auf Ein-Stufen-, Zwei-Stufen- und Direkter-Mengen-Vorhersage basieren. Um die bestmögliche Nützlichkeit unserer Methode zu bieten, werden Modelle mit Box-Level- und einer Kombination aus Box- und Voxel-Level Information integriert, um verschiedene Arten von Annotationen zu handhaben. Wir evaluieren unsere Methode an neun zuvor nicht gesehenen Datensätzen, die neue Modalitäten, anatomische Regionen und Objektstrukturen einführen. nnDetection übertrifft zwei Vergleichsmodelle und fünf Ablationsmodelle in diesem vielfältigen Aufgabenspektrum. Darüber hinaus vergleichen wir das generalisierende Konzept unserer Methode an drei Vergleichsdatensätzen mit aktuellen aufgabenspezifischen Erkennungslösungen und zeigen, dass nnDetection kompetitive oder bessere Ergebnisse erzielt. Unsere Arbeit etabliert ein standardisiertes Werkzeug und einen einfachen Einstiegspunkt im Erkennungsbereich, um zukünftige Forschung zu katalysieren. Es erhöht die Verfügbarkeit volumetrischer Erkennungsmethoden, indem es ohne Expertenwissen auf neue Datensätze angewendet werden kann. Der Programmcode ist unter <https://github.com/MIC-DKFZ/nnDetection> verfügbar.

Acknowledgments

A special thanks goes out to my supervisor, Klaus Maier-Hein, for his consistent support of this thesis and for providing his invaluable guidance throughout the entire process. His forward-thinking vision for methodological research and insightful input enabled us to bring this work on medical object detection to such a wide community and demonstrate our capabilities during several international challenges. His influence extends far beyond this singular project, shaping the entire department by fostering a supportive environment and assembling a world-class team where ideas can thrive and grow.

Secondly, I would like to thank my clinical co-supervisor, Philipp Vollmuth, who provided very valuable input on our clinical collaboration project. He solved open questions quicker than anybody I have worked with, and I am grateful to have had the opportunity to work with him. Further gratitude goes out to Gianluca Brugnara and Edwin Scholze, who explained all the clinical details to me and put our work in perspective. By assembling a great multidisciplinary team, I truly believe we were able to shape future research on stroke diagnostics.

Thirdly, I would like to thank my iTAC, consisting of Paul Jaeger and Fabian Isensee. The excellent work and unique perspectives of you two had a major effect on my PhD, and even today, I am impressed by your ability to cut through all of the "fluff" and ask the right questions at the right time. It was always a pleasure to discuss current roadblocks with you and find new solutions.

Even though we didn't always agree on a comfortable temperature in the room, a warm thank you goes out to the entire fridge: Fabian Isensee, Paul Jaeger, Tassilo Wald, Constantin Ulrich, Gregor Koehler, Maximilian Fischer, Jens Petersen, and Jakob Wasserthal. Everybody in there was so dedicated to their work, and it was super inspirational to sit

next to you and work alongside you. It was very motivating to brainstorm with such talented friends, and I really enjoyed being part of such an amazing work incubator.

Unfortunately, there is not enough space to thank everybody I had the chance to work with within MIC, but it was amazing to be a part of your team. I will certainly miss the time with the second-floor crew (who might or might not be working right now) and the friendly competition with the HIP office. The canoe tours were absolutely fantastic, and I hope the (boat) piracy will continue in the future. I would also like to thank "Group 4" for our regular meetings and exchanges. It was great to summarize thoughts at regular intervals and present them to you to hear your perspectives on them. Furthermore, I would like to thank Marc Ickler, who wrote his Master's thesis with MIC and worked closely with me on detection topics.

I would like to thank everybody who read parts of my thesis and provided valuable feedback and corrections: Laura Lokowandt, Maximilian Zenk, Balint Kovacs, Jessica Kächele, Andrés Martínez Mora, Katharina Eckstein, and Partha Ghosh.

While I spent the majority of my PhD in the fridge, it all started in a different office with Maximilian Zenk, Jan Sellner, and (technically not part of the office) Silvia Seidlitz. I guess this thesis also contradicts our magical motto: "So wird das Nichts mit dem PhD." Every time I see a message in our group chat, I smile, and I look forward to spending many more days with all of you.

Last but not least, I would like to thank my parents, Nora and Anton Baumgartner, who supported me throughout my entire PhD. It was not easy all of the time, but you always believed in me and had my back. You always encouraged me to stay curious, approach challenges in a structured way, and keep going, even when it was difficult. Thank you for supporting me through all this time—this wouldn't have been possible without you. A special place in my heart is reserved for Laura Lokowandt, who supported me through the second part of my PhD and is a major inspiration. I am very grateful that you have entered my life and stuck around even when I needed to work late hours.

Heidelberg, February 2025
Michael Baumgartner

Contents

Abstract	v
Zusammenfassung	ix
Acknowledgments	xiii
1 Introduction	1
1.1 Motivation	2
1.2 Research Questions	6
1.2.1 RQ1: Which methodological design decisions are important for good object detectors?	6
1.2.2 RQ2: How can object detection methods be configured automati- cally?	10
1.3 Outline	14
2 Fundamentals	17
2.1 Object Detection	17
2.1.1 Two-Stage Object Detection	17
2.1.2 One-Stage Object Detection	24
2.1.3 Direct Set Prediction for Object Detection	32
2.1.4 Other Detection Approaches	39
2.2 Evaluation	39
2.2.1 Localization Criterion	40
2.2.2 Matching	41
2.2.3 Counting Metrics	41

2.2.4	Ranking Metrics	42
2.2.5	Patient-Level Evaluation	43
3	Related Work	45
3.1	Task Specific Design of Medical Object Detection Models	45
3.1.1	Detecting Mediastinal Lesions	45
3.1.2	Vessel Occlusion Detection	46
3.1.3	Aneurysm Detection	48
3.1.4	Lung Nodule Detection	50
3.1.5	Detection with Detection Transformers in the Medical Domain .	54
3.2	Self-configuring Model Design for Medical Applications	55
4	Materials and Methods	57
4.1	Task Specific Design of Object Detection Methods	58
4.1.1	Detecting Mediastinal Lesions in CT Images	58
4.1.2	Detecting Vessel Occlusions in CTA Images	63
4.1.3	Exploring Detection Transformers for Medical Object Detection .	67
4.2	Self-Configuring Design of Medical Object Detection Methods	73
4.2.1	Data Set Analysis	73
4.2.2	Development Process	77
4.2.3	Generalisation Process	77
4.2.4	Data Fingerprint	78
4.2.5	Rule-based Parameters	79
4.2.6	Fixed Parameters	83
4.2.7	Empirical Parameters	96
4.2.8	nnU-Net Baselines	97
5	Experiments and Results	99
5.1	Task Specific Design of Object Detection Methods	100
5.1.1	Detecting Mediastinal Lesions in CT Images	100
5.1.2	Detecting Vessel Occlusions in CTA Images	102
5.1.3	Exploring Detection Transformers for Medical Object Detection .	110
5.2	Self-Configuring Design of Medical Object Detection Methods	114
5.2.1	Experimental Setup and Evaluation	114
5.2.2	Benchmarking on the Generalisation Pool	115
5.2.3	Benchmarking against Task-specific Methods	121
6	Discussion	125
6.1	Task Specific Design of Object Detection Methods	125
6.1.1	Detecting Mediastinal Lesions in CT Images	125
6.1.2	Detecting Vessel Occlusions in CTA Images	126
6.1.3	Exploring Detection Transformers for Medical Object Detection .	129

6.2	Self-Configuring Design of Medical Object Detection Methods	131
7	Conclusion	135
7.1	Summary	135
7.1.1	Manual Design of Detection Methods	136
7.1.2	Self-Configuring Design of Detection Methods	137
7.2	Outlook	138
7.3	Closing	140
A	Own Contributions and Publications	141
A.1	Own Contributions	141
A.2	Own Publications	142
B	Additional Results	147
B.1	Detecting Vessel Occlusions in CTA Images	147
B.2	Exploring Detection Transformers for Medical Object Detection	151
B.3	Self-Configuring Design of Medical Object Detection Methods	151
B.3.1	Additional Results	151
B.3.2	Data Set Results	154
B.3.3	Data Set Information	172
	List of Acronyms	195
	List of Figures	199
	List of Tables	203
	List of Algorithms	207
	Bibliography	209

CHAPTER 1

Introduction

Human vision is one of the five senses that help us navigate the world, spot hazards from a far distance and appreciate the beauty of our surroundings. Due to its central role in our daily lives, researchers have long attempted to develop systems that can capture images to replicate this human ability with the help of cameras [1] and computers [2]. Humans are able to quickly learn to recognize objects from a young age [3] but machines are still struggling to achieve this goal [2]. 'Moravec's paradox' [4] describes this observation in robotics and artificial intelligence, where higher-level reasoning can be modelled easily by machines, but replicating basic skills requires enormous efforts and computational resources. The previously described task of localising and classifying objects in images is called *object detection* in computer vision.

Natural images aim to replicate the capabilities of the human eye which captures information in the visible spectrum. However, the electromagnetic spectrum is much wider, and more advanced imaging techniques can be used to capture even more information about surfaces and retrieve information within objects without requiring invasive procedures [5]. This type of imaging has tremendous potential in the medical field and falls within the radiology domain. Information retrieval is not just limited to two-dimensional (2D) information, but medical systems can retrieve Computed Tomography (CT) scans or Magnetic Resonance Images (MRIs), which contain volumetric (or three-dimensional, 3D) information [5]. These images contain essential information about soft tissues, bones, ligaments, tendons, blood vessels, and organs, which expert personnel, like radiologists, can interpret to derive accurate diagnoses, quantify the extent of injuries or diseases, and create treatment plans.

Due to the immense potential of these images, many clinically relevant procedures have integrated them as a central part of their routine:

- **Surgical Planning:** By obtaining a first view of the anatomy and location of relevant structures, clinicians can create a detailed plan for upcoming surgeries, which results in shorter intervention times and reduced risk of follow-up complications [6].
- **Minimal Invasive Surgery/Intervention:** This type of interventional procedure only requires a small cut which reduces blood loss, pain, risk of infection and duration of patient stays in the hospital. Examples of these include closures of holes in the heart [7] and thrombectomy to remove blood clots from arteries [8].
- **Radiotherapy:** This procedure destroys cancer cells via ionizing radiation. Medical imaging enables the acquisition of fine-grained scans to create a plan for the radiation dose in order to maximize its effectiveness and minimize the impact on healthy tissue [9, 10].
- **Diagnostic Imaging:** Many diseases manifest as connected structures within the human body and can be visually identified in medical images. Timely diagnosis of these is essential to maximize survival chances, for example, during cancer diagnostics to avoid metastases or in case of an acute ischemic stroke where it is critical to restore the blood flow to the brain.

This thesis focuses on diagnostic imaging, which builds the foundation of clinical decision-making. Its use cases extend across a vast array of image modalities, anatomical regions and object structures as depicted in Figure 1.1 with examples for rib fractures in CT images [11, 12], lung nodules in CT images [13, 14], kidney tumours in CT images [15], pancreas tumours in CT images [16], cerebral aneurysms in Rotational X-Ray Angiographies (3DRA) and MRI [17, 18], mediastinal lesions in CT images [19], microbleeds in MRI [20, 21, 22, 23, 24] and breast lesions in contrast-enhanced MRI [25, 26].

1.1 Motivation

The broad range of applications and utility of medical imaging leads to a constantly increasing number of acquisitions that need to be inspected by expert personnel, such as radiologists. A study by Smith-Bindman et al. (2008) included data from a large health plan between 1997 and 2006, reporting an increase in CT acquisitions by a factor of two and an increase in MRI acquisitions by a factor of three [27]. Further evidence for this phenomenon can be obtained by the annual statistical release of the NHS England, which shows the continued increase in CT and MRI acquisitions over recent years [28] in England. In 2013/14, 3.8 million CT images were acquired, almost doubling by 2021/22 with close to 6.8 million acquisitions. A similar pattern can be observed for MRI scans,

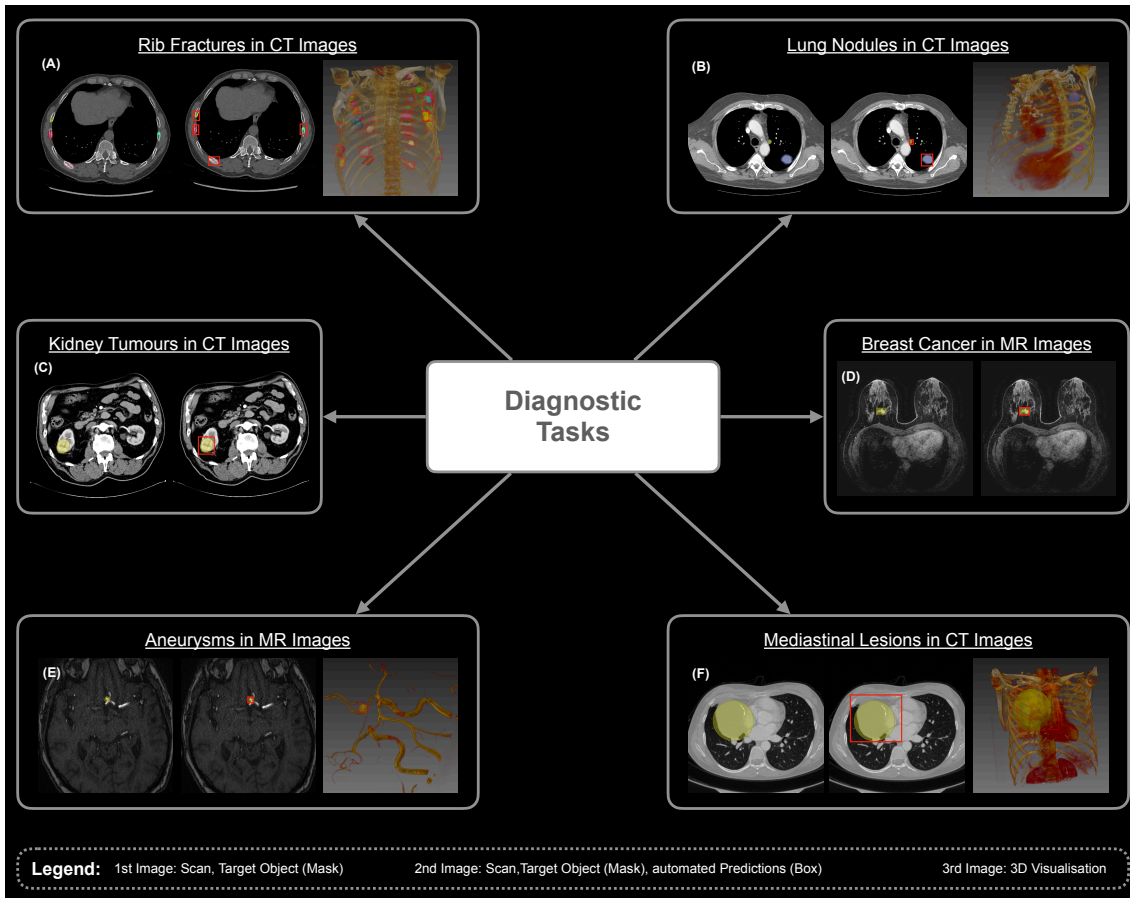


Figure 1.1: Overview of medical image modalities and object structures requiring diagnostic decision making. Diagnostic tasks can be found for various medical image modalities to capture different tissue properties, anatomical regions and object structures. A subset of all possible clinical tasks is presented in this visualisation, including (A) rib fractures in CT images [11, 12] (B) lung nodules in CT images [13] (C) kidney tumours in CT images [15] (D) breast cancer in contrast-enhanced MR images [25, 26] (E) cerebral aneurysms in 3DRA images [17] (F) mediastinal lesions in CT images [19].

with an increase from 2.6 million in 2013/14 to 3.8 million acquisitions in 2021/22. This increase in volumetric imaging data puts an ever-increasing workload on radiologists, which impacts the quality of care and therefore the patients health.

Computer Aided Diagnosis (CAD) systems aim to support clinicians by providing a timely second opinion and automating repetitive parts of their workflow. The support system can provide information on different abstraction levels, directly depending on the desired clinical question. An overview of these levels and an exemplary use case for rib fracture diagnosis is depicted in Figure 1.2 and explained in the following:

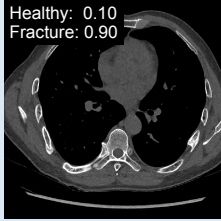
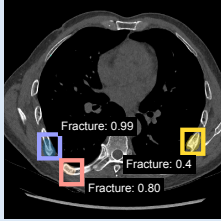
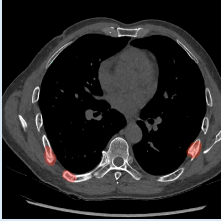

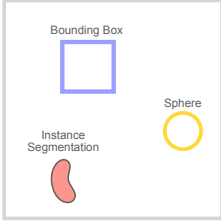
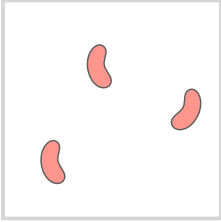
Support Level	Image-Level	Object-Level	Voxel-Level
Clinical Goal	Does the patient need to undergo follow up treatment?	Where are the fractures located? How many fractures are present?	What is the exact shape and volume of the fractures?
Typical Evaluation Metrics	Receiver Operating Characteristic Sensitivity Specificity	Mean Average Precision Free-response Receiver Operating Characteristic	Dice Similarity Coefficient Normalised Surface Distance
Method Level	Image Classification 	Object Detection 	Semantic Segmentation 
Possible Annotation Types			
Aggregation	/	Diagnostic chain: object presence induces patient follow up	Manual heuristics: clustering of voxels, aggregation of confidence scores

Figure 1.2: Different levels of abstraction for Computer Aided Diagnosis support. Automated systems can provide information on different abstraction levels, ranging from the image level down to the voxel level. The granularity of detail in the output influences their suitability to answer certain clinical questions and requires additional aggregation techniques when deriving insights about more abstract levels. The evaluation metrics directly operate on the selected granularity level to measure the performance of the respective model.

- **Image-Level:** The highest level of decision support is derived at the **image-level** (or **patient-level** in this thesis) where the CAD system provides a single numerical value indicating a class or confidence score for the entire image. In this case, no further information is provided, and the location of clinically relevant regions

remains unknown. Methodologically, these use cases can be directly modelled by image classification methods like ResNet [29, 30], DenseNet [31, 32], ConvNeXt [33], Vision Transformer (ViT) [34] which are trained with image-level labels. These labels can be obtained with the least amount of effort since image-level results are usually available within the Picture Archiving and Communication System (PACS) of hospitals, allowing for efficient large-scale collection efforts. On the other hand, methods that do not provide further information on their reasoning can often be seen as black boxes that are undesirable in clinical practice. Driven by the clinical needs, common evaluation metrics for these tasks include the Receiver Operating Curve (ROC), sensitivity and specificity.

- **Object-Level:** The next granularity level is represented by **object-level** information and object detection methods. These systems produce outputs that encompass individual objects, usually in the form of bounding boxes or instance segmentations, and can be trained end-to-end. This allows them to efficiently support diagnostic tasks which depend on the location of regions of interest or the number of objects in an image. Typical methods include Retina Net [35], R-CNN models [36, 37] and DETection TRansformer (DETR) models [38, 39]. By following the typical clinical diagnostic reasoning chain, patient-level results can easily be obtained from object-level results by aggregating the information. For example, if a patient presents multiple rib fractures in a CT image, the patient level result indicates necessary follow-up treatment. Evaluation metrics combine information about True Positive (TP), False Positive (FP) and False Negative (FN) predictions to quantify the ability to localise and classify objects. Metrics like mean Average Precision (mAP) and Free-response Receiver Operating Characteristic (FROC) measure performance in a confidence threshold agnostic manner, while sensitivity and precision can be used after deriving a confidence threshold for the method.
- **Voxel-Level:** The finest spatial granularity is presented on the **pixel- or voxel-level** with semantic segmentation methods, which can derive information about the shape and volume of regions of interest. These methods are usually based on an encoder-decoder architecture like a U-Net [40] or DeepLab [41] to predict the class correspondence per voxel. Different objects of the same class are not further differentiated in this type of use case. Evaluation is often performed with the Dice Similarity Coefficient (DSC), sometimes complemented with the Normalised Surface Distance (NSD) to assess the quality of the segmentation. Manual heuristics are needed to group the voxels into clusters and aggregate the confidence scores to derive object- or patient-level information.

Clinical decision making is ultimately performed on the patient level with grounded information from the object- or voxel-level. Due to its relevance for diagnostic decisions, this thesis focuses on developing CAD applications that can directly operate on the object-level. Most of the evaluation metrics are directly targeted towards measuring

the object-level performance of these methods and extend to the patient-level when evaluating the impact in clinical scenarios.

1.2 Research Questions

The medical image computing domain is predominantly working on the development of semantic segmentation methods as evidenced by the study of Maier-Hein et al. (2018), which analysed 150 biomedical image analysis challenges and found that 70% revolved around segmentation. Existing work on object detection is limited to a few medical applications like lung nodule detection [43, 14], aneurysm detection [44, 18], and breast cancer detection [45, 46]. However, possible clinical applications are much more far-reaching, and thus, it is necessary to explore the possibilities of these methods in the wider medical domain and establish powerful and robust detection methods.

1.2.1 RQ1: Which methodological design decisions are important for good object detectors?

The medical domain encompasses various tasks [47] with vastly varying properties like image modalities, anatomical regions, object structures and annotation types. Deep Learning (DL) is a powerful technology allowing the design of data-driven methods which can be applied to different medical tasks. The correct configuration of these methods requires iterative hyperparameter tuning based on a deep understanding of the detection model and profound knowledge of the clinical problem. This constitutes an error-prone process which often yields sub-optimal results and creates a bottleneck for current research. Until now, it has been unclear which methodological design decisions are essential for creating optimal detection models for varying medical tasks. As part of the first research question, this thesis presents three case studies focussing on the manual configuration of detection models:

RQ 1.1: Are detection methods competitive in international benchmarks?

International competitions, also called challenges, provide a platform to benchmark current state-of-the-art methods by comparing submissions from all over the globe with a standardised evaluation protocol on the same data. The Mediastinal Lesion Analysis (MELA) Challenge 2022 is one such benchmarking effort which attracted prestigious teams from all over the world. It provides the first publicly available data set for mediastinal lesions, establishing a good environment to develop highly performant detection methods. In our solution, we explore the design space of a single-stage anchor-based

detector named Retina U-Net [46]. Figure 1.3 shows three central aspects of our solution to enhance the performance of the proposed detection method: **(1) Adjust to target metric:** the localisation criterion [48] for this task is set to 0.3 which shifts the emphasis of the detection problem towards the precise localisation of the object rather than detecting its appearance **(2) Scale to available hardware:** available hardware resources can significantly influence the configuration of the model and making the best use of it helps to enhance the performance **(3) Model Ensembling:** combining the predictions from multiple models is a powerful technique to compensate for shortcomings of individual ones.

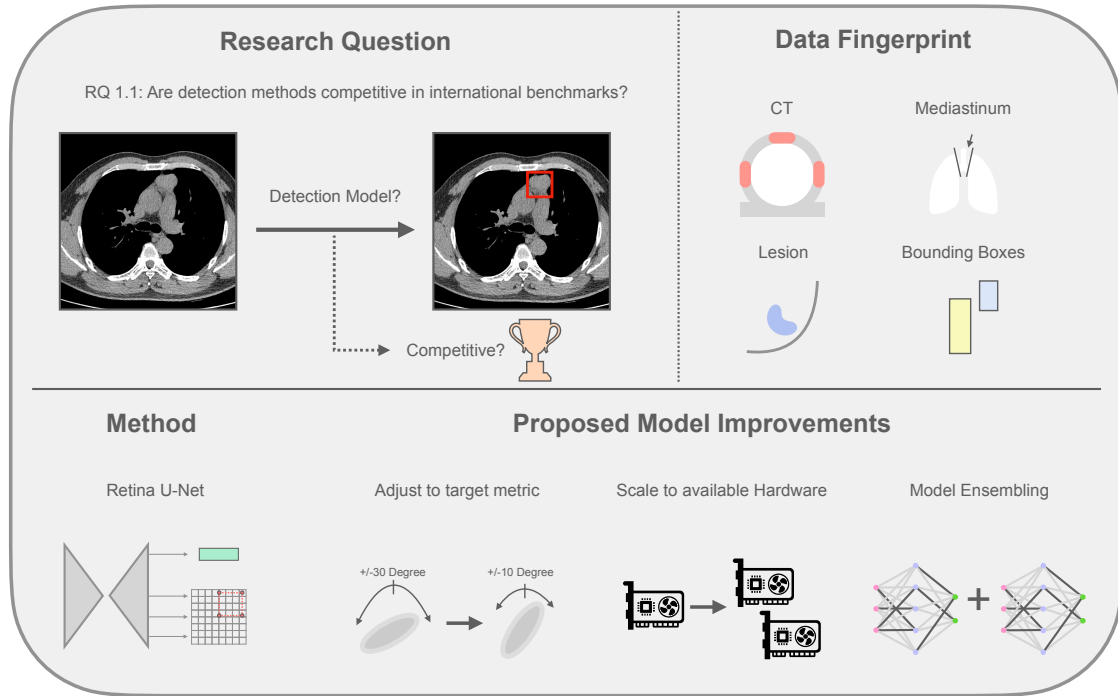


Figure 1.3: RQ1.1 - International Challenge: Design aspects of our submission to the MELA 2022 Challenge. Creating state-of-the-art detection models requires careful design of every component in the training and inference pipeline. Our solution for detecting mediastinal lesions in CT images focuses on the augmentation pipeline to reduce bounding box artefacts, scaling to available compute resources to avoid stitching artefacts during inference and model ensembling to combine strength from different models. The CT image is from the MELA data set [19]. The GPU icon was taken and adapted from [49].

RQ1.2: Can detection models offer a clinical value over existing solutions for vessel occlusion recognition?

The second-leading cause of death worldwide can be attributed to strokes [50]. It represents the third-leading cause of death and disability combined. One in four persons aged over 25 will experience a stroke during their lifetime [51]. Acute Ischemic Stroke (AIS) represents one possible cause for stroke where the blood flow to the brain is blocked. In the year 2019 alone, around 101.5 million people have suffered from a stroke, 77.2 million of which had AIS [52]. Blockage through vessel occlusions is a life-threatening condition which necessitates immediate intervention by expert personnel. Quick and reliable identification of vessel occlusions is therefore a highly important clinical task. Mechanical thrombectomy is a common treatment for accessible occlusion types but requires precise information about the occlusion's location. This information can be obtained from several different imaging modalities, namely Computed Tomography Angiography (CTA), Non-Contrast Computed Tomography (NCCT) and Computed Tomographic Perfusion (CTP). Automated detection systems can help prioritize patients in these scenarios by providing a fast and reliable assessment of the acquired images.

Designing detection models for clinical applications introduces additional requirements which do not occur when trying to solely maximize their performance. In this case, additional constraints on the inference time are imposed to ensure timely feedback to the attending clinician. It is necessary to select a cutoff value for the confidence scores and evaluate the performance on the patient level to mimic diagnostic decision support. We incorporate these requirements in the methodological design of our solution in three aspects: **(1) Adjust to annotation style:** vessel occlusions do not present boundaries like other pathologies but rather represent regions of interest which can be effectively detected by a model using box-level supervision, **(2) Adjust to inference time constraints:** the size of the occlusions only varies between two sizes, which can be effectively captured with two anchors to reduce the inference time of the model and **(3) Scale to available hardware:** by making use of the available computational resources, we design a multi-Graphics Processing Unit (GPU) inference scheme to efficiently process large CTA scans. An overview of the contributions is shown in Figure 1.4.

Additionally, an extensive analysis of our results on the patient-level and object-level, with and without cutoff selection, shows the potential impact on clinical decision making. We benchmark our method against two CE— and FDA-cleared software solutions and demonstrate its superiority on two external cohorts.

RQ1.3: Are direct set prediction models beneficial for medical object detection?

Convolutional Neural Networks (CNNs) are not capable of natively modelling a varying number of objects in an image. This has led to the development of different detection

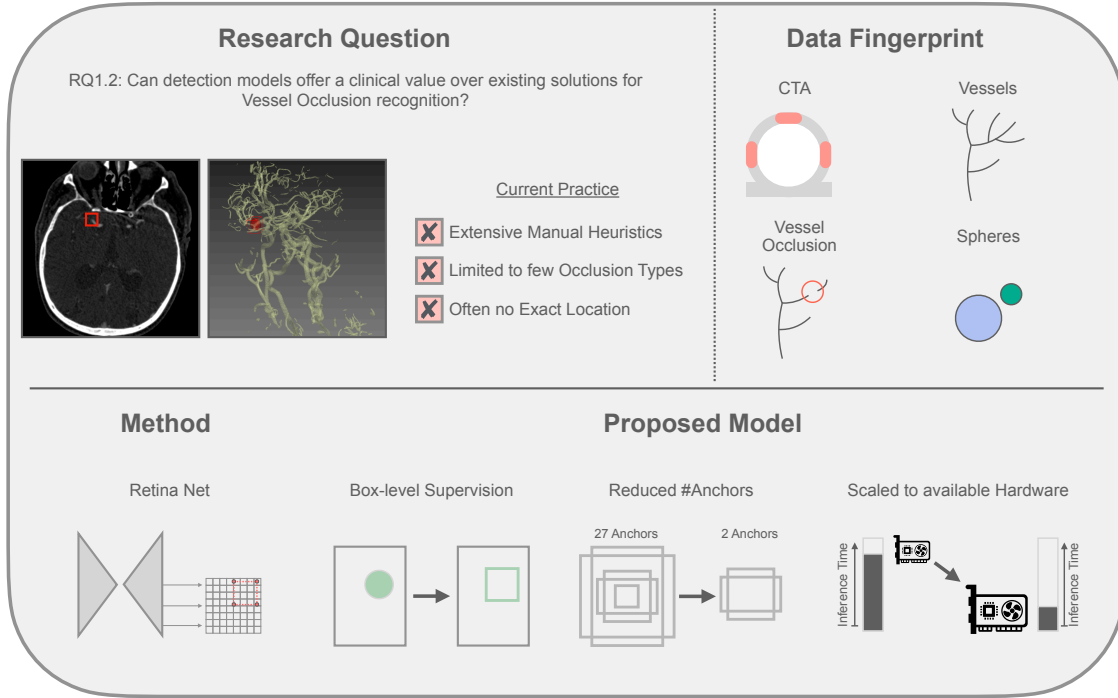


Figure 1.4: RQ1.2 - Clinical Application: Design aspects for developing a clinically relevant vessel occlusion detection model. Clinical scenarios pose different requirements on detection models than comparative or exploratory studies. In this study, we develop a detection model to identify vessel occlusions in CTA images. Since the identification of AIS is an extremely time-sensitive diagnostic task, we specifically tailor the model to be fast during inference. We achieve this by removing unnecessary components during the training, reducing the number of anchors to better represent the limited diversity of the annotated objects, and scaling our training and inference resources to the available hardware budget. The GPU icon was taken and adapted from [49].

formulations, like one-stage anchor-based [35], two-stage anchor-based [36, 37, 53, 54], centre-point based [55], extreme point-based [56, 57, 58] or direct set prediction [38, 59, 60, 61, 39, 62, 63] based detection models. Each of these detection formulations has advantages and disadvantages, which can make them more or less feasible for diagnostic tasks in the detection domain. Anchor-based detectors are used for multiple diagnostic applications, but direct set prediction models have not yet become widely adopted despite several advantages over anchor-based detectors in the natural image processing domain. To address this shortcoming, this research question investigates the feasibility of transformer-based direct set prediction models for medical object detection tasks, see Figure 1.5.

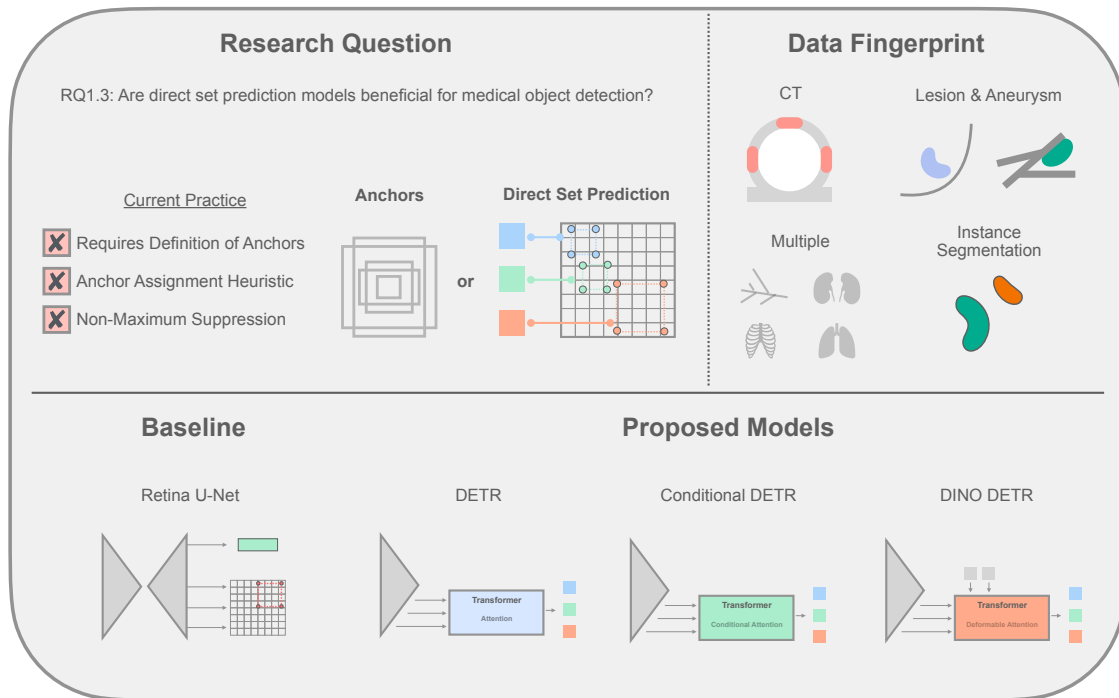


Figure 1.5: RQ1.3 - Overview of anchor-based and direct set prediction models. Anchor-based detectors require careful configuration of anchor properties and hand-designed heuristics for training and inference. Three transformer-based direct set prediction models are evaluated using four medical data sets with varying anatomical regions and object structures.

1.2.2 RQ2: How can object detection methods be configured automatically?

As outlined in RQ1, every medical task has a unique combination of available image modalities, anatomical regions, object structures and annotation types. Most developers expect that data-driven approaches, like deep learning models, can learn these intricate characteristics from the underlying data distribution without encoding additional information. In reality, each medical task requires careful configuration to achieve optimal results, constituting an active research bottleneck due to sub-optimal baseline performance, limited generalisation of methods, and high time and resource demands in the domain. This is caused by the complex design of current methods, which depend on the correct selection of loss functions, patch sizes, architectures, supervision signals and model-specific parameters like anchors.

Current work focusses on designing task-specific solutions [64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 11, 12] which are tailored toward a single data set. Determined configurations might work well for the current task but potentially degrade for other

tasks. As a consequence, many methods fall short of their promises, which hampers clinical adoption and impacts patient care. A recent study has analysed these pitfalls in the semantic segmentation domain [81].

nnU-Net [47] was the first method to systematise the configuration process of semantic segmentation models and move away from the prevailing development paradigm towards designing models generalising across many medical tasks. It introduces a self-configuring method which is based on three groups of parameters: *rule-based* parameters are adapted based on properties of the underlying task, *fixed parameters* remain constant across tasks, and *empirical parameters* are empirically optimised for each task based on cross-validation results. However, detection models depend on a different set of hyperparameters than segmentation models, necessitating a different internal design. Anchor-based models require the definition of anchor sizes and density, which are absent from other domains. Direct set prediction models have a strict upper limit of the possible number of predictions and variation of this parameter leads to different training dynamics and inference performance. All detection models require careful design of their architectural blueprint, data loading strategy and loss functions. Additional layers of complexity specific to the detection domain are added on top of this:

- **Detector Types:** Current studies only focus on the design of individual detector types, making it difficult to obtain a broader view across detector models. It remains unclear if a single model can adapt to all medical detection tasks without requiring changes to its architecture.
- **Annotation Types:** Diagnostic tasks encapsulate a large diversity of object structures, which necessitates the usage of different annotation types. Voxel-wise segmentation offers the richest signals but suffers from high annotation costs, restricting the annotation of large data sets. Coarser annotation styles, like bounding boxes or spheres, can accelerate the annotation process and allow the annotation of regions without clear structural boundaries.
- **Metrics:** The clinical use cases can differ between tasks, which necessitates the usage of different localisation-quality thresholds and metrics. These variations in the evaluation are needed to capture multiple aspects of a method. A thorough performance analysis is not limited to a single metric but aims to capture the full bandwidth of applications to provide an adequate view of the model's performance across different requirements and tasks [48].

To make detection models available to the entire medical community, including both domain experts and Machine Learning (ML) practitioners, this thesis develops nnDetection, the first self-configuring medical object detection method. Two major research questions are tackled in this work: **(RQ 2.1) Is there enough data for developing general detection methods?** and **(RQ 2.2) How can the design of detection methods be automated?** The proposed development process encompasses 10 different detection

data sets with varying tasks, and the evaluation is performed on 9 additional data sets, which include previously unseen image modalities, anatomical regions, object structures and annotation types. By unifying the design of single-stage, two-stage, and direct-set prediction models in a single framework, it can leverage the benefits of each model as part of an ensemble. nnDetection automates the configuration process of all integrated models and can be applied with no user intervention to any 3D detection problem, see Figure 1.6.

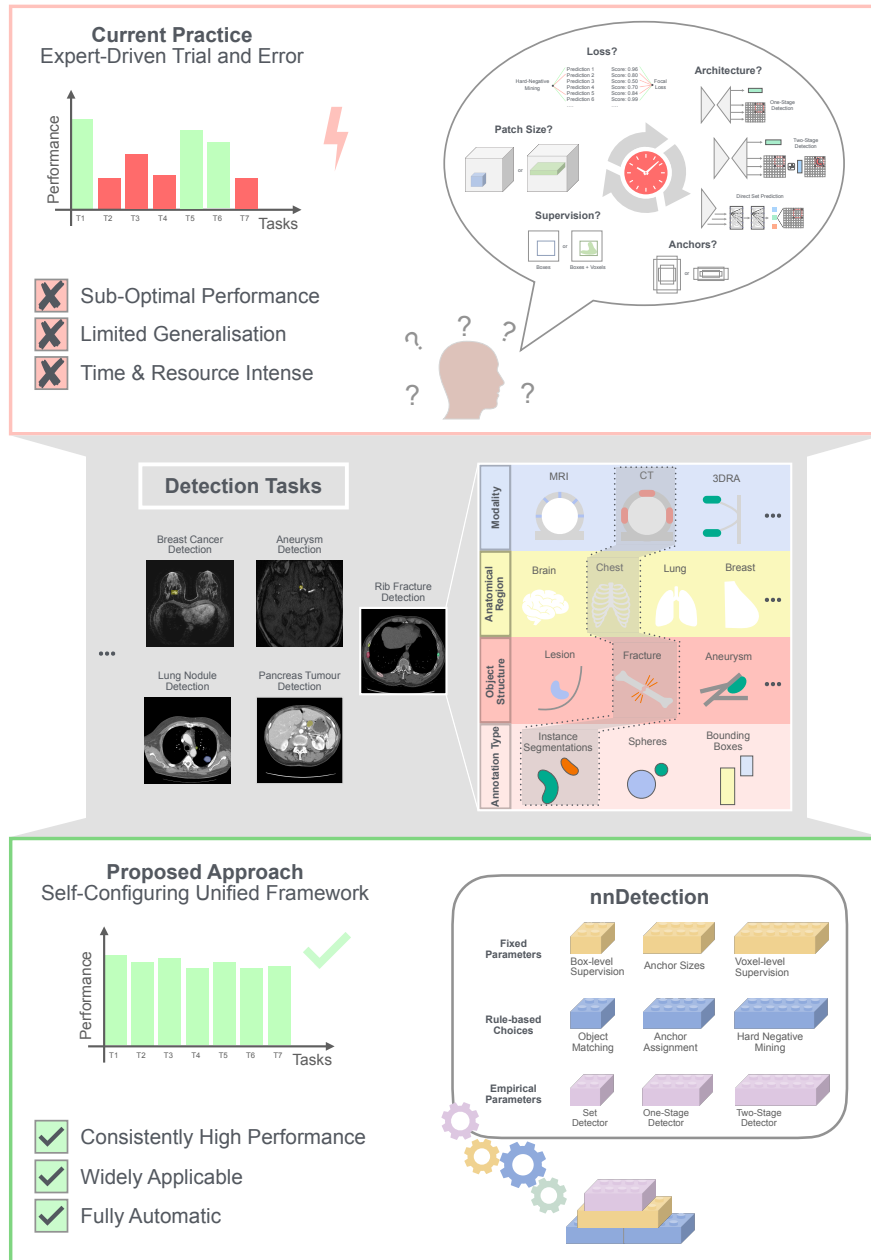


Figure 1.6: RQ 2: Overview of current task-specific design versus proposed self-configuring method design for detection models. Shows the gap between task-specific design (current) and self-configuring (proposed) method design. Deep learning-based detection models require correct configuration to achieve the best possible performance. The configuration process must be repeated for each new task due to varying characteristics. Self-configuring methods, like nnDetection, automate the entire hyperparameter tuning process and provide robust performance across detection tasks without manual intervention. This figure is adapted from [82].

1.3 Outline

This thesis comprises seven chapters in total. The first chapter introduces medical imaging and motivates the need for computer-aided diagnostic tools in the form of detection models.

Chapter 2 explains the fundamentals of object detection. First, the internal mechanics of two-stage detection models, which represent the previous state-of-the-art in the natural image processing domain, are explained. Afterwards, one-stage detection models are introduced as a simpler alternative while achieving competitive results. Direct set prediction models that use the transformer architecture, representing the current state-of-the-art, are described last. mAP and FROC are explained as central metrics in the medical detection domain.

Chapter 3 highlights relevant literature about medical object detection. The first part discusses models which are manually configured to solve individual medical tasks. The second part introduces the concept of self-configuring models represented by nnU-Net.

Our methodological work for different detection tasks is explained in Chapter 4. It first highlights the development of three detection models: (1) a model for mediastinal lesion detection in the context of an international challenge, (2) developing a detection model for vessel occlusions in CTA images with multiple clinically relevant data sets and (3) exploring direct set prediction models for medical tasks. The second part showcases our work on nnDetection, the first self-configuring medical object detection method. It provides detailed descriptions of each component to automatically configure one-stage, two-stage, and direct set prediction models.

Chapter 5 provides an overview of our experiments and initial analysis. The impact of individual design decisions is showcased based on our submission to the MELA challenge, where it ranked third on the test leaderboard. Detailed analysis of the developed vessel occlusion model on two external pseudo-prospective cohorts is provided, showcasing the superiority of our approach when compared to two CE- and FDA-approved software solutions. Experiments on four data sets demonstrate the feasibility of direct set prediction models in the medical detection domain, outperforming previous state-of-the-art anchor-based detection models.

The second part of Chapter 5 evaluates nnDetection across ten development, nine generalisation and three benchmarking data sets. The generalisation pool introduces previously unseen image modalities, anatomical regions, object structures and annotation types. The benchmarking pool allows for the comparison against cutting-edge task-specific models and showcases that nnDetection is not just able to compete against these methods but also achieves new state-of-the-art performance on the PN9 [14] and CTA-A [83] data sets.

Chapter 6 discusses our findings in the broader scope of the medical detection domain by showing its limitations and impact on current work.

A summary of our work in the context of the previously introduced research questions and future research directions is provided in Chapter 7. An overview of the corresponding sections for the research questions is shown in Table 1.1.

Table 1.1: Overview of research questions in this thesis.

Research Questions	Related Work	Methods	Results	Discussion
RQ 1.1	Section 3.1.1	Section 4.1.1	Section 5.1.1	Section 6.1.1
RQ 1.2	Section 3.1.2	Section 4.1.2	Section 5.1.2	Section 6.1.2
RQ 1.3	Section 3.1.5	Section 4.1.3	Section 5.1.3	Section 6.1.3
RQ 2	Section 3.1.4	Section 4.2	Section 5.2	Section 6.2
	Section 3.2			

Disclosure of Contributions

The presented data sets, experiments, and discussions are part of a multidisciplinary effort that involves several clinical and technical expert-level collaborators. To reflect these collaborative efforts, this thesis uses the “we” form rather than the “I” form to reflect these efforts. Chapter A contains detailed information on my contributions and publications.

CHAPTER 2

Fundamentals

Assigning spatial and categorical information to objects in images is a fundamental task of computer vision algorithms. The granularity of the predicted spatial information can vary between applications; delineation by bounding boxes is commonly referred to as object detection [84, 85, 86, 87] and pixel-wise delineation is referred to as instance segmentation [88, 85, 89]. Early works on object detection rely on a sliding window scheme in combination with manual feature descriptors like Viola-Jones [90] or histogram of oriented gradients (HOG) [91] to classify blocks in an image. After the huge success of CNN based approaches, in the form of AlexNet [92] and VGG [93], in the ImageNet competition [94] research started exploring possibilities to model the object detection task with deep learning. Since the vast majority of today's detection systems are based on state-of-the-art neural networks, the following sections will explain different design philosophies of deep learning-based object detection approaches.

2.1 Object Detection

2.1.1 Two-Stage Object Detection

All two-stage detection methods generate their predictions by first creating a large number of, usually class-agnostic, region proposals, followed by a second classification and regression module to refine the initial region proposals and assign the final class and

confidence score to them. One of the earliest methods to leverage neural networks for object detection is the Regions with CNN features (R-CNN) [95] framework, a commonly used two-stage detector design which will be explained in the following sections.

Regions with CNN features (R-CNN)

The introduction of the R-CNN [95] model marks a significant leap forward, outperforming previous efforts on multiple detection data sets by a large margin. It aims to replicate the success of CNN based architectures from the ImageNet [94] classification benchmark in the object detection domain.

Network design: The high-level structure of the R-CNN [95] method consists of three central components: (1) a region proposal module to generate an initial set of region candidates, (2) a CNN based feature extractor which encodes the information from the regions into a single, fixed-sized vector and (3) class-specific classifier modules to determine the class of each region. The overall architecture is depicted in Figure 2.1.

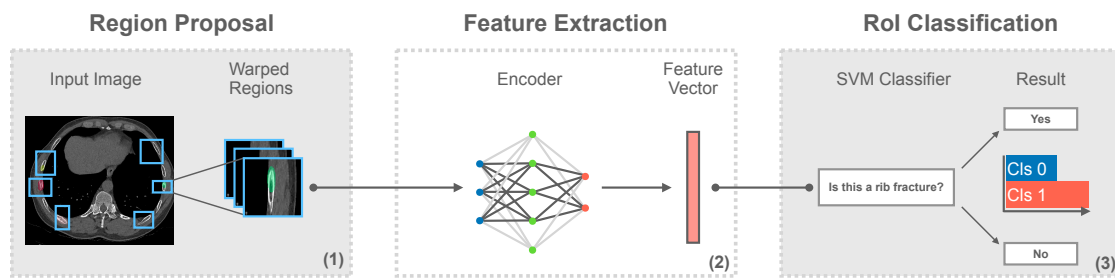


Figure 2.1: Structural overview of Regions with CNN features (R-CNN) [95] detection method. An initial set of region proposals is generated via the selective search algorithm [96] and warped into fixed-sized regions (blue). Alex Net [92] is used to extract a feature vector (red) from each region and processed by linear Support Vector Machines (SVMs) [97] to classify the proposals. CT slice taken from RibFrac data set [11, 12]. Reference segmentations are shown to indicate Regions of Interest (Rols). The structure of this figure is adapted from [95].

Region proposals: The region proposal module is realized via the selective search method [96] to enable a fair comparison against prior work. However, there are no restrictions on the design of the region proposal algorithm and other methods can be used instead. Selective search first generates a large number of small segments [98], aiming to find potential object positions without encapsulating multiple objects into a single segment. These segments are progressively grouped into larger clusters, generating a hierarchical grouping that captures objects on multiple scales. The groups present potential objects and are called region proposals. During the inference process of the R-CNN method, 2000 initial class agnostic region proposals are generated.

Feature extraction: The feature extractor (encoder) is represented by Alex Net [92], which generates a feature vector of size 4096 for each region proposal. Object detection labels are scarce and expensive to annotate, which poses a significant problem for data-hungry neural networks. To alleviate this problem, the feature extractor is first pre-trained on the ImageNet [94] data set to initialize its weights. Since the selected CNN architecture requires an input image size of 227×227 , all-region proposals undergo affine image warping to a fixed image size. To adapt to the changed classification setting of the warped region proposals, the network is then fine-tuned on the target data set via a classification task. This requires the assignment of a target label to each region proposal: region proposals with an Intersection over Union (IoU) of 0.5 and higher are considered positive examples during training, while all other classes are considered negative. The classification layer of this network is not further used for detection.

Region classification: The classifier for the object detection stage is represented by class-specific linear Support Vector Machines (SVMs). Greedy Non-maximum suppression (NMS) removes duplicate predictions for each class separately. The label assignment of the SVM classifier differs from the assignment rule used for the feature extractor: only ground truth bounding boxes are considered as positive samples, while proposal boxes with an IoU below 0.3 are considered negative. Proposals which are above an IoU threshold of 0.3 are ignored during training. To balance the positive and negative samples, Hard Negative Mining (HNM) [99, 100] is used.

Optionally, during a third training stage, additional regression targets can be predicted to refine the initial proposal bounding box to boost the localisation performance. The proposals are only used for this training step if their IoU with the most similar ground truth bounding box exceeds 0.6. The regression targets $t_{\{c,s\}}$ are set to

$$t_c = \frac{(b_c - \hat{b}_c)}{\hat{b}_s} \quad (2.1)$$

$$t_s = \log\left(\frac{b_s}{\hat{b}_s}\right) \quad (2.2)$$

where b_c is the centre of the ground truth, b_s is the size of the ground truth, \hat{b}_c is the center of the prediction and \hat{b}_s is the size of the prediction.

Fast R-CNN

The original R-CNN approach suffers from several downsides: **(1)** the training needs to be conducted in three stages by first finetuning the CNN, second training the SVM models for classification and finally training the regression models and **(2)** training and inference

are very slow since each region is propagated individually through the neural network. Fast R-CNN [101] tackles these problems by simplifying the training into a single-stage process and sharing the feature computation across the region proposals. A schematic overview is shown in Figure 2.2.

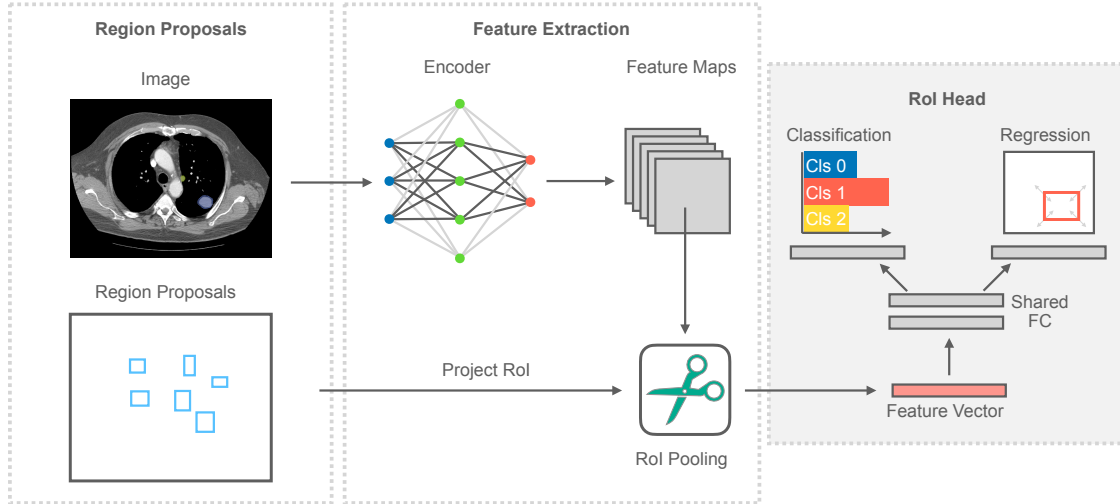


Figure 2.2: Structural overview of the Fast R-CNN [101] detection method. The entire image is processed by an encoder network to extract a low-resolution feature representation. Region proposals are then projected into the same low resolution, and Region of Interest Pooling (RoI Pooling) is used to extract information into a fixed-sized feature vector. Fully Connected (FC) layers first process the created representation, followed by a classification and regression branch to create the final predictions. CT slice is taken from the LIDC [13] data set. Reference segmentations are shown to indicate Regions of Interest (RoIs).

RoI Pooling: By propagating the entire image through the neural network at once, the feature computation is shared across all RoIs, which allows efficient training of Fast R-CNN. The resulting representation of the image has smaller spatial dimensions than the original image, usually by stride $S = 32$. As a consequence, the region proposals need to be projected into the low-resolution space first. The top left corner can be projected with $p_{low} = \lfloor p_{full}/S \rfloor - 1$, and the bottom right corner is represented via $p_{low} = \lceil p_{full}/S \rceil + 1$ as proposed in [102]. p_{low} and p_{full} refer to the coordinates in low and full resolution, respectively. To obtain a fixed-sized feature vector, the projected RoI is divided into roughly equally sized bins, where max pooling is applied. The coordinates of the individual bins follow the same rounding convention as the projection formula [101, 102].

Region classification and regression: The features are then processed by two shared Fully Connected (FC) layers and followed by two branches: the classification branch assigns the class probabilities, including background, to the individual proposals, and

the regression branch provides additional regression offsets to refine the initial proposal. Multi-task learning is used to train the two branches simultaneously with a classification L_{cls} and regression L_{reg} loss, defined as Equation (2.3).

$$L(\hat{y}, y, t^y, b) = L_{\text{cls}}(\hat{y}, y) + \lambda_{\text{reg}} \mathbb{1}_{y \neq 0} L_{\text{reg}}(t^y, b) \quad (2.3)$$

t^y denotes the predicted regression targets for class y . L_{cls} is set to the Cross Entropy (CE) loss and is used to train the classification branch. $y = 0$ indicates the background class, letting the indicator function filter for predictions which are assigned to a foreground object for the regression loss. λ_{reg} is a scalar factor to balance the different loss functions. L_{reg} is set to the smooth L1 loss [101] formalised in Equation (2.4)

$$L_{\text{reg}} = \sum_{i \in \{c, s\}} \text{smooth}_{L_1}(t^y, b) \quad (2.4)$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (2.5)$$

The regression targets t follow the same formulation as Equations (2.1) and (2.2).

Mini-batches of RoIs are assembled by sampling one of four samples with an IoU of at least 0.5, which are considered positives, and the remaining RoIs are sampled with an IoU between $[0.1, 0.5)$ which are considered negatives. The regression is performed for each class separately, and NMS is used to suppress duplicate predictions. The computational efficiency of the network can be further improved by using a truncated Singular Value Decomposition (SVD) to compress the fully connected layers, which become a bottleneck when using a large number of RoIs.

Faster R-CNN

Previous iterations of the R-CNN method still rely on external region proposal mechanisms, like selective search [96], which become a bottleneck during real-time object detection and can hamper performance. Faster R-CNN [36] replaces the external region proposal mechanism with a CNN, which simultaneously generates the feature representation used for the Fast R-CNN detector. An overview of the model is provided in Figure 2.3.

Region Proposal Network (RPN): A neural network, in the original publication, realized as a CNN [36], is used to extract a high-dimensional feature representation of the input image. A sliding window scheme is used to run through all spatial locations

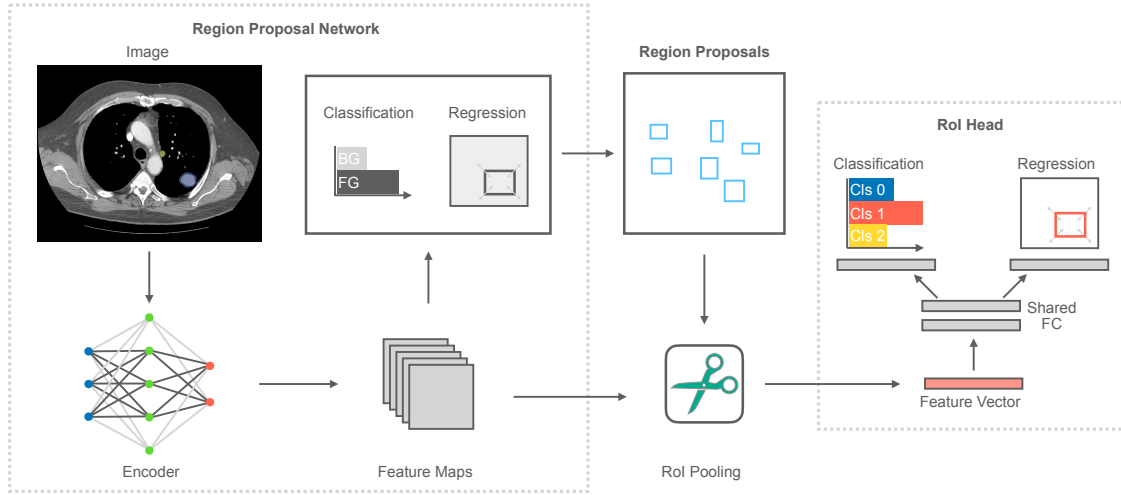


Figure 2.3: Structural overview of Faster R-CNN [36] detection method. An encoder network is used to project the image into a high-dimensional feature space. A sliding window approach, in combination with predefined anchors, is used to predict a set of region proposals by classifying and regressing the anchors. RoI Pooling is used to extract fixed-sized feature vectors from the feature maps, which are processed by two branches like the Fast R-CNN detector. CT slice is taken from the LIDC [13] data set. Reference segmentations are shown to indicate RoIs.

and predict multiple region proposals per location. Each location is associated with multiple anchor boxes [36] with varying scales and aspect ratios to represent templates for objects of different sizes. The prediction is performed via a lightweight classification and regression head consisting of a single shared FC layer and two small branches with one FC layer for classification and regression each. Since the head is shared across all locations, the sliding window scheme, as well as the FC layers, can be efficiently implemented via convolutions. This module is responsible for predicting an initial set of region proposals for the Fast R-CNN detector and is called RPN.

RPN training: During training, the anchors must be assigned a label for the classification and regression losses. Anchors with an IoU of at least 0.7 with a ground truth object or the best matching anchor for an otherwise unassigned ground truth are considered positive examples. This assignment scheme is also called a one-to-many assignment rule since multiple anchors can be assigned to the same ground truth object. Anchors with an IoU below 0.3 are considered negative examples, and anchors between the two IoU thresholds are ignored for the loss computation. The loss functions follow a similar formulation as in Fast R-CNN, see Equation (2.3), where the CE loss (normalised by the batch size) is used to train the classification branch and the smooth L1 loss (normalised by the number of anchor locations) is used for the regression branch. λ_{reg} is set to 10 to balance the contributions of the classification and regression losses. Notably, the

parameters are shared across spatial locations, but an independent regressor is trained for each class and anchor scale. To balance foreground and background samples, the same number of foreground and background anchors is sampled to compute the losses. If there are not enough positive anchors, additional negative anchors are added to keep the batch size consistent.

This design allows for efficient end-to-end training of the Faster R-CNN detector. The training can either be performed in an *alternating fashion* [36], where only the RPN or the RoI head is trained at a time, or in an *approximate joint fashion* [36] where the RPN and RoI head are trained simultaneously and no gradients are backward-propagated through the bounding box coordinates of the RPN. The approximate joint training scheme has demonstrated results similar to those of the alternating training scheme while speeding up the training. The method is not just significantly faster to train than previous approaches but also achieves improved performance on the PASCAL VOC2012 [84] and MS-COCO [85] data sets.

Mask R-CNN

In some scenarios, predicting bounding boxes can be limiting and exact delineations in the form of segmentations are required, such as determining the exact volume of a lesion. Instance segmentation algorithms which produce additional masks for the presented objects are needed for these applications. Mask R-CNN [37] provides a simple extension to the Faster R-CNN detector by adding an additional mask prediction branch to the RoI head. As a result, it can produce high-quality instance segmentations and shows improved bounding box detection performance [37].

Mask head: The mask head uses a fully convolutional design to process the feature map and can be combined with a transposed convolution [103] to increase the spatial resolution of the prediction. The loss used to train the RoI head Equation (2.3) is extended with an additional mask component L_{mask} to learn the segmentation. Since the classifier branch determines the predicted class, binary segmentation masks for each class are computed. The Binary Cross Entropy (BCE) loss compares the mask predictions against the reference annotations.

RoI Align: Predicting segmentations requires the precise alignment of the features to the pixel output and necessitates a refined cropping and resizing strategy for the RoIs. RoI pooling [101](Section 2.1.1) has multiple quantization steps: (1) the quantization of the RoI from the image to the low-resolution feature space and (2) the quantization of the individual bins for the pooling operation. These can lead to misalignments between the features and the produced output, which must be avoided for precise masks. RoI Align [37] avoids these quantization steps by representing the coordinates via floating point numbers and using bilinear interpolation for four regularly spaced points within each bin. A max pooling operation aggregates these points to derive the fixed-sized RoI.

R-CNN Extensions

Due to the effectiveness and wide adoption of the R-CNN detector scheme, several extensions were proposed to refine its design and further improve its result. Cascade R-CNN [53] and Hybrid Task Cascade (HTC) [54] extend the two-stage detection process to multiple stages, which iteratively refine the region proposals of the previous stage. This design aims to progressively refine the initial proposals and improve the localisation quality. The RoI head was further refined in [104], which found that the classification and regression branches benefit from different architectural designs. [105] and [106] identify shortcomings in the assignment and sampling procedures which can be improved to further boost performance. Furthermore, using one-stage detectors (Section 2.1.2) as RPNs provides informative confidence scores for region proposals which can be used to improve the final scores of the predictions [107]. Sparse R-CNN [108] moved away from the design of predicted region proposals and instead learns constant region proposals, which are processed by several refinement stages. This design does not require additional de-duplication steps, removing the need for NMS. Finally, different feature extraction networks, also called backbones, can be used to scale up the capacity of R-CNN detection models and achieve highly competitive results on the MS-COCO data set [109]. The backbone is not limited to CNN architectures; ViTs [34] offer a highly effective alternative [110, 111, 112].

2.1.2 One-Stage Object Detection

One-stage detection methods predict objects in a single shot without requiring operations to be executed on individual RoIs. This simplifies the design of the methods and allows for quicker inference procedures, which are essential in certain applications.

You Only Look Once (YOLO)

The You Only Look Once (YOLO) detector follows a simple design principle by dividing the image into a grid of cells and predicting objects in each cell. The image is propagated forward in its entirety, allowing the network to reason across the whole image in a single shot. Each grid cell can predict M bounding boxes and M confidence scores. The confidence scores provide an estimate for the presence of an object $\hat{p}(\text{object})$ (also referred to as objectness) and the quality of the predicted bounding box \hat{q} . A cell only predicts those objects whose centre lies inside the cell. The quality of the predicted bounding box is measured by the IoU with the ground truth object. Furthermore, each cell predicts class confidence scores $\hat{p}(\text{cls}|\text{object})$ representing the conditional probability of the predicted bounding box corresponding to an object of the respective class.

To obtain the final object confidence score \hat{o}_{pred} during inference, the predicted confidence score and the class confidence score are multiplied according to Equation (2.6).

$$\hat{o}_{\text{pred}} = \hat{p}(\text{cls}|\text{object}) * \hat{p}(\text{object}) * \hat{q} = \hat{p}(\text{cls}) * \hat{q} \quad (2.6)$$

As a consequence, each cell is only capable of predicting a single class, which can be a limitation in images with dense object clusters. To avoid duplicate predictions, NMS is utilized. The concept of the YOLO detector is depicted in Figure 2.4.

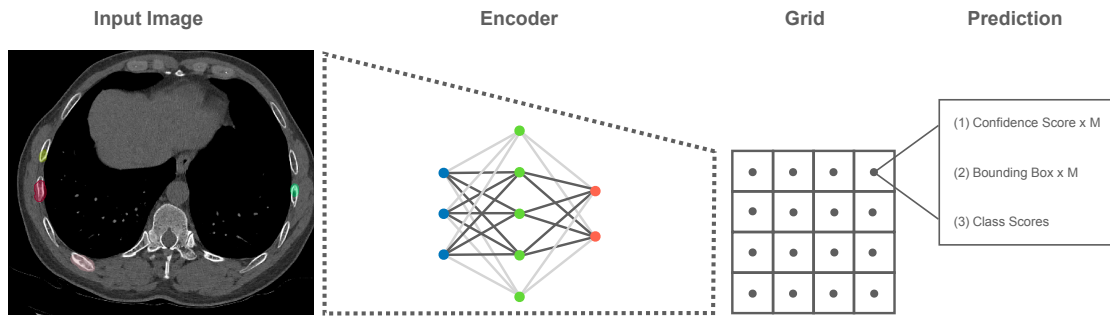


Figure 2.4: Structural overview of You Only Look Once (YOLO) [113] detection method.

The input image is forwarded through a CNN to produce a low-resolution feature representation which natively divides the image into grid cells. Each grid cell can predict a fixed number of bounding boxes and confidence scores to learn size-specific object properties. A single set of class scores is predicted for each cell representing the conditional class probability. Reference segmentations are shown to indicate Rols. CT slice taken from RibFrac data set [11, 12].

Network design: The input image is processed by an encoder, originally using a deep neural network inspired by GoogLeNet [114], to extract features from the image in a grid-like structure. The density of the grid is directly influenced by the stride of the network. In order to benefit from the availability of large image classification data sets in the natural image processing domain, the original publication proposes to pre-train the network on the ImageNet [94] data set and only perform a fine-tuning for the detection task.

Assignment and losses: During the training, the predictions must be assigned to a corresponding ground truth object to compute the losses. The procedure of assigning a prediction to a ground truth object is also referred to as *assignment* in this thesis. The bounding box with the highest IoU is selected for predicting the ground truth object. Ground truth bounding box centres are encoded with respect to the cell boundaries, and the whole image normalizes its size. The sum-squared error is used to optimize the coordinates, the square root of the sizes, the confidence score and the class confidence

scores between the predictions and ground truths. To balance the loss, those cells that only contain background present a lower weight for loss computation.

YOLOv2: The originally presented design for YOLO still suffers from some disadvantages when compared to other architectures like Fast R-CNN [101] and Faster R-CNN [36]: (1) a comparably large number of localization errors and (2) rather low recall. **YOLOv2** [115] introduces several design changes to alleviate the aforementioned challenges. The network is replaced with a more powerful CNN network, called Darknet-19 [115], which utilizes Batch Normalization [116] to speed up the convergence speed.

Furthermore, the anchor scheme, initially introduced in Faster R-CNN [36](Section 2.1.1) is integrated into the model to increase its recall. The anchor sizes in [36] were manually chosen and might not achieve the best coverage across the given data set. To automatically derive an improved set of anchors, this model utilizes k-means clustering to find better object clusters and give a more appropriate anchor initialization. Furthermore, the encoding of the predicted regression targets is exchanged to stabilize the training: the coordinates are encoded with respect to the current cell, and a sigmoid function normalizes the outputs of the network.

Finally, to force the network to learn scale-adaptive features, a multi-scale training scheme is utilized to change the resolution of the input images during the training. The resulting model shows promising results on the Pascal VOC 2007 [84] and MS-COCO [85] data sets.

YOLOv3: The design can be further refined by integrating additional components into the detector. Instead of relying on a one-to-many assignment for training like [101, 36], it uses a one-to-one assignment where the ground truth object is only assigned to the anchor with the best IoU. Any anchor exceeding the pre-defined IoU threshold without being assigned to a ground truth object is only used to compute the objectness loss and ignored otherwise. Each classifier is trained independently by using the BCE loss and a sigmoid activation function. Furthermore, the CNN structure was extended with residual connections [29] and scaled in depth. The resulting network is called Darknet-53. Instead of relying on a single scale for prediction, a design inspired by the Feature Pyramid Network (FPN) [117] is utilized to predict objects from multiple scales, which boosts performance on small objects. The developed method follows the original principles of YOLO by offering a good trade-off between detection performance and runtime.

Single Shot MultiBox Detector (SSD)

The Single Shot MultiBox Detector (SSD) [118] follows a similar design principle as the YOLO [113] detector by circumventing region proposals and directly producing the object outputs. It utilizes a similar anchor design as Faster R-CNN[36] and combines it with a multi-scale feature approach to predict at multiple resolution levels. By varying the

anchor scales between levels, features can specialize on different object sizes. A schematic of this approach is visualized in Figure 2.5

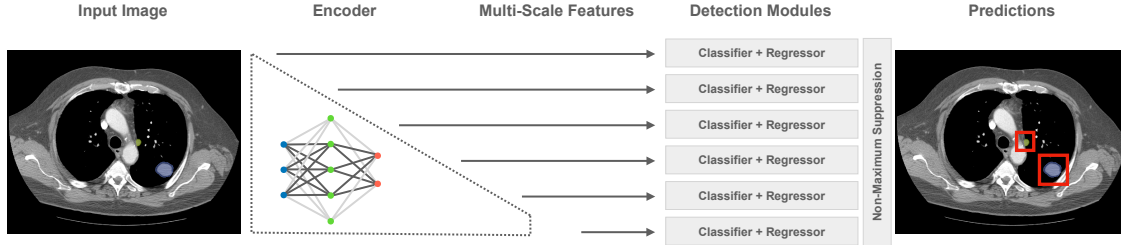


Figure 2.5: Structural overview of Single Shot MultiBox Detector (SSD) [118] detection method. The input image is processed by an encoder, for example, a CNN, which produces features at multiple resolution levels. Each level used for the detection task is processed by a detection module to classify and regress the pre-defined anchors. The sizes of the anchors are varied between the levels to learn size-specific features. The output from the levels is collected and processed with NMS to deduce the final set of predictions. Reference segmentations are shown to indicate Rols. CT slice is taken from the LIDC [13] data set. The structure of this figure was adapted from [118].

Network design: The VGG-16 [93] network is used as the basis for the design of the SSD detector. It is modified to a fully convolutional design, and additional convolutions progressively reduce the spatial dimensions of the features, which are appended at the end. This design allows the extraction of feature maps of varying sizes that capture information at different granularity levels.

Each feature map used for detection contributes A anchor boxes per spatial location. A 3×3 convolution is used to represent a sliding window approach which refines and classifies (C classes) the anchors at each position. Each anchor is regressed via four coordinates in the 2D case. Given a feature map of size $M \times N$, this results in $M * N * A$ object predictions represented as a $(4 + C) * A \times M \times N$ feature map. The regression targets follow the same encoding as in the R-CNN detector [95].

Anchors and multi-scale features: In theory, there are no restrictions on how the anchor boxes are selected for the different feature levels. Intuitively, levels with low resolution are located deeper in the network and have larger receptive fields, which allows them to capture information from larger objects effectively. High-resolution feature maps can encode fine-grained information, which is essential for predicting small and potentially dense objects. To encode this behaviour into the network, levels can be associated with different scales such that larger anchors are positioned in deeper layers, and small anchors are located in earlier layers. This design allows the levels to specialize in different object sizes. If L levels should be utilized for the detection, the scales can be distributed as in Equation (2.7).

$$s_l = s_{\min} + \frac{s_{\max} - s_{\min}}{L - 1}(l - 1), l \in [1, L] \quad (2.7)$$

s_{\min} represents the minimal scale and is set to 0.2 [118], s_{\max} represents the maximal scale and is set to 0.9 [118]. The intermediate scales are distributed regularly. Additionally, each location is associated with anchors of different shapes, which are modelled by different aspect ratios of their sizes. The centre of each anchor is placed at the centre of the respective feature location.

Training scheme: All anchors with an IoU above 0.5 and the best matching anchor per ground truth object are considered positive examples. A multi-task loss is utilized, being composed of a classification and coordinate regression loss. The CE loss is used to train the classifier, while the smooth L1 [101] loss is used to train the regressor. The proposed detection design suffers from a large class imbalance since there are many more negative anchors than positive ones. Hard Negative Mining (HNM) is used to counteract this imbalance by only sampling the highest-scoring negatives up to a ratio of three (negative): one (positive) anchors. Furthermore, extensive data augmentation forces the network to learn multi-scale information by either sampling an entire image, extracting a sub-patch with a minimal IoU with ground truth objects or extracting a random sub-patch.

Retina Net

The previously presented one-stage detectors offer a simple design and quick inference speed but fail to compete with state-of-the-art two-stage detection detectors on common natural image benchmarks like the MS-COCO [85] data set. One of the major bottlenecks when training this type of detector is the presence of many more negative anchors than positive ones, skewing the gradients and making training inefficient. Furthermore, the vast majority of these negative examples are located in easy-to-classify regions, which only contribute to minor improvements in detection performance. To alleviate this problem, different variants of HNM were used in detectors [118, 119] but failed to adequately tackle this issue for one-stage detectors. In contrast to two-stage detectors, one-stage detectors work with a larger number of anchors, aggravating the aforementioned imbalance of positive and negative examples.

Focal Loss: To address the significant imbalances between positive and negative anchors as well as easy and hard-to-classify examples, the focal loss [35] was proposed. This loss is presented as a central contribution of Retina Net [35] and presents an alternative to HNM to automatically adjust the weighting, see Equation (2.8).

$$L_{\text{focal}} = -\alpha_t(1 - \hat{p}_t)^\gamma \log(\hat{p}_t) \quad (2.8)$$

$$\hat{p}_t = \begin{cases} \hat{p}, & \text{if } y = 1 \\ 1 - \hat{p}, & \text{if } y = 0 \end{cases}, \quad \alpha_t = \begin{cases} \alpha, & \text{if } y = 1 \\ 1 - \alpha, & \text{if } y = 0 \end{cases}$$

$y \in 0, 1$ indicates the binary ground truth class, \hat{p} is the predicted probability of the classifier for a single class. By using a sigmoid activation function on the output of the classifier and one-hot encoded reference classes, the loss can be extended to the multi-class case. Focal loss can be adjusted via two hyperparameters: (1) α can be used to balance the importance of positive and negative examples, and (2) γ balances the weight of easy and hard examples. By setting $\gamma \geq 1$, easy-to-classify examples receive a much lower weight than hard examples. If $\gamma = 0$, the focal loss becomes equivalent to the commonly used BCE loss.

In addition to the focal loss, the initialization of the anchor classification layer is changed to stabilize convergence at the beginning of training, where many negative examples can dominate the training signal. The prior value p_{prior} is introduced as a hyperparameter, for example $p_{\text{prior}} = 0.01$ [35], and Equation (2.9) is used to initialize the bias value of the last convolution of the classification branch in the detection head.

$$\text{bias} = -\log\left(\frac{1 - p_{\text{prior}}}{p_{\text{prior}}}\right) \quad (2.9)$$

Feature Pyramid Network (FPN): The effective usage of multi-scale features is essential for achieving robust performance with one-stage detectors. To achieve this, Retina Net [35] uses a ResNet [29] backbone network to extract features and a Feature Pyramid Network (FPN) to recombine coarse features with fine-grained ones. Features from the lower levels are progressively up-sampled via bilinear interpolation and combined with higher-level features via elementwise addition. All created feature maps have the same number of channels, so the detection head can be shared across all levels. The detection head of Retina Net [35] consists of a classification and regression branch, each with four intermediate convolutions and an additional output convolution. The classification branch is responsible for classifying the anchor, while the regression branch predicts regression offsets for each class-agnostic anchor. The parameters between the branches are not shared. The resulting network architecture is shown in Figure 2.6

Ablation experiments in [35] show that increasing the number of anchors per location improves performance. Consequently, each spatial location in the feature maps is associated with a set of nine anchors defined by three sizes and three aspect ratios. The final model is the first detector able to unify the simple design of one-stage detectors with state-of-the-art results from two-stage detection models.

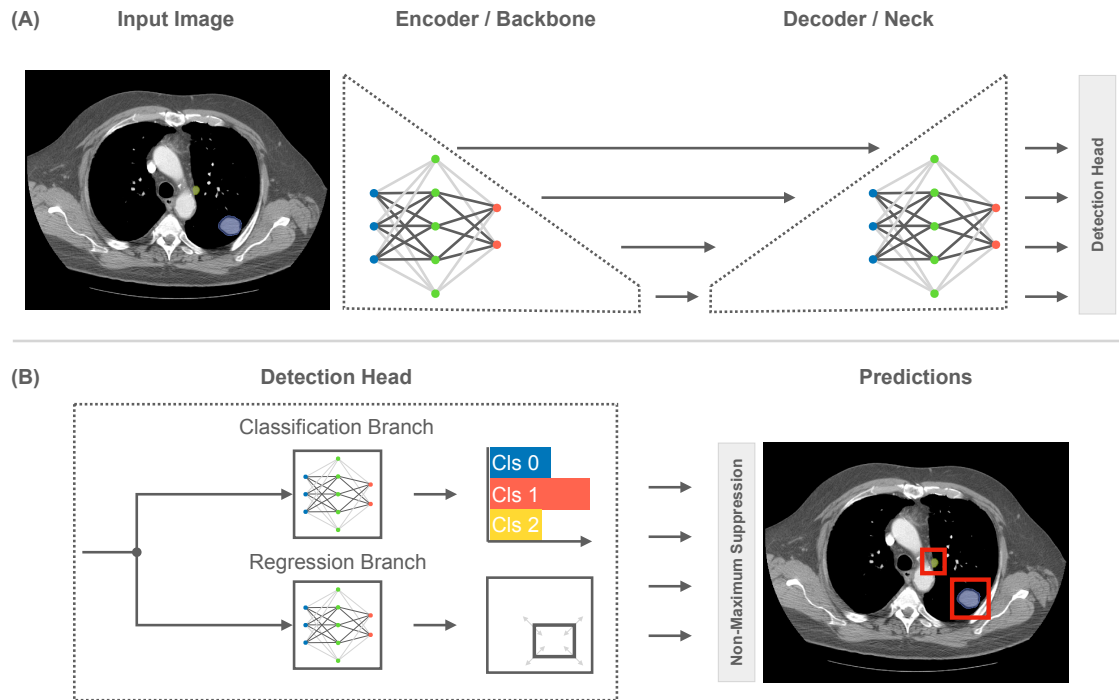


Figure 2.6: Structural overview of Retina Net [35] detection method. The input image is first processed by an encoder-decoder (also called backbone-neck in this thesis) architecture to extract multi-scale information. Each level is processed by a shared detection head consisting of two branches: one for classification and one for regression. The outputs are collected and processed via NMS to remove duplicate predictions. Reference segmentations are shown to indicate Rols. CT slice is taken from the LIDC [13] data set.

One-stage Detection Extensions

YOLO: This line of detectors is still continued by various other authors with a focus on balancing speed and performance. Over time many more variations were proposed including YOLOv4 [120], YOLOv6 [121], YOLOv7 [122], YOLOv9 [123] and YOLOX [124] using more powerful backbone architectures, more sophisticated augmentation schemes, different loss functions and model scaling.

Architecture scaling: The availability of increasingly larger data sets and faster computing resources fuels the development of powerful models. Pre-training is still an essential part of the object detection community, which started leveraging bigger classification data sets like ImageNet21k [125] for training backbones and large-scale detection data sets like Objects365 [87] for pre-training-entire detection networks [126, 127]. Further advancements for backbones are driven by self-supervised learning like Masked Autoencoders [128, 109] or Multi-Modal settings [112, 111, 129, 130] which allows the usage

of unlabeled data to learn representations across millions of images [129]. These rich resources enable the training of other feature extractors like ViTs [34, 110]. Increasing the model’s capacity is not just limited to the encoder network but is also pursued by the decoder network, which is responsible for recombining low-level features with high-level ones. These designs add additional connections [131, 132, 133] to the decoder to combine information from different layers or apply the decoder several times [126]. Scaling individual components might be suboptimal for achieving the best possible performance. Hence, different model scaling strategies have been proposed to analyse the impact of training length, augmentation settings and compound scaling [134, 126].

Loss functions: Further advancements can be observed in the development of loss functions to improve both the classification and localization performance of current detection models. The generalized focal loss [135, 136] follows a similar idea as the YOLOv1 [113] detector by merging the quality measure of the bounding box prediction and the classification information into a single value. This forces the neural network to align the confidence score to the expected bounding box quality and avoids diverging predictions. Loss functions to regress the anchors are also improving by moving away from approximations to directly penalize the difference in IoU between the prediction and the reference box. The formulation of the IoU loss can be found in [137] but is usually replaced by the Generalised Intersection over Union (GIoU) loss [138]. The original IoU loss was not able to compute gradients between predictions and reference boxes, which did not overlap with each other. This downside is alleviated by the GIoU loss, which relies on the encapsulating box of the prediction and ground truth to penalize non-overlapping boxes. Further extensions in the form of the Distance IoU [139] loss and Complete IoU [139] loss incorporate additional characteristics of the centre points and shapes to refine the regression performance.

Assignment strategies: The assignment of anchor boxes to ground truth boxes is realized as a manual heuristic which can be manipulated and exchanged for alternative formulations. Adaptive Training Sample Selection (ATSS) [140] moved away from a single static IoU threshold towards a more flexible formulation which can derive a dynamic threshold based on the presented characteristics of the object. The proposed change in the assignment can bridge the gap between classic anchor-based detection approaches and newer approaches like Fully Convolutional One-Stage Object Detection (FCOS) [55], which do not rely on the design of prior anchor boxes. Other assignment strategies which aim to remove the dependency on an IoU can also be used, for example, in the form of a probabilistic assignment strategy [141]. Instead of solely relying on the similarity of the boxes, it is also possible to implement other similarity measures that combine differences from classification and localization into a single score. Multi-task loss functions are able to quantify these differences and can be combined with optimization techniques, like optimal transport, to derive an assignment [142]. Recently, one-to-one [143] or one-to-few [144] assignment strategies have become more popular to avoid the dependence on NMS for de-duplicating predictions.

2.1.3 Direct Set Prediction for Object Detection

DEtection TRansformer (DETR)

Anchor-based models decompose object detection as a classification and regression task by using a surrogate task. This workaround is needed for CNN based approaches, which are not natively able to predict a dynamic number of objects. As a consequence, these approaches rely on several manual heuristics: sizes and shapes of anchor boxes, assignment rules to find correspondences between anchor and ground truth boxes, as well as the NMS heuristic to remove duplicate predictions. The DETR model popularized a different formulation for object detection tasks, namely *direct set prediction*. Instead of relying on anchors [35, 36, 37] or other priors (like centre points [55]), matching is directly performed on the outputs by searching for an optimal bipartite matching. This yields a one-to-one correspondence between predictions and ground truth objects, eliminating the need for handcrafted heuristics. The core functionality of DETR models can be split into two components: (1) a mechanism to match predictions and ground truth objects to compute the detection loss and (2) an architecture able to predict a set of bounding boxes and class correspondences.

Matching and Loss: In order to compute a loss between the predictions and ground truth objects, which is necessary to train any deep learning-based detection model, a matching between the set of predictions and the set of ground truth objects needs to be derived. The mechanism to perform the matching needs to fulfil two requirements: (1) No additional postprocessing steps, like NMS, should be required to produce the outputs. This necessitates that the network only produces a single output per object and requires a one-to-one matching strategy during training. (2) The matching must be permutation invariant since objects do not have a natural ordering within images.

Let \hat{y} denote the set of predictions from the model and y the set of ground truth objects. The number of predictions the model provides remains constant for all images, while the number of ground truth objects naturally varies between images. To ensure that all objects in an image can be predicted, the number of predictions, represented as a hyperparameter, must exceed the number of objects in the image. To still compute a one-to-one assignment, the set of ground truth objects is extended with additional "no object" \emptyset elements until it has the same size as the set of predictions. To fulfil the previously derived requirements for the matching strategy, an optimal bipartite matching can be determined; see Figure 2.7.

To find a suitable matching, it is necessary to measure the similarity between predictions and ground truth objects to assign a cost for each potential pair. The cost function L_{cost} measures both similarity in the class assignment and spatial similarity. $L_{\text{cost-cls}}$ measures the similarity between the predicted class and the assigned class and follows a similar design as a classification loss function. Spatial similarity can be measured via typical

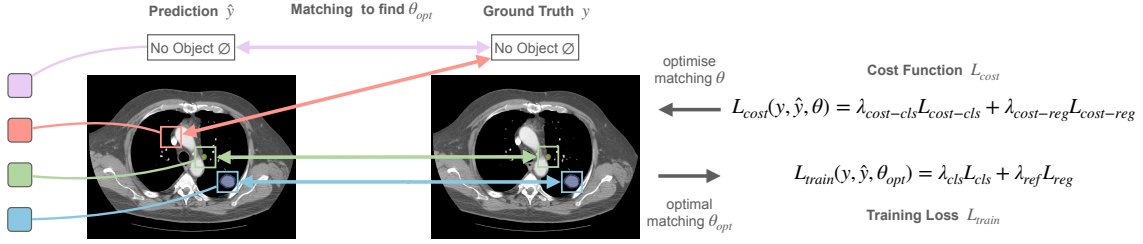


Figure 2.7: Matching between predictions and ground truth of DETR [38] detection method. A direct set prediction method produces a set \hat{y} representing bounding boxes and class correspondences. The ground truth set of objects is extended with additional elements representing background (or "no object") until it contains the same number of elements as the set of predictions. The cost function L_{cost} measures the similarity between the predictions and ground truth objects to assign a cost to the current assignment θ . After finding the best matching θ_{opt} , the training loss can be computed. Reference segmentations are shown to indicate RoIs. CT slice is taken from the LIDC [13] data set.

regression loss functions and is expressed as $L_{cost-reg}$. Let i denote the i -th pair between the two sets, c_i the class of the ground truth, \hat{p}_i the predicted probability for element i , \hat{b}_i the predicted bounding box, b_i the assigned ground truth bounding box and L_{GIoU} the GIoU loss. The cost for an assignment θ , which defines the pairing between the sets, can then be written as

$$\begin{aligned} L_{cost}(y, \hat{y}, \theta) &= \sum_i^N \lambda_{cost-cls} L_{cost-cls,i} + \lambda_{cost-reg} L_{cost-reg,i} \\ &= \sum_i^N -\lambda_{cost-cls} \hat{p}_i + \lambda_{cost-reg} \mathbb{1}_{c_i \neq \emptyset} (\lambda_{L1} \|b_i - \hat{b}_i\| + \lambda_{GIoU} L_{GIoU}(b_i, \hat{b}_i)). \end{aligned}$$

The resulting matching problem can be effectively solved by using the Hungarian method [145] to find the optimal bipartite matching θ_{opt} . After finding the matching which minimizes the cost function, the training loss function L_{train} can be formulated as

$$L(y, \hat{y}, \theta_{opt}) = \lambda_{cls} L_{cls} + \lambda_{reg} L_{reg} \quad (2.10)$$

where L_{cls} is set to the CE loss and L_{reg} is set to the sum of the GIoU and L1 loss. This design allows the direct prediction of a set of bounding boxes and confidence scores without relying on any other proxy task formulation.

Architecture: An overview of the individual components of the DETR architecture is shown in Figure 2.8. The first component of the DETR architecture is an encoder

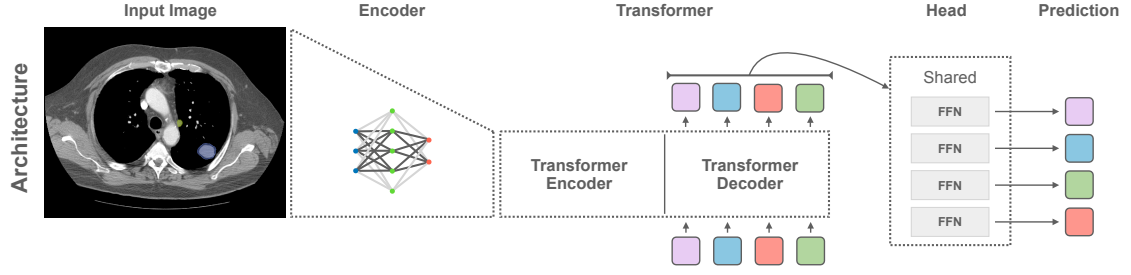


Figure 2.8: Architectural components of DETR [38] detection method. An encoder network first processes the image to extract low-resolution representation. Afterwards, it is fed into a transformer model with an encoder and a decoder model. The decoder uses cross-attention and learned object queries to produce an initial set of features which are predicted by a shared Feedforward Neural Network (FFN) to produce the final confidence scores, classes and bounding boxes. Reference segmentations are shown to indicate RoIs. CT slice is taken from the LIDC [13] data set.

network which processes the input image to extract a low-resolution feature map. Since the design does not impose any special restrictions on the architecture, both CNN and transformer-based feature extraction networks can be used. A point-wise convolution reduces the number of channels of the generated feature representation. The spatial dimensions of the result are flattened into a sequence where each spatial location is interpreted as a token. To maintain information about the location of individual tokens, a spatial encoding is added to them. The spatial encoding is an alternating collection of sine and cosine functions with different frequencies to provide a unique indicator for each position in the feature map. The one dimensional formulation [146] is shown in Equation (2.11),

$$PE_{(l,2k)} = \sin\left(\frac{l}{10000^{\frac{2k}{d}}}\right), PE_{(l,2k+1)} = \cos\left(\frac{l}{10000^{\frac{2k}{d}}}\right) \quad (2.11)$$

where l denotes the spatial position, k is the index of the channel, and d refers to the dimension of the feature vector.

Transformers [146] are powerful models to process sequences of information with Multi-Head Attention (MHA). The core design depends on the attention mechanism, which can be interpreted as a weighted dictionary lookup. Let the queries Q , keys K and values V be represented by matrices. The *scaled dot-product attention* [146] mechanism can then be formulated as

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (2.12)$$

where the dot product between Q and K measures their similarity. The result is used to assign a weight to the entries in V . MHA uses multiple of these attention operations in parallel by first applying different linear projections to its inputs and concatenating the results after the attention operation. Another linear layer processes the concatenated output. When the same input is used for Q , K and V , this operation is also called Multi-Head Self-Attention (MHSA).

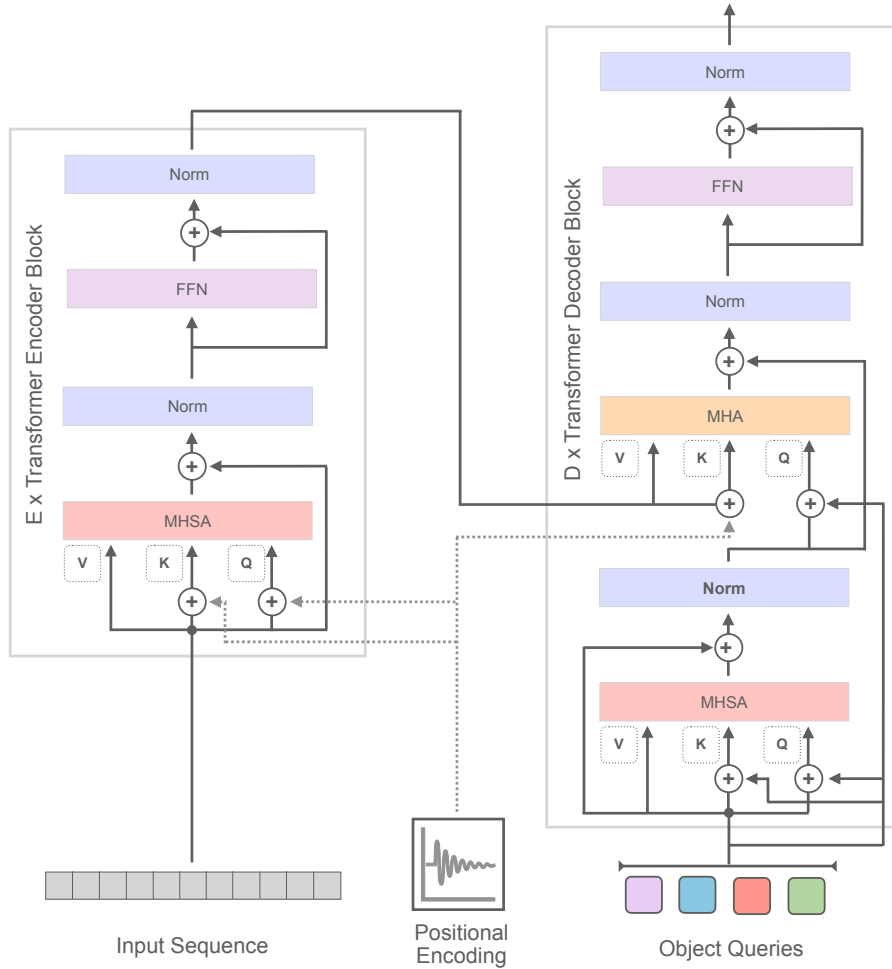


Figure 2.9: Transformer architecture of DETR (DEtection TRansformer) [38] detection method. The transformer encoder processes the flattened feature map from the backbone network with multiple transformer blocks using MHSA. Positional embeddings are to preserve the location information of individual tokens. The transformer decoder decodes multiple queries in parallel using self-attention and cross-attention. Object queries are added to ensure the prediction of different objects. Image adapted from [38].

An overview of the transformer architecture of the DETR detector is shown in Figure 2.9. The transformer encoder consists of $E = 6$ transformer blocks, each using a MHSA

operation, two normalization layers, a FFN and two residual connections. The positional encoding is added to the keys and queries before each attention layer in the encoder network. The transformer decoder allows the decoding of queries while obtaining information from the transformer encoder. This is achieved by using cross-attention, where the information from the transformer encoder is used as the keys, and the queries are computed from the decoder. Due to the permutation invariance of the transformer operations, an additional positional encoding is added to the decoder's queries to make them distinct from each other. These encodings are represented as learnable embeddings and are called object queries. The decoder consists of $D = 6$ transformer decoder blocks where the intermediate outputs are also processed by the detection head to produce additional auxiliary losses.

The detection head is represented by two branches responsible for classification and bounding box prediction. A linear layer represents the classification branch, while the bounding box branch is a three-layer Multilayer Perceptron (MLP). To further optimize ranking-based detection metrics, the highest-scoring foreground class is used for each prediction.

Deformable DETR

While the introduction of the direct set prediction scheme via transformers marked the beginning of a new type of detection model, it also suffered from some downsides when compared to classical detectors. The transformer and, thus, the detection process only has access to a single resolution, due to the quadratic increase in compute and memory with longer sequence length. As a consequence, it obtains very promising results for large objects but suffers from performance deficits for detecting smaller structures. Furthermore, it requires a significantly extended training schedule to compete against anchor-based detectors. The model was trained for 500 epochs on the MS-COCO [85] data set, exceeding typical training times from Faster R-CNN by an entire magnitude.

To address these shortcomings, Deformable DETR [147] introduces the concept of Deformable Attention, which replaces the typical self-attention mechanism in the encoder and cross-attention layers of the decoder. By further refining its design with iterative bounding box refinement [147] and a two-stage formulation [147], it achieves highly promising results on the MS-COCO [85] benchmark.

Multi-Scale Deformable Attention: MHSA sets each point in relation to all other points, which produces a lot of computational overhead. Deformable Attention [147] uses a sparse attention mechanism by restricting computations to a sparse set of reference points. The basic principle is visualised in Figure 2.10. The query feature is used to predict multiple offsets from the reference point where the information is extracted from a feature map. Additionally, a linear layer and a softmax operation are used to determine attention weights from the query feature. The information from the points and the

attention weights are aggregated into a single vector. Like multi-head attention, this operation is performed for multiple heads in parallel, using different projection layers to extract information from different sampling offsets. The results from the heads are concatenated and processed by a linear layer. This principle can be extended to process multi-scale information by projecting the reference point to different resolution levels and predicting separate offsets and attention weights for them.

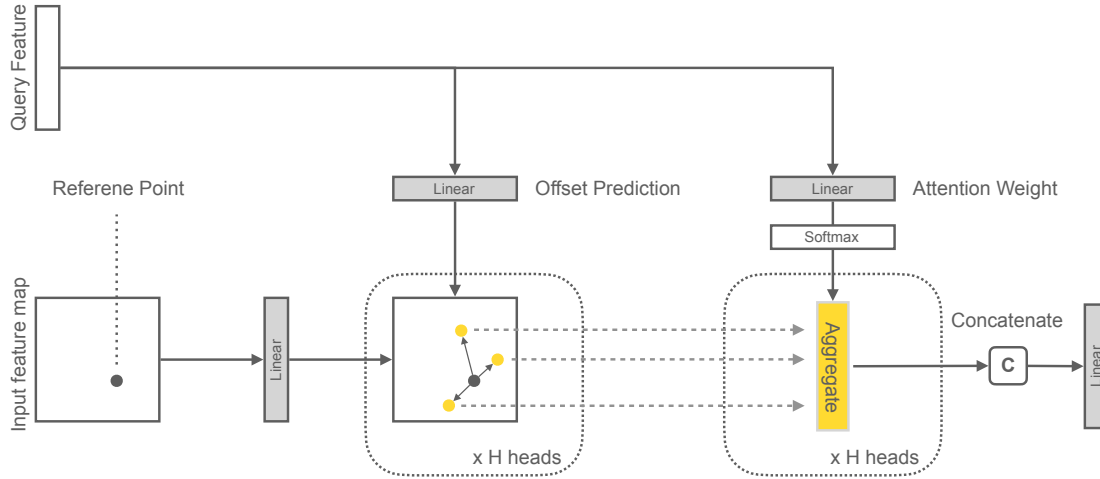


Figure 2.10: Schematic of Deformable Attention [147]. The query feature is used to predict offsets and attention weights for the reference point. The offsets are applied to the reference point to extract information from multiple spatial locations of a feature map. Attention weights are used to aggregate the information from different positions. This process is performed for multiple attention heads in parallel, and the result is concatenated. The final output is obtained by applying a linear layer to the concatenated features. Information can also be extracted from multiple resolution levels simultaneously, but only a single level is shown for simplicity. Image structure adapted from [147].

Iterative Bounding Box Refinement: All intermediate features produced by the transformer decoder are used to produce auxiliary bounding box predictions. Instead of predicting the boxes from scratch each time, iterative bounding box refinement refines the predicted bounding box from the previous block. The bounding box b of block bl can be obtained by $b_{bl} = b_{bl-1} + \Delta_{bl}$ where Δ_{bl} denotes the predicted regression delta from the bl -th block (coordinate normalization is omitted for simplicity). In contrast to the original DETR model, the FFN detection heads are not shared between decoder blocks. To stabilize training, the gradients are only propagated back through the predicted regression deltas and not through the previous bounding boxes. Since the bounding boxes are iteratively refined, their centres can be used as reference points for the deformable

attention operation. Additionally, the sampling offsets can be scaled by the predicted bounding box size.

Two-Stage Deformable DETR: Using each spatial position as a query in the transformer decoder results in unreasonable memory consumption. Therefore, an initial set of object proposals is generated first. To obtain a high-quality set of initial reference points, the encoder features are first processed by a detection head which predicts confidence scores and bounding boxes. Each spatial location in the feature map serves as an initial template with scale $0.05 * 2.0^l$ where l denotes the l -th level used for prediction. The highest-scoring predictions are selected as reference points for the transformer decoder. Furthermore, the predictions undergo a positional encoding step and linear projection to be used as positional embeddings of the queries.

Training Details: Deformable DETR is based on the same principles as the original DETR architecture and follows a direct set prediction approach. Since the number of predicted objects is usually significantly larger than the number of ground truth objects in the image, many objects are assigned to the 'no object' category, leading to an imbalanced classification task. This can be alleviated by using the focal loss [35], which assigns a dynamic weight to the predictions and allows for a further increase in the number of predictions from the model. The final prediction is based on the top- k highest-scoring predictions.

DETR Extensions

The introduction of the DETR architecture sparked the interest of the domain, and a large number of follow-up publications improved upon its design principles. Conditional DETR [61] decouples the spatial information from the content information in the attention operation to speed up training convergence. Additionally, reference points are used to represent spatial information of the object queries. This direction was further developed into the design of Dynamic Anchor Box - DETR (DAB-DETR) [60], which is not limited to reference points but introduces learnable anchor boxes to incorporate information about the size of the objects. This improves the focus of the queries, and the attention operation can be interpreted as a soft RoI Pooling in this scenario. DN-DETR [59] introduces denoising groups to accelerate the training due to instabilities in the Hungarian matching. Ground truth objects are augmented via noise, and the model is tasked to reconstruct the proposals, which provides an additional auxiliary task to stabilize the training. DINO-DETR [39] proposes a contrastive denoising task, a *mixed query selection* scheme to initialize the positional part of the decoder queries while keeping the content parts static and a refined scheme for iterative bounding box refinement.

2.1.4 Other Detection Approaches

While this thesis focuses on the design of anchor-based one-stage, anchor-based two-stage and direct set prediction methods due to their popularity in the natural image processing domain, more formulations for the detection task exist. FCOS [55] proposes a one-stage detection design which is built upon centre points rather than anchor boxes. This design reduces the number of hyperparameters associated with the design of a one-stage detector and shows improved performance. Zhang et al. [140] investigated the performance difference between point-based and anchor-based one-stage detection models and close the gap by proposing a new matching strategy.

Instead of directly representing objects, extreme point detectors formulate the detection task as a keypoint prediction and grouping problem. CornerNet [56] represents objects by their top-left and bottom-right corner points, which are predicted as part of a heatmap. The grouping of the points is determined via predicted associative embeddings. ExtremeNet [57] uses all four corners and the centre point to represent objects. Grouping is performed by looking at each possible combination of corners and using geometrical constraints to filter irrelevant combinations.

Pix2Seq [148] takes inspiration from the language modelling domain and represents objects and labels as tokens which are conditioned on the image. The proposed approach includes minimal task-specific biases and produces its output via next-token prediction.

2.2 Evaluation

The evaluation of object detection tasks requires multiple sequential steps since the predicted set of objects and the set of ground truth objects do not naturally inhibit correspondence. Consequently, the similarity between the predicted bounding boxes and the ground truth bounding boxes must be measured via a localization criterion. Based on the measured similarities and the confidence scores from the predictions, it is possible to match predictions and ground truth objects. This assigns the matched label to each prediction and determines missed ground truth objects, as visualised in Figure 2.11. Given a defined confidence threshold, this yields the number of True Positives (TPs), False Positives (FPs), and False Negatives (FNs), which can be used in subsequent ranking-based or counting-based metrics to quantify the detector performance. Evaluation on the object level does not consider True Negative (TN) predictions since there is an infinite number of possible predictions which do not correspond to any object in the image. To weigh each class equally, each step is performed separately for all classes, and the results are averaged at the end. This also ensures that only predictions and reference objects of the same class are matched. A systematic approach for selecting the correct localization criterion, matching scheme and metric can be found in [48]. Potential pitfalls

for individual components can be found in [149]. The following sections will introduce the localization criterion, matching scheme and metrics used in this thesis.

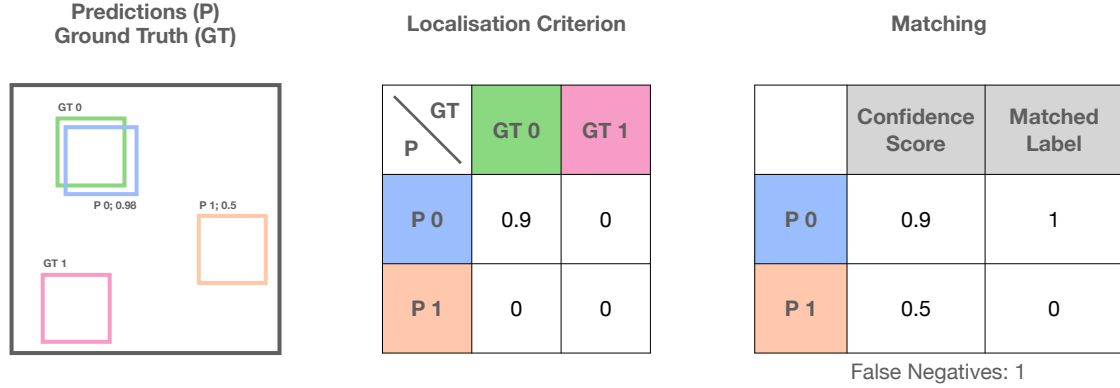


Figure 2.11: Object-level evaluation procedure. Object detection tasks have two sets of objects: ground truth objects and predictions with an associated confidence score. A localization criterion calculates the similarity between the predictions and ground truth objects. During the matching phase, predictions are assigned to a single object and a matched label is derived. Based on the matched labels and the false negatives, it is possible to compute counting and ranking-based metrics.

2.2.1 Localization Criterion

The localization criterion defines the desired granularity of the spatial information produced by the detection model. If the location and size of the objects should be incorporated, for example, when a rough delineation is needed, the Intersection over Union (IoU) can be used:

$$\text{IoU}(A, B) = \frac{A \cap B}{A \cup B} = \frac{\text{Intersection}(A, B)}{\text{Union}(A, B)}. \quad (2.13)$$

The IoU measures the overlap between two objects and can be applied to both instance segmentation and bounding boxes. Since most evaluation metrics depend on counting statistics, at least one additional IoU threshold t_{IoU} must be derived to define a sufficiently accurate prediction. A typical choice is the selection of multiple thresholds to reflect varying degrees of granularity in the predictions. Since the IoU has a cubic decline in 3D and quadratic decline in 2D cases, the IoU thresholds are usually lower for 3D detection tasks.

Another commonly used localization criterion in the medical domain is the distance between the centre points of the ground truth and prediction. This criterion primarily

measures the position of the predictions rather than the correct size of the object. Similar to the IoU localization criterion, a maximal distance needs to be defined to define sufficiently accurate predictions. To adapt the necessary precision on a per-object basis, the threshold can be selected based on the radius of the ground truth object. This is especially useful for spherical annotations. For very small objects, it is also possible to select a constant maximal distance.

2.2.2 Matching

After defining the localization criterion to measure the similarity between objects, the actual correspondence between the prediction and ground truth must be determined. In addition to the spatial information, all methods in this thesis produce a confidence score to allow for general-purpose detection. In this scenario, *greedy matching by confidence score* [48] can be utilized to find the correspondences. Starting with the highest scoring prediction, the most similar ground truth object is assigned to it if the prediction is sufficiently accurate and the ground truth was not assigned to another prediction. Duplicate predictions are considered FP, and thus, each ground truth can be assigned to at most one prediction. This process is repeated by iteratively going through all of the predictions in descending order of their confidence scores. If no-confidence scores are available for the predictions, other matching strategies can be used. Please refer to [48] for information on these.

2.2.3 Counting Metrics

If the object level performance needs to be evaluated at a single confidence threshold, also called a single working point, counting metrics can be used to compute common statistical quantities. Recall R (also called sensitivity) and precision P are often used to quantify the detection performance due to their effectiveness in incorporating TP, FP and FN, see eqs. (2.14) and (2.15).

$$R = \frac{TP}{TP + FN} = \frac{|\text{Correct Objects}|}{|\text{Ground Truth Objects}|} \quad (2.14)$$

$$P = \frac{TP}{TP + FP} = \frac{|\text{Correct Objects}|}{|\text{Predicted Objects}|} \quad (2.15)$$

Both quantities can also be combined into a single metric which is called F_β score or F_1 score if $\beta = 1$ which can be formulated as

$$F_\beta = (1 + \beta^2) \frac{P \cdot R}{(\beta^2 \cdot P) + R} \quad (2.16)$$

$$(2.17)$$

$$F_1 = 2 \frac{P \cdot R}{P + R} = \frac{2TP}{2TP + FP + FN}. \quad (2.18)$$

2.2.4 Ranking Metrics

If the predictions are associated with confidence scores, it is possible to determine the method's performance at all possible working points to capture the full breadth of its capabilities. Ranking-based metrics do not depend on a single confidence threshold but rank the predictions relative to each other to compute the counting statistics at each working point. This is especially useful if general-purpose detection models are being developed without a predefined task which might impose additional requirements. The two most commonly used ranking metrics are introduced now.

mean Average Precision (mAP)

As mentioned earlier, a common combination of metrics to measure object detection performance is precision and recall. It is possible to compute both counting metrics at varying working points by changing the confidence threshold and thereby constructing a precision-recall curve. To compute the final metric, the resulting curve is first smoothed by selecting the highest precision value of all points with lower recall. The monotonically decreasing curve is interpolated at multiple predefined recall values, which yield the precision values \hat{P} . The exact number of interpolation points varies between benchmarks. For example, the MS-COCO [85] benchmark uses 101 points. Averaging the retrieved values results in the mean Average Precision (mAP) for a single localization threshold t_{IoU} , as shown in Equation (2.19).

$$AP_{MS-COCO}(t_{IoU}) = \frac{1}{|r|} \sum_r \hat{P}_{R=r} \quad (2.19)$$

$$r \in \{0, 0.01, \dots, 1\}$$

Additionally, this procedure can be repeated for multiple IoU thresholds and the values can be combined into a single value by computing their mean to capture performance across multiple localisation thresholds.

Free-response Receiver Operating Characteristic (FROC)

Instead of measuring the number of FPs with respect to the number of TPs, it is also possible to relate them to the number of images. This can be expressed in the False Positives per Image (FPPI) value and can be used in combination with the recall to construct the Free-response Receiver Operating Characteristic (FROC). The FROC score is computed by determining the recall at pre-defined FPPI values F and averaging them, see Equation (2.20).

$$\text{FROC} = \frac{1}{|F|} \sum_{f \in F} R_{\text{FPPI}=f} \quad (2.20)$$

2.2.5 Patient-Level Evaluation

In clinical scenarios, it is often necessary to aggregate object-level information to make decisions about patients. The spatial information is irrelevant in these cases, and the evaluation can be performed using patient-level statistics. In contrast to the object-level evaluation, it is also possible to determine patients who were correctly rejected by the algorithms, commonly referred to as TN. This enables the computation of additional counting-based metrics like Specificity (SP), Positive Predictive Value (PPV) and Negative Predictive Value (NPV), as shown in Equation (2.23).

$$SP = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (2.21)$$

$$PPV = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.22)$$

$$NPV = \frac{\text{TN}}{\text{TN} + \text{FN}} \quad (2.23)$$

In addition to the counting metrics, it is also possible to compute patient-level ranking-based metrics to capture all possible working points. The Receiver Operating Curve (ROC) curve computes the sensitivity with respect to the false positive rate, which can also be written as $1 - \text{Specificity}$. To compare methods by means of a single value, it is possible to compute the area under the ROC curve, which is commonly denoted as the Area under the Receiver Operating Curve (AUROC).

CHAPTER 3

Related Work

This chapter presents an overview of related work for medical object detection methods for volumetric data, which is the primary focus of this thesis. It starts by highlighting methods that aim to automate specific tasks in the medical domain, such as mediastinal lesion-, vessel occlusion-, aneurysm-, and lung nodule detection. The second part of this chapter showcases an alternative to this design: the development of methods that are not limited to a single medical task but can be generalized across a broad range of applications.

3.1 Task Specific Design of Medical Object Detection Models

Existing solutions for volumetric object detection tasks in the medical domain focus on individual clinical applications and design task-specific methods. The following sections will highlight the most relevant applications in the scope of the previously defined research questions, see Section 1.2.

3.1.1 Detecting Mediastinal Lesions

The mediastinum, a region located between the lungs, contains multiple vital anatomical structures, including the trachea, oesophagus, nerve pathways, vessel structures and the

heart. Its critical location and the significance of its organs require the timely detection of lesions within the mediastinal area. Despite its clinical relevance, this detection task did not receive significant attention in past studies, likely caused by the absence of publicly available data sets that researchers can easily use. To address the scarcity of publicly data sets for this region, the Mediastinal Lesion Analysis Challenge was hosted at the The Medical Image Computing and Computer Assisted Intervention (MICCAI) 2022 conference. As part of this competition, a data set consisting of 1100 CT scans was established by the organisers, and the training and validation cases consisting of 880 scans are publicly available.

3.1.2 Vessel Occlusion Detection

As outlined in Section 1.2.1, detecting vessel occlusions constitutes a very important problem and automated algorithms can provide additional support in these critical situations. Mechanical thrombectomy is a common treatment for accessible occlusion types but requires precise information about the occlusion’s location, which can be obtained from multiple image modalities like CTA, NCCT and CTP. This thesis specifically includes a study on CTA acquisitions, see Section 1.2.1, since it is a broadly available imaging modality. However, manually screening these scans by clinicians is a process that can be both time-consuming and prone to errors [150]. As a result, multiple research publications as well as commercial solutions exist in this field, which will be presented in the following:

- **Amukotuwa et al. (2019)** [151]: This study outlines the high-level structure of the commercial tool *RAPID-CTA* and its evaluation within a cohort of 926 patients with CTA images. The method comprises many hand-crafted processing steps, including cropping to the head region, co-registration, bone removal, vessel extraction, and hemisphere comparison. Occlusions are determined via manually-defined reference values for vessel segment length or voxel intensities. A second analysis of the performance of *RAPID-CTA* within a cohort of 477 patients can be found in [152].
- **Barman et al. (2019)** [153]: The proposed method is a deep learning-based classification network consisting of inception modules which aim to leverage the symmetries of the brain. A high-dimensional embedding is extracted from the two hemispheres of the brain through a neural network. By directly subtracting the embeddings from each other, the authors aim to attenuate the differences in the representations. The evaluation was conducted on CTA images from 217 subjects. This line of work was continued in [153], which evaluates it on a cohort of 297 patients.
- **Stib et al. (2020)** [154]: Most deep learning methods can only be applied to a sub-volume of the entire 3D scan due to Video Random Access Memory (VRAM) limitations, which constitutes a problem for typical classification tasks. After

extracting the vasculature and cropping the images to the brain region, it is possible to use a Maximum Intensity Projection to project the 3D data onto a 2D plane, which allows the direct application of common classification deep learning models. This study specifically leverages a DenseNet-121 [31]. The data set comprises multiphase CTA, which includes acquiring multiple time points to capture the peak arterial, the peak venous and the late venous phase.

- **Thamm et al. (2020)** [155]: Many methods rely on extracting the vessel tree to compute statistics or incorporate additional prior information. By designing an elaborate processing pipeline consisting of several manually-defined heuristics to remove bone structures, perform registration to a brain atlas and apply multiple filter operations Thamm et al. extract a vessel tree from CTA images. Based on the absence of expected vessel segments, it is possible to infer the presence of vessel occlusions.
- **Olive-Gadea et al. (2020)** [156]: This study evaluates a DenseNet [31] classification system, called *Methinks LVO*, on a cohort of 1453 patients with 823 Large Vessel Occlusion (LVO) cases. In contrast to other studies, NCCT images are used for this method.
- **Dehkharghani et al. (2021)** [157]: The authors evaluate the commercial software *RAPID-LVO* on a cohort of 217 patients with 109 positive findings.
- **Yahav-Dovrat et al. (2021)** [158]: This study evaluates the commercial solution *Viz LVO* and provides a high-level overview of its internal workings. A deep learning-based segmentation approach based on the U-Net [40] architecture is used to segment the vessel structures. Manual heuristics are applied to the extracted vessel tree to obtain information about potential vessel occlusions.
- **Rava et al. (2021)** [159]: The deep learning-based *Canon Auto Stroke* software solution is evaluated on a cohort of 303 patients with 202 LVO cases.
- **Paz et al. (2021)** [160]: 151 consecutive patients with 66 LVOs are evaluated in a clinical setting with the *RAPID-LVO* software. The reported performance of this study is significantly lower than previously conducted evaluations by other studies.
- **Luijten et al. (2022)** [161]: This study evaluates the performance of the *StrokeViewer* software solution on two patient cohorts: (1) the MR CLEAN study consisting of 1110 patients and (2) the PRESTO cohort consisting of 646 patients. A deep learning-based model differentiates between positive and negative cases, and a bounding box is provided in case of a positive finding.
- **Seker et al. (2022)** [162]: The authors evaluate the *e-CTA* solution on a cohort of 627 patients.
- **Thamm et al. (2022)** [163]: After extracting the vessel tree from the image, for example, with the vessel segmentation solution from Thamm et al., it is possible to

determine occlusions without relying on additional image features. This study uses deep learning-based classification networks to classify previously extracted vessel trees and demonstrates the regularization effect of intense elastic deformation in this setting. Applying these transformations to the images directly would lead to surrealistic images due to interpolation artefacts, which are not present when performing the classification on the vessel tree alone. A five-fold cross-validation scheme is utilized to train and evaluate the proposed method on 168 patients.

- **Thamm et al. (2022)** [164]: This work follows up on [163] and proposes another augmentation scheme to increase the diversity in the training data set. After splitting the vessel tree into multiple sub-trees, it is possible to recombine trees from different patients to increase the variability of the data set. The improvements of this augmentation scheme are demonstrated in a cohort with 151 scans.
- **Kassam et al. (2022)** [165]: Instead of relying on CNN architectures, which are specifically designed for images, this study presents the use of a Graph Neural Network to classify vessel trees. To avoid overfitting, augmentation is performed on the feature level, with the help of noise, and on the vessel level using the recombination approach presented in [164]. The performance of this approach is similar to CNN based networks, but the required computational resources are drastically reduced.

In summary, previous work heavily relies on elaborate pre-processing steps to extract vessel information, use symmetry information between the left and right hemispheres and perform cropping operations to remove undesired anatomical regions. Consequently, these approaches rely on a wide range of hand-crafted heuristics to function properly and severely limit the positions of detectable occlusions. Several commercial tools exist for detecting LVOs, but information on their internal design is sparse, and performance numbers vary between studies and evaluation protocols. This makes a direct comparison of these solutions difficult. An overview of all presented approaches is shown in Table 3.1.

3.1.3 Aneurysm Detection

Aneurysms represent a protrusion of vessel structures and can be located near the brain region. Unruptured Intracranial Aneurysms (UIAs) can spontaneously rupture, causing a Subarachnoid Haemorrhage (SAH) potentially leading to death or severe disabilities [167, 168]. CTA or Time-of-Flight MR-Angiography (TOF-MRA) scans can be used to assess the risk or decide on follow-up treatment. Both imaging modalities consist of a large number of axial slices which need to be manually examined by trained personnel. Automating this cumbersome process can alleviate errors and reduce diagnostic times, which benefits patients.

Table 3.1: Overview of existing literature on vessel occlusion detection. Cohort size refers to the number of used scans and the positive cases only include patient scans which are considered as positives by the study. The method column showcases the name of the software solution. Only a single study provides a public code release.

Name	Year	Cohort Size	Positive Cases	Method	Public Code
Amukotuwa et al. [151]	2019	926	395	RAPID-CTA	
Amukotuwa et al. [152]	2019	477	106	RAPID-CTA	
Sheth et al. [166]	2019	297	224	custom	
Barman et al. [153]	2019	217	94	custom	
Stib et al. [154]	2020	540	270	custom	✓
Thamm et al. [155]	2020	-	-	custom	
Olive-Gadea et al. [156]	2020	1453	823	Methinks LVO	
Dehkharghani et al. [157]	2021	217	109	RAPID-LVO	
Yahav-Dovrat et al. [158]	2021	1167	75	Viz LVO	
Rava et al. [159]	2021	303	202	Canon Auto Stroke	
Paz et al. [160]	2021	151	66	RAPID-LVO	
Luijten et al. [161]	2022	1110 / 646	1110 / 141	StrokeViewer	
Seker et al. [162]	2022	301	140	e-CTA	
Thamm et al. [163]	2022	168	109	custom	
Thamm et al. [164]	2022	151	-	custom	
Kassam et al. [165]	2022	151	-	custom	

The Aneurysm Detection And segMentation Challenge (ADAM) challenge is an international competition presented at MICCAI 2020. It consisted of two tasks: one for aneurysm detection and one for aneurysm segmentation. As part of the challenge, a public data set was released containing 113 training cases with 129 UIAs. Treated aneurysms and artefacts associated with the treatment were annotated but not considered during the evaluation. The first place of the detection track was won by a preliminary version of our method presented in Section 4.2 which is part of the second research question introduced in Section 1.2.2. In contrast to other submissions to the challenge, this approach was specifically tailored towards detecting objects rather than using voxel-level methods with post-processing.

Noto et al. (2023) [18] released a second data set with annotated aneurysms containing 284 subjects with 198 aneurysms. An ablation study was conducted on the capabilities of handling spherical labels since these labels were manually generated faster. They propose to formulate the aneurysm detection task via semantic segmentation using a 3D U-Net [40] architecture. Since aneurysms can only be located at vessel structures, a new data-loading strategy is developed to incorporate this anatomical bias into the training.

This task-specific design decision can also be found in work from Assis et al. [169] who propose a one-stage anchor-free object detection model to produce spherical predictions.

The patches to train the model are specifically extracted from regions containing vessel structures to alleviate the data scarcity issue.

Ceballos-Arroyo et al. [83] opt to encode the vessel information directly into the model. The detector is based on the Deformable DETR architecture, using a positional embedding of the image. The vessel information is encoded into a distance map and added to the embedding to incentivize predictions near the vessel structure. This study uses a publicly available CTA data set [170] with aneurysm annotations.

In conclusion, previous studies focus heavily on designing task-specific models which can leverage prior information about the vessel tree to improve their predictions. Multiple public TOF-MRA and one public CTA data set are available to compare model performance on this clinically important task.

3.1.4 Lung Nodule Detection

The highest number of cancer-related deaths worldwide is caused by lung cancer [171]. Low dose CT offers the possibility to detect lung cancer in earlier stages with better treatment options and is subsequently a potential tool to reduce mortality from lung cancer [172]. However this requires the assessment of an increased number of scans, ultimately increasing the workload of clinicians. Fueled by the clinical relevance and the need to provide automated detection systems to reduce assessment times for clinicians, many methods were developed to detect lung nodules in CT images. This is also represented by the availability of multiple large-scale benchmarking data sets that are used to compare state-of-the-art methods. The ANODE09 [173] study marks one of the earlier benchmarks, which compares multiple systems on a standardized data set of 55 scans. The public availability of the LIDC [13] data set consisting of 1018 cases paved the way for a second benchmark called LUNA16 [43]. This data set represents a subset of LIDC [13], consisting of 888 images and was used by several studies over the years. Most recently, the PN9 [14] data set was released, which contains over 8,000 CT scans and 40,000 annotated nodules. Newer works have started evaluating their methods on this challenging data set. An overview of current lung nodule detection methods on LIDC [13], LUNA16 [43] and PN9 [14] is provided in Table 3.2 and the individual methods are shortly explained in the following paragraphs.

- **Dou et al.(2017)** [65]: A two-stage pipeline is proposed, which first generates many candidate locations, and these are subsequently filtered by a FPR stage. First, a CNN is trained on small patches to differentiate between background and foreground. Inspired by HNM approaches from detection networks, the classification network forward propagates a large number of patches and only propagates 50% of the samples with the highest loss back. The trained network is applied to the entire CT scan in a fully convolutional way to extract a scoring map for nodule locations. A

Table 3.2: Overview of existing literature on lung nodule detection on LIDC [13], LUNA16 [43] and PN9 [14]. The detector type describes the internal design of the proposed detection method. Methods consisting of classification modules that do not offer additional regression capabilities were categorised as "Other" since they do not adhere to the typical definition of a detection method that produces proposal boxes. The False Positive Reduction (FPR) stage only includes methods which train a separate classification network, which is not integrated into the training of the detection model. The second stage of the R-CNN model is sometimes also referred to as a FPR but is not included in this definition.

Name	Year	Training Data	Detector Type	FPR Stage	Public Code
Dou et al. [65]	2017	LUNA16	Other	✓	
Ding et al. [66]	2017	LUNA16	Two-Stage	✓	
Zhu et al. [75]	2018	LUNA16, LIDC	Two-Stage	✗	✓
Wang et al. [67]	2018	LUNA16	One-Stage	✓	
Khosravan and Bagci [76]	2018	LUNA16	Other	✗	
Liao et al. [78]	2019	LUNA16, DSB17	One-Stage	✗	✓
Liu et al. [73]	2019	LUNA16	One-Stage	✓	
Tang et al. [77]	2019	LIDC	Two-Stage	✗	✓
Cao et al. [64]	2020	LUNA16	Seg.	✓	
Li and Fan [72]	2020	LUNA16, LIDC	One-Stage	✗	✓
Jaeger et al. [46]	2020	LIDC	One-Stage	✗	✓
Gong et al. [174]	2020	LUNA16	One-Stage	✗	
Song et al. [74]	2020	LUNA16	One-Stage	✗	✓
Mei et al. [70]	2021	LUNA16, PN9	Two-Stage	✗	✓
Luo et al. [69]	2022	LUNA16	One-Stage	✗	✓
Xu et al. [71]	2022	PN9	Two-Stage	✗	✓
Harsono et al. [79]	2022	LIDC	One-Stage	✗	
Lu et al. [68]	2023	LUNA16	Two-Stage	✗	

CNN with residual connections is used in the second stage to classify and regress larger patches, which are extracted from the initially proposed nodule locations.

- **Ding et al. (2017)** [66]: A 2.5D Faster R-CNN [36] approach is used to generate initial nodule candidates from a stack of axial CT slices. To improve the detection performance for small nodules, an additional transposed convolution is added at the end of the backbone to increase the resolution of the feature map used for the detection task. This approach yields a high recall but still includes many FP. A 3D CNN is used to incorporate additional context and classify the candidates in a separate FPR stage.
- **Zhu et al. (2018)** [75]: *DeepLung* is based on the Faster R-CNN [36] architecture and incorporates dual path [175] blocks in its design. To improve the performance for small nodules, a U-Net [40] like encoder-decoder design is used to up-sample the low-level features before using them for the detection task. The outputs are

spheres that reduce the regression problem to four coordinates (three for the centre and one for the radius). Candidate nodules are subsequently categorized into benign and malignant by a separate classification network.

- **Wang et al. (2018)** [67]: The detector design is based on a 2.5D version of the Faster R-CNN architecture, where three consecutive slices are used as the input. A FPN predicts objects across multiple scales with a single network. Conditional 3D NMS is introduced to suppress similar predictions across different slices. The generated nodule candidates are processed by a 3D CNN with two branches to reduce the number of false positive predictions. The first branch generates a Gaussian heatmap around the nodule, which is used as an additional input channel for the second branch, that is responsible for the binary classification.
- **Khosravan and Bagci (2018)** [76]: *S4ND* redefines the detection task as a cell-wise classification task. The image is divided into a grid of cells and each cell is classified by the CNN. A cell is considered positive if a nodule is located in it and a weighted binary cross-entropy loss is used to counteract the label imbalance.
- **Liao et al. (2019)** [78]: This approach was originally designed for the Data Science Bowl (DSB) 2017 [176] to differentiate between patients with and without lung cancer. The proposed 3D detector is inspired by the U-Net [40] architecture, which progressively up-samples low-level features. The detection is performed on the highest-resolution feature map. Additional location information is inserted by providing an additional location crop, which encodes normalized coordinates. Since the DSB 2017 data set is only annotated with patient-level information, a classification network is trained in the second stage via Multiple Instance Learning.
- **Liu et al. (2019)** [73]: A two-stage pipeline is proposed consisting of a 3DFPN nodule detector and a HS^2 network to reduce false positive predictions. The detector follows a one-stage detector design and uses a spherical representation for its predictions. Each potential nodule location is represented by a Location History Image [73] and processed by a separate classification network.
- **Tang et al. (2019)** [77]: *NoduleNet* is an end-to-end network to perform nodule detection, classification and segmentation in a single framework. The detector design is built on a 3D version of the Faster R-CNN [36] architecture and uses RoI Pooling to extract features from multiple levels of the backbone. The segmentation module progressively up-samples cropped features and recombines them with fine-grained features.
- **Cao et al. (2020)** [64]: A segmentation-based approach based on the U-Net [40] architecture generates an initial set of region proposals. The CNN utilises residual and dense connections to improve the gradient flow. A novel patch sampling strategy is introduced to improve detection performance. SE-ResNet [177], DenseNet [31] and

InceptionNet [178] are used in an ensemble to build the FPR stage of the proposed method.

- **Li and Fan (2020)** [72]: *DeepSEED* uses a one-stage detector design with an encoder-decoder scheme as its feature extractor. Squeeze-and-excitation modules are inserted to improve the feature extraction process. The predictions use a spherical design to reduce the regression to four coordinates, and a dynamically scaled CE loss is used to reduce false positive predictions.
- **Jaeger et al. (2020)** [46]: This work introduces the Retina U-Net architecture, which extends the design of the Retina Net [35] model to leverage additional semantic segmentation supervision. The baseline models include the Retina Net architecture, the Mask R-CNN [37] architecture and a segmentation-based approach. Furthermore, extensive ablation experiments are conducted with 2D, 2.5D and 3D versions of the networks on the LIDC [13] and a private breast diffusion MRI data set [46]. The code is publicly released as part of the Medical Detection Toolkit (MDT).
- **Gong et al. (2020)** [174]: Many detection networks use an anchor-based design philosophy, which depends on carefully adjusted anchor sizes. Centre point-based approaches like CenterNet [179] allow the design of detection networks which do not rely on anchors. A 3D design of this approach, in combination with a feature aggregation module, was used to detect lung nodules in [174]. To reduce the number of false positive predictions, a FPR is utilized in combination with Motion-History Images.
- **Song et al. (2020)** [74]: This one-stage detection method uses a centre-based detection approach to alleviate the need for manually designed anchors. The proposed network architecture includes additional coordinate maps and Squeeze-and-Excitation [177] modules. Matching between centre points and ground truth objects is performed via distance maps where the top k nearest points are considered positive. Only a subset of the negative points are used for the loss computation, which is determined via Adaptive Points Mining (APM), and the remaining ones are ignored. The classification loss is computed via a modified version of the focal loss called re-focal loss.
- **Mei et al. (2021)** [70]: *SANet* is based on a U-Net [40] like encoder-decoder architecture to build a 3D RPN. The prediction is only performed on the highest-resolution feature map. The slice-grouped non-local module is introduced to enhance representation, which extends across multiple slices. The detection head uses information which was cropped from multiple resolution stages to classify the proposals from the RPN.
- **Luo et al. (2022)** [69]: *SCPM-Net* builds upon *CPM-Net* [74] to propose a centre-based lung nodule detection network. Instead of producing bounding boxes, the

predictions are represented by bounding spheres, which aim to provide a better representation of spherical lung nodules. A sphere-based IoU (SIoU) measure is derived to directly measure the similarity between two spheres. Combined with additional geometric priors, multiple loss functions can be derived, which are extensively analyzed in the presented ablation experiments.

- **Xu et al. (2022)** [71]: This work builds upon the design of Mei et al. but introduces the long short slice grouping (LSSG) design to enhance the feature extraction process. The new method is called *LSSANet*.
- **Harsono et al. (2022)** [79]: *I3DR-Net* is a one-stage detector based on the Retina Net [35] design. It uses a 3D weight-inflated initialization for its backbone network to leverage ImageNet [94] pre-training. A modified version of the FPN is used to reduce its memory footprint and use finer features for the detection.
- **Lu et al. (2023)** [68]: *FFNET* uses a 3D FPN to predict nodules across multiple scales with a single network. Features from a single resolution from the encoder and decoder are used for a filter network, which is applied to low-scoring nodule proposals. High-scoring proposals are not further processed.

The early availability of public data sets has fueled the research on lung nodule detection. LUNA16 [43] represents the most widely used benchmark, and the recently introduced PN9 [14] data set might become the next major benchmark due to its size and difficulty. Two-stage and one-stage detection models are actively used to detect lung nodules throughout the literature. All included publications except one (Jaeger et al. [46]) evaluate their method on a single pathology and do not consider other detection tasks. Furthermore, task-specific priors such as lung segmentations or spherical representations are introduced to improve their detection performance.

3.1.5 Detection with Detection Transformers in the Medical Domain

The DETR model (see section 2.1.3) popularised a new way to formulate detection tasks as a direct set prediction problem. However, the initial design suffered from several shortcomings: the required training times significantly exceeded the training times of anchor-based detectors, and the performance on small objects lagged behind. Follow-up studies continued to address individual shortcomings by using sparse attention mechanisms in the form of deformable attention [147], introducing prior information in the form of dynamic anchor boxes [60] and proposing additional auxiliary tasks like denoising [59]. The architecture that finally demonstrated state-of-the-art performance on the MS-COCO [85] benchmark is the DINO DETR [39] architecture.

Despite this success, the adoption of DETR models in the medical domain is slow. Potential reasons for this include (1) the medical domain contains much smaller data sets than

its natural imaging counterpart, and transformers are notorious for requiring a lot of training data; (2) medical tasks encapsulate a broad range of tasks, each having its own composition of image and label properties and (3) most approaches are designed for 2D images and adapting them to volumetric 3D data poses an additional overhead and requires careful design of the entire pipeline.

One of the first works on the DETR architecture for detection problems in the medical domain was established by Wittmann et al. [180]. The study evaluated multiple architectures for organ detection and proposed its own model to incorporate prior information about the location of anatomical structures in images. The focus of this study was on organ detection, which emphasizes fine delineations of the organ structures rather than requiring decisions about the existence of objects. In conclusion, this new methodological innovation to detect pathologies in volumetric images has not been adopted, and further research is needed to evaluate the feasibility of these models and determine important design aspects.

3.2 Self-configuring Model Design for Medical Applications

Task-specific methods only incorporate a single data set into their design, often aiming to incorporate more powerful components in the form of architectural modules [70, 71], output representations [69] or other training strategies. A detailed analysis of the KiTS19 [181] challenge, however, revealed that the impact on performance from the careful configuration of hyperparameters is more important than architectural changes, which are often found as novel contributions. Determining these parameters constitutes an iterative tuning process which requires large computational resources and expert knowledge. As a result, the performance of the same high-level method can achieve vastly different results on a task.

As elaborated in the introduction of this thesis (see Chapter 1), automated solutions for medical image analysis are required for many different tasks, spanning across different anatomical regions, target structures and imaging modalities. Each application has its own composition of acquisition parameters, which clinicians determine to capture the most relevant information for the current situation. Typical decisions include selecting a suitable acquisition modality, selecting an appropriate slice thickness, and only capturing relevant regions. This aggravates the configuring problem since it needs to be repeated for every new data set.

nnU-Net is the first method to systematically approach the configuration process of deep learning-based semantic segmentation methods. Its application does not require specialized knowledge of the underlying technology since it automatically determines its

hyperparameters and can be applied without manual intervention. Data set properties like spacing and modality are summarised in the Data Fingerprint, which is used as the basis for the automated configuration process. All hyperparameters are divided into three groups:

- **Fixed Parameters** are kept constant across all tasks since they are robust to changes in the Data Fingerprint. These include the Model Blueprint, Optimizer, training- and inference pipeline.
- **Rule-based Parameters** are adapted between tasks based on the properties from the Data Fingerprint. They include strategies for the resampling procedures, determining the network topology, patch- and batch-size for training, and adding a cascaded model.
- **Empirical Parameters** must be adapted between data sets but can not be determined from the Data Fingerprint. They are empirically determined by observing the validation performance and include parameters for post-processing and model selection.

It is not sufficient to examine a single task to derive this categorization and determine the correct mechanisms for adapting the hyperparameters. nnU-Net [47] proposes a new development paradigm, by using information from the 10 tasks of the Medical Segmentation Decathlon (MSD) [16] it aims to generate a robust configuration procedure for arbitrary semantic segmentation tasks. To evaluate its generalization capabilities a separate pool of data sets is needed to evaluate the performance on previously unseen data sets. When nnU-Net [47] is applied to international challenges, it outperforms most existing solutions, including many task-specific methods.

In summary, nnU-Net [47] establishes a new development paradigm for semantic segmentation methods by systematically approaching the configuration process. However, its design decisions are limited to semantic segmentation methods involving only voxel-level information. The prevailing development paradigm in the medical detection domain is still limited to individual tasks, and no self-configuring methods or diverse pool of data sets are available to establish such methods.

CHAPTER 4

Materials and Methods

This chapter introduces two different concepts for designing medical object detection methods: first, the manual design of detection methods is introduced, which includes task-specific modifications to address individual shortcomings of the baseline model. These modifications build the foundation of RQ1 (see Section 1.2.1) and provide insights into the design of highly effective detection methods in international challenges and clinical settings.

Second, the design of the first self-configuring medical object detection method, named nnDetection, is presented, marking the beginning of a new development paradigm for medical detection methods (see RQ2 in Section 1.2.2). Instead of developing a task-specific model, it follows the design principles of nnU-Net [47] to build a generalising method based on rule-based, fixed and empirical parameters to automate the configuration process of one-stage, two-stage and direct set prediction models. The resulting method, can handle various annotation types and is able to generalise across different image modalities, anatomical regions and object structures.

The introduced methods will build the basis for the results in Chapter 5 and address the methodological challenges arising from the research questions in Section 1.2. The design of manual detection pipelines is based on [182, 183, 184]. The design of self-configuring medical object detection methods is based on [185, 82].

4.1 Task Specific Design of Object Detection Methods

This section outlines three studies where we developed manual pipelines to address medical detection problems. Section 4.1.1 presents a detailed approach for detecting mediastinal lesions in CT images. This method was part of the MELA challenge 2022, where it ranked third in the competition. The challenge data set and methodological aspects are presented in Section 4.1.1. Section 4.1.2 introduces our clinical study targeting the detection of vessel occlusions. An overview of the three utilized cohorts is described first. Afterwards, the proposed single-stage object detection pipeline is presented. The manual configuration of DETR models is described in Section 4.1.3.

4.1.1 Detecting Mediastinal Lesions in CT Images

Disclosure of this work

This section includes portions of our work that has been published in:

Baumgartner, M., Full, P.M., Maier-Hein, K.H. (2023). "Accurate Detection of Mediastinal Lesions with nnDetection". In: *Xiao, Y., Yang, G., Song, S. (eds) Lesion Segmentation in Surgical and Diagnostic Applications. CuRIOUS KiPA MELA 2022 2022 2022. Lecture Notes in Computer Science, vol 13648*. Springer, Cham.

Data Set Analysis

Background: The provided data set comprises 1100 chest CT images (770 training, 110 validation, 220 testing), acquired using Somatom Definition AS (Siemens Medical Systems, Germany) or Brilliance 40 (Philips Medical Systems, Netherlands) scanners at the Shanghai Pulmonary Hospital affiliated with the Tongji University. Multiple expert clinicians annotated lesions in the mediastinum, resulting in 884 annotated lesions in the training and validation split. All lesions are annotated via bounding box labels rather than voxel-wise delineation.

Preprocessing: To inject additional prior knowledge into the training by enabling voxel-wise supervision, the provided annotations are converted into spherical object annotations as depicted in Figure 4.1. Prior work such as [80] and [186] have shown improved results by including segmentations as an additional auxiliary task during training even when artificially generated from coarser annotation styles like bounding boxes. This also enables the usage of publicly available data augmentation frameworks, like batchgenerators [187], which are built around segmentation maps and do not support bounding box annotations.

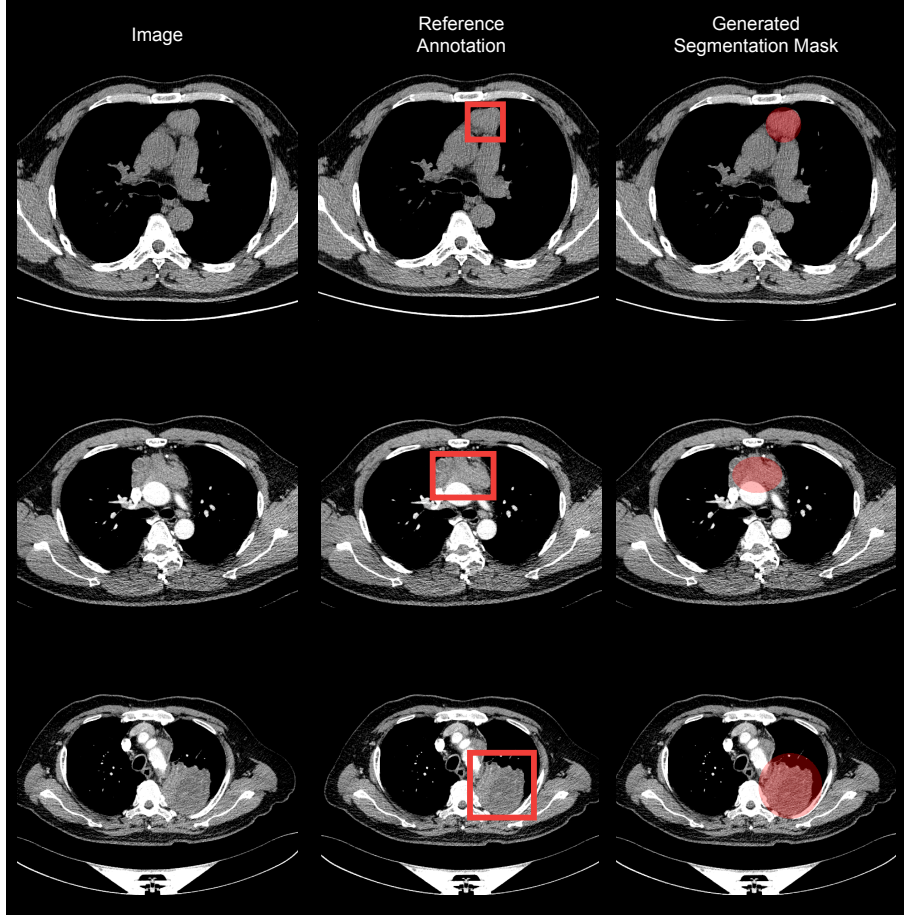


Figure 4.1: Visualisation of the bounding box and the converted spherical segmentation annotations for the MELA data set. The first column shows CT images from the data set, and the second column shows the annotated bounding boxes as an overlay. The converted spherical segmentations are shown in the third column. All images and annotations were visualised via MITK [188].

The preprocessing of the CT images is subdivided into two steps: first, the intensity values of the image voxels are normalised by clipping them to the $[-901.0, 554.0]$ Hounsfield Unit (HU) range, followed by z-score normalisation. The mean of -11.77 HU and standard deviation of 242.75 HU for the intensity normalisation are computed across the entire training data set. Next, all images are resampled to the selected target spacing. The MELA data set contains large objects which do not fit into typical three-dimensional patches due to VRAM constraints of current hardware. This can lead to duplicate predictions during the inference phase of the model, which count as FP predictions during evaluation. Consequently, the target spacing is set to $[1.40, 1.43, 1.43]$ mm, which is two times the median spacing of the data set to downscale the images. We only resample those scans where the spacing varies by more than 5% from the determined target spacing. The images

are resampled with third-order B-spline interpolation, while the labels are resampled with nearest neighbour interpolation.

Methodological Design

Our method for detecting mediastinal lesions in CT images employs a single-stage object detection model called Retina U-Net [46]. This model is simple yet effective in predicting bounding boxes from pre-defined anchors. Additionally, semantic segmentation supervision can be incorporated during the training process.

Training: All models are trained for 50 epochs, each comprising 2500 batches. Optimization is performed using Stochastic Gradient Descent (SGD) with Nesterov momentum set at 0.99. The baseline model is trained using a batch size of six, whereas all subsequent models employ a batch size of eight. Each batch is constructed such that half contains at least one foreground object, with the remaining half being sampled randomly. To make the model more robust during inference, an additional random offset is applied to patches which are forced to contain a foreground object. The offset is applied to each dimension where the object size does not exceed 70% of the patch size. It is determined such that the object does not exceed the patch boundaries. If the object size exceeds the patch size, a random centre point within the bounding box of the object is selected as the patch centre.

Network Topology: The selected network blueprint follows the architectural model design of Retina U-Net [46]. The encoder utilizes a series of stacked convolutions, each followed by instance normalisation [189] and a Leaky Rectified Linear Unit (LReLU) non-linear activation function. The decoder architecture is based on the FPN [117], but instead of employing linear interpolation, transposed convolutions [103] are used for upsampling. The detection head, constructed from multiple convolutional layers with group normalization [190] and LReLU activation functions, is shared across the last four resolution levels.

The architecture of the head includes two branches: one for anchor classification, trained using BCE loss, and another for anchor regression, trained with a weighted L1 loss. At the highest resolution level, an additional semantic segmentation head is employed to leverage voxel-wise supervision. A combination of the CE loss and Dice loss [191] is used as the segmentation loss. The total training loss is defined as follows:

$$L_{total} = L_{BCE} + 2 \cdot L_{L1} + L_{CE_seg} + L_{Dice_seg} \quad (4.1)$$

Considering the large data set size, the model's capacity has been increased by setting the initial number of channels to 48 (from 32), doubling in each level until a maximum number of 384 channels is reached. The resulting network architecture is illustrated in Figure 4.2.

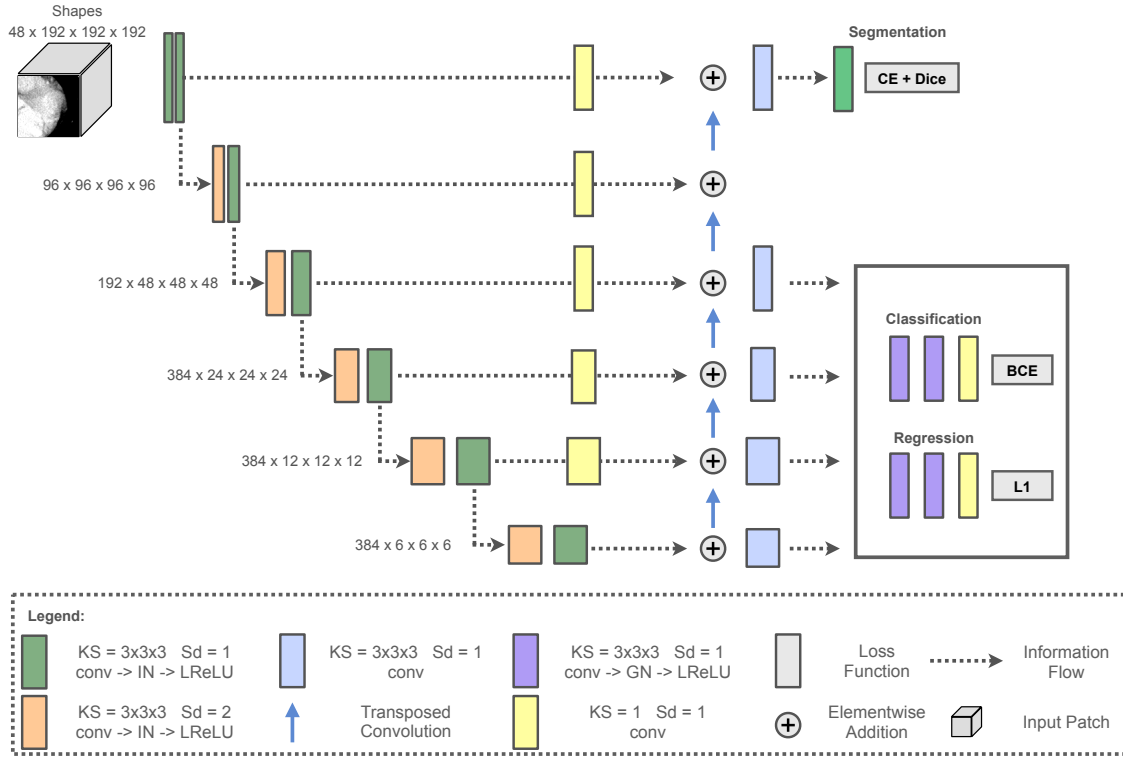


Figure 4.2: Visualisation of the Retina U-Net architecture for the MELA data set. The model can be divided into four parts (1) the encoder consisting of consecutive down-sampling layers and convolutions depicted in orange and green (2) a FPN as the decoder to combine information from the encoder and the upsampled feature maps (3) a shared detection head to predict bounding box classification and regression targets and (4) a semantic segmentation head. This figure is adapted from [182].

Large Patch Size (LP): During inference, when object sizes extend beyond the patch size, the same object can be predicted multiple times due to overlapping patches. Merging these predictions can be difficult since the IoU between the predictions is lower than for objects which are fully contained inside the patches. This phenomenon is referred to as stitching artefacts in this thesis since the correct predictions of the network are not correctly combined during post-processing. While the increased target spacing results in reduced object sizes, commonly used patch sizes of $[128, 128, 128]$ remain insufficient for encapsulating sufficient contextual information during the training and testing phase. To fully leverage the available computational hardware resources, training was executed on two NVIDIA A100 (40GB) GPUs, with a batch size of four per GPU and a large patch size of $[192, 192, 192]$.

Reduced Rotation in Augmentation (Aug B): To mitigate overfitting and improve the generalization of our methods, extensive spatial data augmentations are utilised. These augmentations increase the diversity of the data set by training on augmented versions

of the same information, specifically through rotation and scaling of the images. When utilizing coarse annotations, such as bounding boxes or ellipsoids, rotations which are not multiples of 90 degrees can introduce ambiguous boundaries, leading to localization errors during training. This is crucial in the context of the MELA challenge, where the IoU threshold is set to 0.3, necessitating precise localization for optimal model performance.

To address this problem, a second augmentation pipeline with a reduced rotation magnitude is used to train an additional set of models. To maintain a diverse augmentation strategy, this pipeline also includes intensity-based augmentations and additional spatial augmentations, such as the transpose operation and additional rotations of 90 degrees. An overview of the entire augmentation pipeline is illustrated in Table 4.1.

Table 4.1: Shows two different augment schemes for the MELA data set. The left column describes the augmentation operation, which is executed with the batchgenerators [187] framework. p refers to the probability of being applied to a single sample, and \times indicates the absence of a transform. The magnitude of a transformation is described by m . The columns in the middle and on the right show two different augmentation schemes. Table reproduced from [182].

Augmentation	Baseline (Aug A)	Reduced Rotation (Aug B)
Elastic Deformation	\times	\times
Rotation (m in degrees)	$p=0.3$ $m=[-30, 30]$	$p=0.1$ $m=[-10, 10]$
Rotation 90	\times	$p=0.5$
Transpose Axes	\times	$p=0.5$
Gaussian Noise	$p=0.1$	$p=0.1$
Gaussian Blur	$p=0.2$	$p=0.2$
Median Filter	\times	$p=0.2$
Multiplicative Brightness	$p=0.15$	\times
Brightness Gradient	\times	$p=0.3$
Contrast	$p=0.15$	$p=0.2$
Simulate Low Resolution	\times	$p=0.15$
Gamma	$p=0.3$	$p=0.1$
Inverse Gamma	$p=0.1$	$p=0.1$
Local Gamma	\times	$p=0.3$
Sharpening	\times	$p=0.2$
Mirror (per axes)	$p=0.5$	$p=0.5$

Final submission: The organisers of the MELA challenge allowed multiple submissions to the test leaderboard. Based on the cross-validation experiments, we submitted four models in total. The baseline model (M1), a model trained with large patch size (M2) and a model trained with Aug B and the large patch size (M3). The best submission was achieved by ensembling the predictions of M2 and M3 with Weighted Box Clustering (WBC) at an IoU threshold of 0.2.

4.1.2 Detecting Vessel Occlusions in CTA Images

Disclosure of this work

This section includes portions of our work that has been published in:

Brugnara, G.^{*}, Baumgartner, M.^{*}, Scholze, E. D.^{*}, Deike-Hofmann, K., Kades, K., Scherer, J., ... & Vollmuth, P. (2023). "Deep-learning based detection of vessel occlusions on CT-angiography in patients with suspected acute ischemic stroke." *Nature Communications*, 14(1), 4938.

^{*} contributed equally

Data Set Analysis

This study uses one internal and two external cohorts. The **University Clinic Heidelberg (UKHD)** cohort, consisting of 1179 patients with 800 patients having at least one vessel occlusion, was used to train and evaluate the developed model on in-distribution images. All CTA scans within this cohort were acquired using Siemens scanners, with the majority (1136/1179) being obtained with a *Siemens SOMATOM Definition AS* scanner. 1128 of the 1179 scans were reconstructed using either the B26f or I30f reconstruction kernel, and the median slice thickness was 0.75 mm (IQR [0.75–0.75]). A total of 835 patients were utilized for training, while 344 were reserved for the internal test set. The test set was artificially balanced to contain the same number of control and occlusion-positive patients. Among the test set scans, 75% (258/344) were obtained during the early arterial phase, where most of the contrast is present in the arteries. In summary, the Heidelberg cohort represents a large but homogeneous data set, as detailed in Tables B.1 and B.2.

The first external test set, represented by the **FAST** cohort, was collected from the regional stroke consortium Rhine-Neckar and encompasses three hospitals. This cohort was gathered in a pseudo-prospective manner, resulting in a lower prevalence of vessel occlusions than the internal test set (52/327 patients with vessel occlusions). The FAST cohort includes CTA scans acquired on various scanners with different reconstruction kernels, including 167 out of 327 scans obtained using the *Siemens Sensation 40* model, a scanner not present in the Heidelberg cohort. Furthermore, a significant proportion of the scans were performed in the Peak Arterial (41%) or Equilibrium (35%) phases, showing a substantial distribution shift from the UKHD test set. The median slice thickness in this cohort was 1 mm (IQR [1.00–1.00]), with additional details available in Tables B.1 and B.2.

The **University Clinic Bonn (UKB)** cohort, which represents the second external test set, was collected from the University Clinic Bonn. All CTA scans in this cohort were acquired using the *Philips IQon – Spectral CT*. Most scans depict the Peak Arterial and

Peak Venous phases. The median slice thickness was 1 mm (IQR [1.00–1.00]), with further details provided in Tables B.1 and B.2.

All visible vessel occlusions are annotated in the CTA scans with spherical markers, with a diameter of 30 voxels for LVOs or 15 voxels for Medium Vessel Occlusions (MeVOs). The annotations include both treated occlusion and incidental findings and were created with the assistance of the radiological reports. Additionally, High-grade Stenosis (HGS) were annotated in the two external cohorts. The centre of each sphere was positioned at the most proximal point of contrast loss. Each occlusion is assigned to one of three groups:

- **Anterior LVOs** which include occlusions in the common carotid artery (CCA), internal carotid artery (ICA), the M1-segment of the middle cerebral artery (MCA) and A1-segment of the anterior cerebral artery (ACA) [192, 183]
- **Anterior MeVOs** which are located in the M2-/M3-segment of the MCA or the A2-/A3-segment of the MCA [193, 183]
- **Posterior VO**s which includes LVOs in the vertebral artery (VA), basilar artery (BA) and the P1-segment of the posterior cerebral artery (PCA) and MeVOs of the P2/3-segment of the PCA [192, 193, 183]

Methodological Design

The proposed one-stage detection model, called HD-CTA, follows a three-step process to predict new images. First, all input images are preprocessed, including resampling to the same image spacing and normalizing intensity values. Second, five anchor-based Retina Net [35] detectors, trained using a cross-validation scheme, are applied to the preprocessed images. Finally, the predictions generated by these detectors are postprocessed and aggregated to produce the final detection results. An overview of this pipeline is illustrated in Figure 4.3.

Preprocessing: All images were resampled to the median voxel spacing of the training set (0.5, mm \times 0.453, mm \times 0.453, mm). Intensity values are clipped to $[-148.0, 988.0]$ and subsequently normalized using z-score normalisation. Since voxel intensities in CTA images represent HU, which are measured on an absolute scale, the mean and standard deviation are computed across the entire data set.

Model: The Retina Net model comprises three primary components: an encoder network, a decoder architecture, and a detection head.

The encoder network includes six resolution levels, each comprising two convolutional blocks. Each block consists of a convolution, an instance normalization layer [189], and a LReLU non-linear activation function. The first convolution in each level has a stride greater than one to represent a pooling operation and reduce the spatial dimensions of

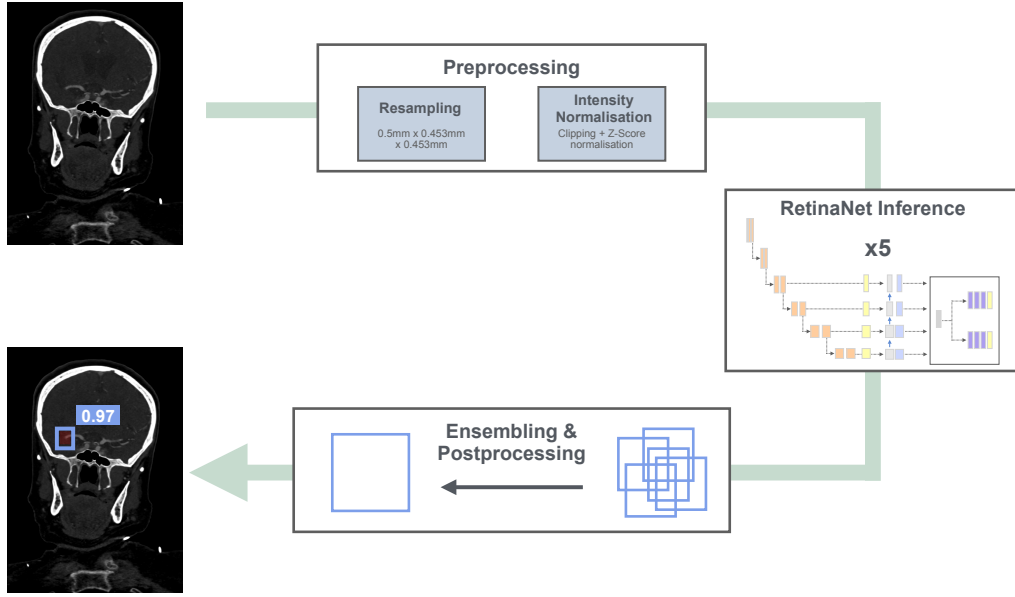


Figure 4.3: Inference Procedure for Vessel Occlusion Detection. First, the images are preprocessed by resampling them to the same spacing and normalizing the voxel intensity values. Afterwards, five Retina Net one-stage detectors are executed to produce a set of predictions. The predictions are ensembled via WBC to produce the final result. This figure is adapted from [183].

the feature maps. The initial level begins with 32 channels, with the number of channels doubling at each subsequent level, up to a maximum of 320 channels.

The decoder architecture follows the design of a FPN [117] to progressively upsample low-resolution features. These features are combined via elementwise addition with the corresponding encoder features, which were processed by a $1 \times 1 \times 1$ convolution to adjust the number of channels. The decoder spans the last four resolution levels.

Retina Net [35] operates as an anchor-based detection model, where detection is performed by regressing and classifying predefined bounding boxes (anchors), which are densely distributed across the image. During training, anchors are assigned to ground truth objects via ATSS to compute the loss function. This enables the dynamic selection of IoU thresholds to divide anchors into positive and negative examples. The classification branch is trained using focal loss [35], while the regression branch is only trained on positive anchors using the smooth L1 loss [101]. An overview of the architecture is provided in Figure 4.4.

Since the occlusion annotations are limited to two predefined sizes, the network is trained with two anchor sizes of $[8, 10, 10]$ and $[15, 14, 14]$ at the first level used for the detection. Deep levels scale these sizes via their relative stride to the first level.

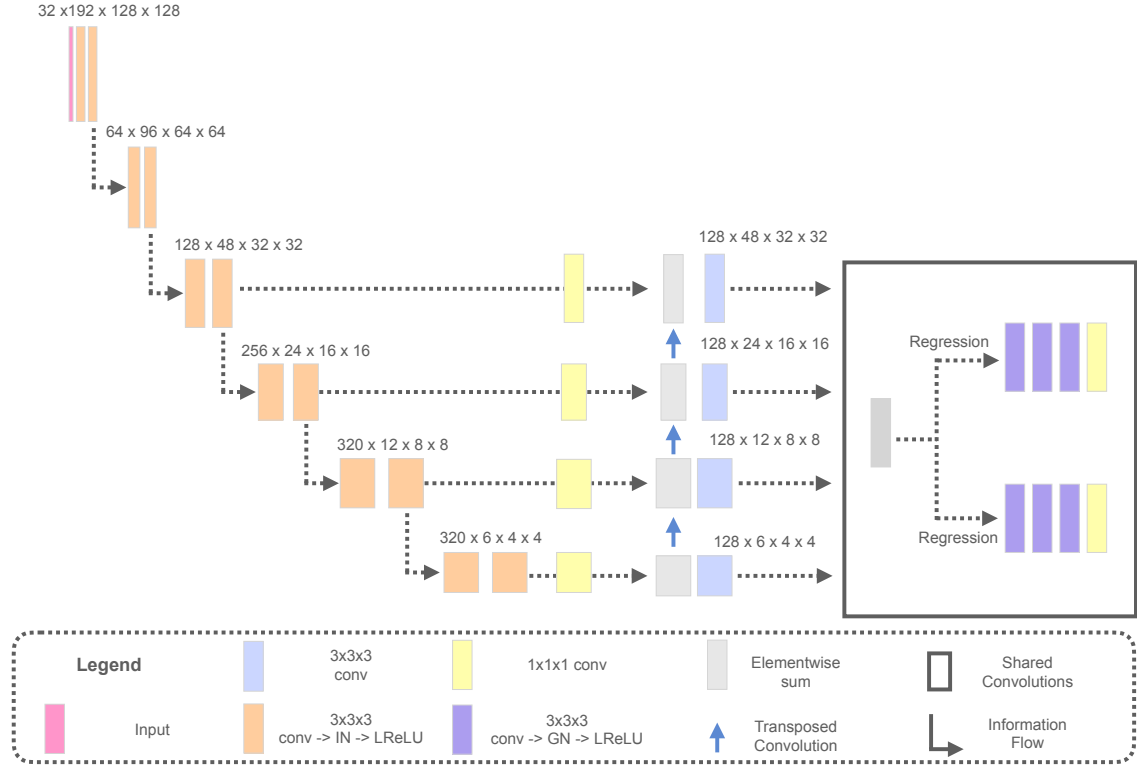


Figure 4.4: Retina Net architecture for vessel occlusion detection. Shows the Retina Net architecture which can be divided into three parts: first an encoder is used to extract a multi-scale representation of the image, second a decoder is used to recombine low-resolution features with high-resolution features and third a shared detection head is used to predict bounding box coordinates and confidence scores. This figure is adapted from [183].

The model is trained for 60 epochs, each consisting of 2500 batches. SGD with Nesterov momentum is used as the optimizer. Stochastic Weight Averaging (SWA) with a cyclical learning rate is employed during the final 10 epochs. The batch size is set to eight while ensuring that half of the samples in each batch contain at least one occlusion. Online data augmentation was applied to artificially increase the diversity of the training data. The augmentation pipeline closely follows nnU-Net [47] but excludes the simulation of low-resolution data.

Inference: Inference was conducted using a sliding window approach with 50% patch overlap. Duplicate predictions within and across patches were eliminated using NMS with an IoU threshold of 0.3. Predictions located near the patch centre are given higher weight than those near the edges. Low-confidence predictions below the confidence threshold of 0.2 are discarded. WBC with an IoU threshold of 0.4 is used to ensemble the

predictions. No additional weighting based on the volume of the predictions is applied. Bounding boxes smaller than seven voxels in any dimension are discarded.

Inference was performed on a DGX A100 system from which four NVIDIA A100 GPUs were used, each with 40 GB of VRAM. To enable flexible scaling to an arbitrary number of GPUs, patches extracted from a single patient are distributed across the different devices and aggregated prior to the ensembling step.

Threshold Selection: Detection methods predict a set of bounding boxes with associated confidence scores to rank the predictions against each other. Fully automated decision-making requires an additional confidence threshold to determine the final set of predictions and discard low-scoring ones. Given that the validation sets during cross-validation are evaluated by a single model, while the test sets are assessed using an ensemble, there is potential for a shift in the distribution of confidence scores.

To address this issue, an additional experiment was conducted to determine the confidence threshold using an ensemble. During the experiments, the UKHD training cohort is split into two subsets called the mini-training set (n=418) and mini-validation set (n=417). The mini-validation set was artificially balanced, like the internal UKHD test set, and included 210 patients with at least one vessel occlusion and 207 control patients. Five-fold cross-validation was used to train the ensemble on the mini-training set, followed by an evaluation of the mini-validation set. The confidence threshold is set to 0.647 to maximize the F2 score on the mini-validation set.

4.1.3 Exploring Detection Transformers for Medical Object Detection

Disclosure of this work

This section includes portions of our work that has been published in:

Ickler, M. K.*, Baumgartner, M.*, Roy, S., Wald, T., & Maier-Hein, K. H. (2023, June). "Taming Detection Transformers for Medical Object Detection." In *BVM Workshop (pp. 183-188)*. Wiesbaden: Springer Fachmedien Wiesbaden.

* contributed equally

Many methods formulate detection as a classification and regression task of predefined anchor boxes, requiring additional post-processing steps due to duplicate predictions of the same object. DETR models formulate detection differently as a direct set prediction problem. The transformer architecture is used to implicitly learn to suppress duplicate predictions and produces a set of objects, each represented by a confidence score and a bounding box. Despite their beneficial properties, they remain under-explored in the

context of medical detection tasks. This study analyses their feasibility by applying three different DETR models to four medical object detection tasks. The following section introduces the utilized data sets and methods.

Data Set Analysis

This study uses four data sets to derive insights of DETR models across different data set sizes, object structures and task difficulties:

The **CADA** data set was initially used as part of a MICCAI Challenge [17] and consists of 109 images with 127 objects. It represents the lower end of data set sizes in terms of images and can be effectively solved by anchor-based detection models. The images are 3DRA scans from a digital subtraction AXIOM Artis C-arm system. They were acquired at the Neurosurgery Department, Helios Klinikum Berlin-Buch. All labels are provided as instance segmentations. **RibFrac**[11, 12] represents the largest data set regarding the number of objects. It comprises 4422 rib fractures across 500 thin slice CT images from two scanners (GE Healthcare & Siemens Healthineers). Annotations are provided in instance segmentation format, and no further processing is necessary. It was originally used in the MICCAI 2020 challenge as an instance segmentation problem. Due to the inclusion of images with many fractures, it challenges modern detection models. **KiTS19** [181] contains 204 CT scans in the corticomedullary contrast phase who underwent nephrectomy. The data was originally part of the 2019 Kidney Tumor Segmentation Challenge and was annotated with semantic segmentation labels. We applied connected component analysis to the labels to cluster them and manually checked the smallest objects to remove clustering errors. Some examples are shown in Figure 4.5. KiTS19 represents a medium-sized data set where most tumours are quite large. The fourth data set is **LIDC** [13], which encapsulates thoracic CT scans acquired from various scanner manufacturers (GE, Philips, Siemens, and Toshiba) and reconstruction kernels. The pulmonary nodules were annotated by multiple clinicians and span primary lung cancer, metastasis or benign nodules. Each radiologist provided a score between one and five to indicate the likelihood of malignancy of the nodule and segmented nodules, which are larger than 3mm. The detection problem is formulated with two classes and includes all nodules with at least two annotations: nodules with an average malignancy score of three or larger were considered malignant, and all other nodules are considered benign. The segmentation of each nodule was derived via majority voting, where missing segmentations were considered to be zero. This data set represents the largest data set regarding the number of images and poses a difficult classification problem since benign and malignant nodules share a similar appearance.

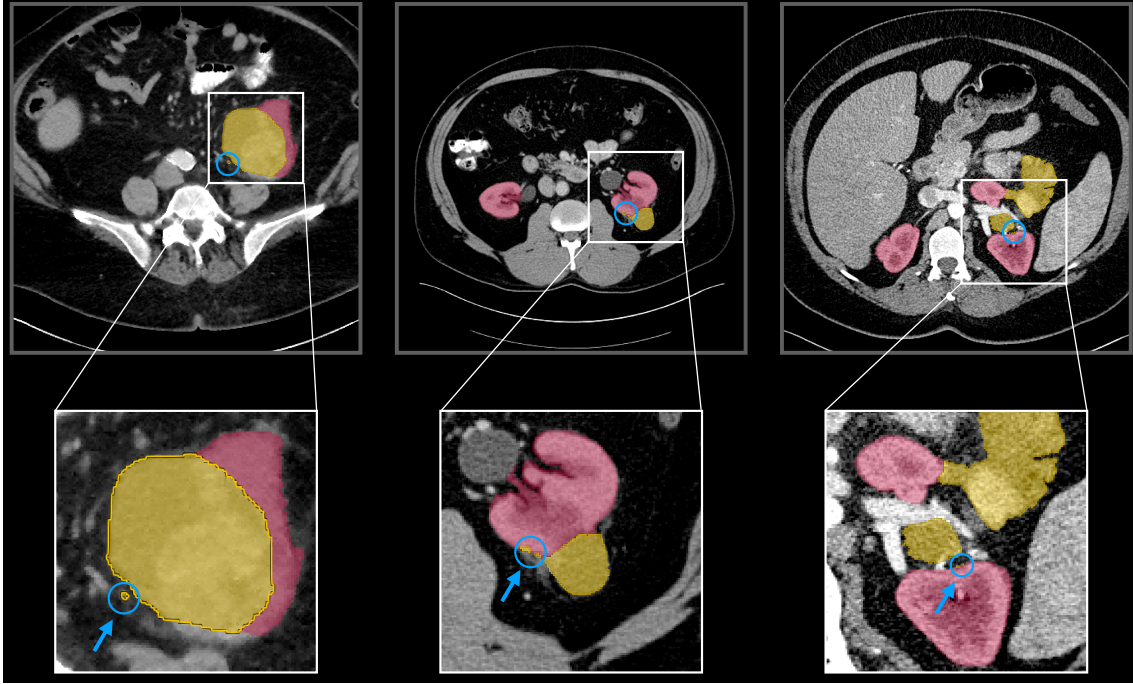


Figure 4.5: Removed object clusters from KiTS19 data set. This figure illustrates three examples where small clusters of pixels were connected to individual objects. The tumour is shown in yellow, the kidney in red and the cluster is delineated by a blue circle and arrow. The lower row provides zoomed crops. Since they do not represent real objects, they were manually removed from the annotations. This figure is adapted from [185].

Methodological Design

This study explores the feasibility of DETR models for medical object detection tasks. It comprises three different direct set prediction models, namely DETR [38], Conditional DETR [61] and DINO DETR [39]. An overview of the employed architectures is shown in Figure 4.6. The following section will shortly describe these methods.

DEtection TRansformer (DETR): Detection Transformer models [38] formulate detection as a direct set prediction task without requiring additional proxy formulations. The architecture comprises four components: a feature encoder, transformer encoder, transformer decoder and detection head. First, a features encoder extracts low-resolution features from the image. In our design, this is achieved by using plain blocks of convolutions in combination with instance normalisation [189] and LReLU non-linear activation functions. The final feature map is flattened to a sequence of tokens, which is used as input for the transformer encoder, to combine information across all spatial positions. Absolute positional encodings [194, 38] are added to the query and key of each self-attention layer to retain the positional information. The resulting output is fed to the transformer

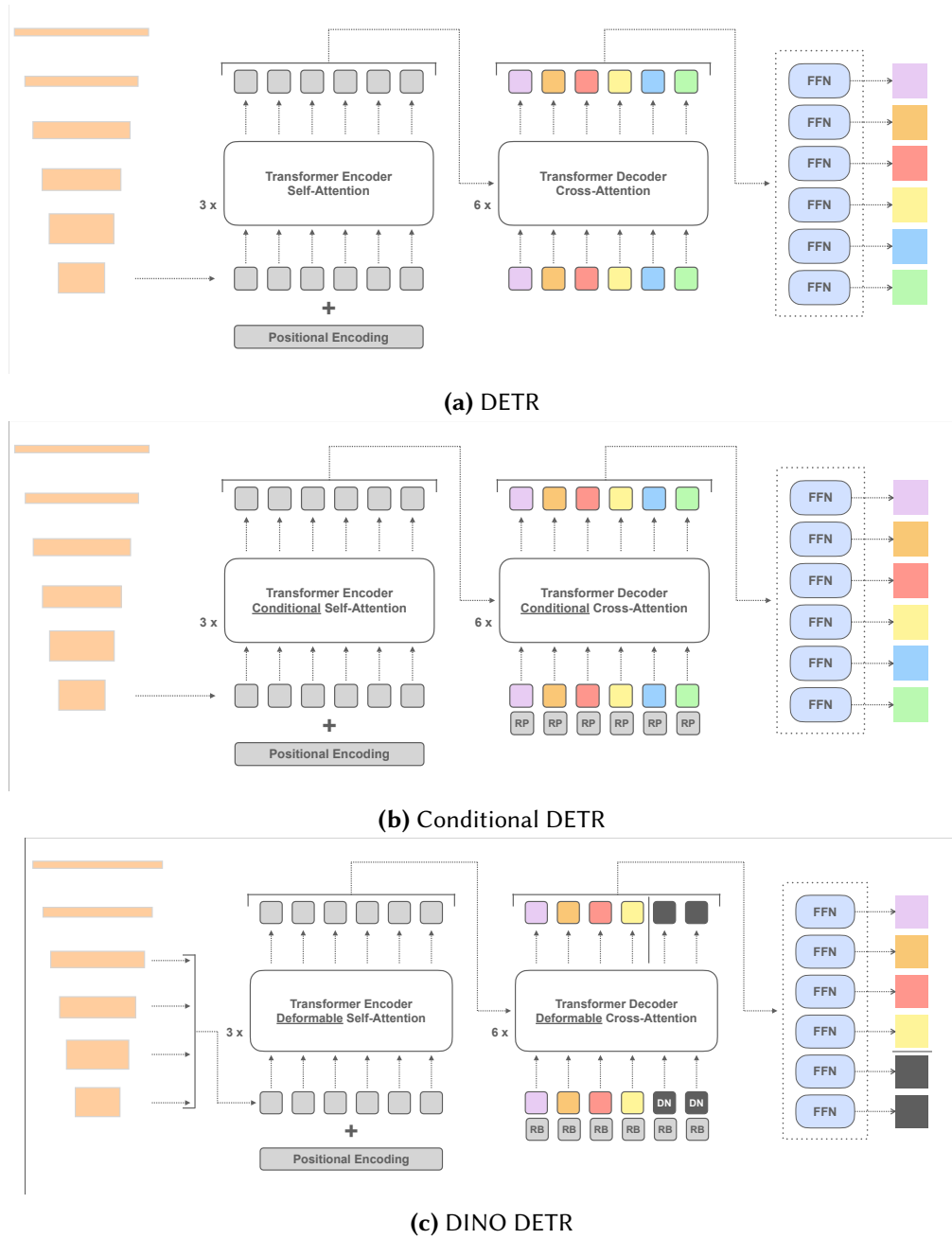


Figure 4.6: Architectural patterns of DETR, Conditional DETR and DINO DETR. Shows the architectural design of three DETR models [38, 61, 39], which can be divided into a feature encoder (orange blocks), transformer encoder, transformer decoder and detection heads. The object queries are visualised as coloured boxes. Conditional DETR assembles the object queries from a content part and a reference point (RP). DINO DETR replaces the reference point with a reference bounding box (RB). De-noising queries are shown as boxes with 'DN'. Figure adapted from [184].

decoder, which uses cross-attention to extract information about objects. Learnable object queries are used to prompt the decoder network for objects. Finally, a FFN is applied to predict bounding box coordinates and confidence scores, representing the detection head. Additional auxiliary predictions and losses are computed from each transformer decoder block while sharing FFN parameters. During inference, the second-highest confidence score for each predicted bounding box is used if the background class is assigned the highest confidence.

Conditional DETR: One of the major downsides of the DETR model compared to classical anchor-based detectors is its required training length to converge properly. Many solutions have been explored to speed up the training [61, 63, 60, 59, 39, 62, 195], one of them being Conditional DETR [61]. The architectural design of Conditional DETR [61] is similar to the original DETR [38] architecture but uses a different attention mechanism. In classical attention, each key and query is composed of content $\{k, q\}_c$ and spatial $\{k, q\}_s$ information, which are summed to $\{k, q\}$. The image features represent the content information, and the positional information is inserted via the positional encoding. When computing the dot product within the attention operation between $\{k, q\}$ the resulting terms include several cross-dependencies between content and spatial information, see Equation (4.2).

$$\begin{aligned} q^T k &= (q_c + q_s)^T (k_c + k_s) \\ &= q_c^T k_c + q_c^T k_s + q_s^T k_c + q_s^T k_s \end{aligned} \quad (4.2)$$

The products where spatial information is set in correlation with the content information can be difficult to learn, and as such, having a formulation which de-correlates these mechanisms is preferable. Instead of building the sum of the content and spatial information, it is also possible to concatenate them, yielding $q_c^T k_c + q_s^T k_s$ as the dot product. This formulation has shown impressive improvements in convergence speeds [61]. The spatial information of the queries is modelled by reference points (RP), which are predicted by a MLP from the object queries. To ensure the same representation space, the reference points are embedded via the same sine embedding function as the positional embedding of the keys and multiplied via a scaling factor, which is determined via a MLP from the decoder output. A scale of one is used for the first decoder layer.

Conditional DETR [61] follows the loss design of Deformable DETR [147] and replaces the CE loss with the focal loss [35] to train the classification branch. During inference, the top k highest scoring predictions are used as the detection result and k is set to the number of queries.

DINO DETR [39] builds upon several other proposed mechanisms to create the first DETR based model which is competitive against state-of-the-art anchor-based detectors on the COCO data set [85]. Instead of using reference points to encode spatial information

of the queries, dynamic anchor boxes [60] are used to additionally encode information about the spatial size of the proposal. Furthermore, denoising queries [59] are added during the training process to provide additional supervision signals during training. This concept is further refined by DINO DETR, which introduces additional negative samples during the denoising step and is named contrastive denoising training [39]. To leverage the power of multi-scale information during the prediction process, multi-scale deformable attention [147] combines information from multiple levels of the encoder network. Deformable DETR [147] introduced the concept of *iterative bounding box refinement* where initial proposals are iteratively refined by the transformer decoder network. DINO DETR extends this process to the *look forward twice scheme*, which applies the predicted bounding box deltas to two bounding boxes, allowing for improved gradient flow.

Training Parameters: To provide a fair comparison between the architectures, the same feature extraction network is used across all methods. Training is conducted for 2500 batches per epoch with a batch size of four. The number of epochs was manually adjusted for each model and data set, as shown in Table 4.2. PolyLR [47] with an exponent of 0.9 starting at 0.0001 is used to adjust the learning rate throughout the training. Gradient updates are performed via AdamW [196], and a weight decay of 0.0001 is used as regularization. The patch size is adopted from the anchor-based baseline model and kept the same across all models. The augmentation scheme follows a similar design as the preliminary version of nnDetection, described in Section 4.2.6. However, it is modified to speed up the training by decreasing the maximal rotation to 20 degrees and increasing the lower limit of the scaling augmentation to 0.8. Our volumetric adaptation of DINO DETR uses the lowest three resolution maps for processing multi-scale information, the hidden dimension is set to 120 or 128, and the FFN dimension is set to 1024. The number of queries is adapted for each data set and model, as shown in Table 4.2.

Baseline: The Retina U-Net [46] architecture is used as a strong anchor-based detection baseline. Its effectiveness was demonstrated throughout many experiments, including Section 5.1.1, and achieves robust performance across the selected tasks. The network configuration was automatically derived by the preliminary version of nnDetection, which is introduced in Figure 4.9. All methods are implemented within the same framework and use the same data preprocessing, data loading and inference pipeline.

Table 4.2: Hyperparameters of DETR models for selected data sets. Shows the number of queries and training epochs for three DETR models across four data sets. Abbreviations: DE=DETR, CD=Conditional DETR, DI=DINO DETR. Table reproduced from [184].

	CADA			RibFrac			KiTS19			LIDC		
architecture	DE	CD	DI	DE	CD	DI	DE	CD	DI	DE	CD	DI
# queries (denoising)	6	12	20 (8)	20	50	90 (20)	6	12	20 (8)	12	24	40 (8)
# epochs	50	50	25	125	100	75	100	100	35	200	100	75

4.2 Self-Configuring Design of Medical Object Detection Methods

Disclosure of this work

This section includes portions of our work that has been published in:

Baumgartner, M.^{*}, Jäger, P. F.^{*}, Isensee, F., & Maier-Hein, K. H. (2021). "nnDetection: a self-configuring method for medical object detection." In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part V* 24 (pp. 530-539). Springer International Publishing.

Baumgartner, M., Ickler, M. K., Jäger, P. F., Isensee, F., Ulrich, C., Wald, T., Holzschuh, J., Kovacs, B., Ghosh, P., for the ALFA study, & Maier-Hein, K. H. (2021). "nnDetection: A Self-configuring Method for Volumetric 3D Object Detection" *Currently in preparation*.

^{*} contributed equally

Self-configuring method design offers great versatility when performed on diverse data sets and with robust design decisions in mind. As nnU-Net [47] already demonstrated, the resulting models can outperform task-specific models in the semantic segmentation domain and offer easy applicability to new tasks. This results in strong and standardised baselines for both researchers and practitioners. However, such a model does not exist for the detection domain, which is especially important for diagnostic decision-making. This section introduces three data set pools which are used to develop and evaluate nnDetection, the first self-configuring medical object detection method. Afterwards, the design of nnDetection is explained in detail and learnings from the manual design of detection methods (Section 4.1) are highlighted. This method addresses RQ2 (Section 1.2.2) by establishing a detection model which can be applied to all volumetric detection problems without requiring extensive expert knowledge.

4.2.1 Data Set Analysis

The development of self-configuring methods requires at least two pools of data sets: one pool for the development process, called the development pool, and one pool to evaluate the performance of previously unseen data sets. The pool of evaluation data sets in our study is divided into two parts: the generalisation pool and the benchmarking pool. While there exist multiple publicly available data sets, each of them is used to develop task-specific models rather than creating methods to work robustly across many tasks.

In total, 22 data sets are used to develop and evaluate our method. An overview of all utilized data sets is illustrated in Figure 4.7.

Medical image computing is primarily focused on semantic segmentation, with 70% of challenges being targeted at voxel-level evaluation [42]. However, diagnostic tasks can not be effectively evaluated at the voxel-level because they depend on the presence of objects rather than individual voxels. Common segmentation metrics, like the Dice score, are not capable of capturing this characteristic [149]. As a consequence, many detection tasks are published as segmentation data sets, which complicates the development of general-purpose object detection methods. We identified 11 data sets which can also be used as valuable detection tasks and converted their labels into a detection-compatible format for our study. Detailed information on each data set and the label conversion steps are included in Section B.3.3.

Development Pool: For the development of nnDetection 10 different data sets are used to cover various modalities, anatomical regions and object structures: MSD-L (D01, liver, tumour) [16], MSD-P (D02, pancreas, tumour) [16], MSD-HV (D03, liver, tumour) [16], MSD-C (D04, colon, tumour) [16], CADA (D05, brain, aneurysm) [17], RibFrac (D06, chest, fracture) [11, 12], KiTS21 (D07, kidney, tumour and cyst) [15], PICA1 (D08, prostate, tumour) [197, 198], ADAM (D09, brain, aneurysm) [44] and LIDC (D10, lung, nodule) [13].

The smallest data set in terms of images includes 109 scans (D05), and the largest one has 1295 images (D08). D09 has the fewest annotated objects ($n=125$), and D06 has the highest number of objects ($n=4422$). All data sets are used for cross-validation experiments during the development phase. Cross-validation results can be found in Figure B.1. For D01 - D04, an additional test split was extracted at the beginning of the development process to make configuration decisions for the inference pipeline (Figure B.4).

Generalisation Pool: The first pool of data sets used for evaluating the performance of nnDetection consists of nine additional data sets. These were withheld from the development of nnDetection and simulate the real-world application of our method to previously unseen data sets. The data sets include: KiPA (D11, kidney, tumor) [199, 200, 201, 202], MRA-A (D12, brain, aneurysm) [18], CT-PC (D13, pancreas, cyst) [203], DUKE (D14, breast, primary tumor) [25, 26], BraTS-M (D15, brain, metastasis) [204, 205], CT-PaCS (D16, pancreas, cancer) [206], MELA (D17, mediastinum, lesion) [19], VALDO-M (D18, brain, microbleed) [20, 21, 22, 23, 24] and LNDb (D19, lung, nodule) [207].

This pool comprises upcoming detection benchmarks and data sets originally proposed for segmentation tasks. It covers a wide variety of tasks and introduces previously unseen anatomical regions, image modalities, object structures and annotation types. Since no other detection baselines exist for these tasks, benchmarking is performed against nnU-Net Plus and nnU-Net Basic (see Section 4.2.8), which extend the nnU-Net framework with additional post-processing steps to predict bounding boxes. Each of the data sets is split into a training and testing split.

Pool	Icon	ID	Name	Anatomical Region	Object Type	Original Annotation	CR	BB	IS	SS	Modality	CT	3DRA	TOF MRA	ADC	T1w	T1c	T1n	T2w	T2f	T2s	Number of Image	Number of Objects
Development		D01	MSD-L	Liver	Tumor					✓		✓										131	862
		D02	MSD-P	Pancreas	Tumor					✓		✓										281	283
		D03	MSD-HV	Liver	Tumor					✓		✓										303	401
		D04	MSD-C	Colon	Tumor					✓		✓										126	129
		D05	CADA	Brain	Aneurysm				✓				✓									109	127
		D06	RibFrac	Chest	Fracture				✓			✓										500	4422
		D07	KITS21	Kidney	Tumor, Cyst				✓			✓										300	826
		D08	PICAI	Prostate	Tumor				✓						✓				✓			1295	232
		D09	ADAM	Brain	Aneurysm				✓					✓								113	125
		D10	LIDC	Lung	Nodule				✓			✓										1018	1884
Generalisation		D11	KIPA	Kidney	Tumor					✓		✓										70	72
		D12	MRA-A	Brain	Aneurysm		✓		✓					✓								296	198
		D13	CT-PC	Pancreas	Cyst					✓		✓										221	533
		D14	DUKE	Breast	Primary Tumor			✓							✓	✓	✓					911	911
		D15	BrATS-M	Brain	Metastasis					✓					✓	✓	✓				✓	561	4588
		D16	CT-PaCS	Pancreas	Cancer					✓		✓										1843	385
		D17	MELA	Mediastinum	Lesion			✓				✓										880	884
		D18	VALDO-M	Brain	Microbleed					✓					✓	✓			✓			72	236
		D19	LNDb	Lung	Nodule				✓			✓										236	768
		D20	LUNA16	Lung	Nodule		✓					✓										888	1186
Benchmarking		D21	PN9	Lung	Nodule			✓				✓										8798	40436
		D22	CTA-A	Brain	Aneurysm					✓		✓										1476	1590

Figure 4.7: Development, Generalisation and Benchmarking Pool of nnDetection. nnDetection is developed on the development pool consisting of 10 data sets which cover a wide range of image properties and objects. To evaluate the performance on unseen data sets, which include new modalities, object structures, anatomical regions and various annotation styles, the generalisation pool consisting of 9 further data sets is used. The benchmarking pool contains three data sets to compare against state-of-the-art task-specific models. The dataset with the highest number of images and objects in each pool is highlighted in bold. The dataset with the lowest number of images and objects in each pool is underlines. This figure is adapted from [82].

D11 has the fewest images ($N = 70$) and objects ($n = 72$) in the generalisation pool. D16 includes the highest number of images ($N = 1843$), and D15 the highest number of objects ($n = 4588$). The MRI data sets D14, D15, and D18 introduce modalities which were absent from the development pool. The DUKE data set is used for primary tumour detection in the breast area, an anatomical region which is not part of the development pool. Furthermore, D15 and D18 introduce metastases and microbleeds as novel object structures in the generalisation pool. The annotation style of the detection problem varies between the data sets, including spherical annotations in D12 and bounding box annotations in D17 and D14. In summary, the generalisation pool represents a highly diverse collection of medical detection problems to challenge current methods in many different scenarios. Cross-validation and test set performance visualised via box plots can be found in Figures B.2 and B.3.

Benchmarking Pool: The benchmarking pool summarises commonly used data sets to develop and compare task-specific models. Three data sets are used to compare nnDetection against current state-of-the-art task-specific methods:

- **LUNA16** [43] is a lung nodule detection data set consisting of 888 images. It is a filtered subset of the LIDC [13] data set but reduces the multi-class problem to a single-class problem. Furthermore, only spherical labels are provided, resulting in different data fingerprints than LIDC [13]. An official split with 10 subsets is publicly available and is used in a cross-validation fashion. Since the exact splitting procedure is not clearly defined, different splitting approaches are possible, leading to inconsistencies when reporting results. To provide the best possible comparison, we ran our experiments in two variations: the '8-1-1' split uses eight training folds, one validation and one testing fold and the '9-0-1' uses nine training folds and one testing fold. The empirical parameters of nnDetection are directly optimised on the test split in the latter version.
- **PN9** [14] is the largest publicly available detection data set, exceeding previous efforts in terms of the number of images and objects by a whole magnitude. Various lung nodule classes are annotated via bounding boxes, resulting in over 40.000 object labels. The data set is split into three subsets: training, validation and testing. All images are already resampled to a unified spacing of $1\text{mm} \times 1\text{mm} \times 1\text{mm}$ as well as clipped and normalised to an intensity range of $[0, 255]$.
- **CTA-A** [170] is a large annotated data set for the detection of aneurysms in CTA images. It was used by Ceballos-Arroyo et al. to compare multiple detection methods. Images are split into three groups: training, internal testing and external testing. The internal testing data set offers the opportunity to evaluate the developed method on unseen images with minimal distribution shift, while the external testing data set was collected at different institutions.

4.2.2 Development Process

The entire design of nnDetection was performed on the development pool consisting of 10 diverse medical detection tasks. Initial design decisions were inspired by nnU-Net [47] and consecutively questioned for validity in the detection domain. Further experiences were carried over from the manual design of detection methods introduced in Section 4.1. These were driven by simple yet robust methods rather than focusing on complex architectural novelties. This results in an automatic configuration process specifically tailored towards detection models.

nnDetection is based on the same parameter groups as nnU-Net [47]: rule-based, fixed and empirical parameters. Initially, a data fingerprint is automatically extracted from the data to build the basis for the following configuration process. It formalizes the most important aspects of the data set into accessible properties, which can be used by the rule-based design to dynamically determine data set-specific hyperparameters. The development experiments have shown that some parameters are data set-independent and can be kept constant; these are called fixed parameters. The last group, namely empirical parameters, require adaptation but can not be represented by a rule. They are determined by performing cross-validation experiments on the training data and empirically determining the best configuration based on the obtained results.

In contrast to the semantic segmentation domain, the detection domain encapsulates different annotation types and models, offering different strengths and weaknesses. These inhomogeneities are incorporated into the design by supporting five detection models within a unified framework. nnDetection offers models from three categories: 1STAGE models (Section 2.1.2) offer a simple yet robust design to predict objects from predefined anchors, 2STAGE models (Section 2.1.1) extend the one-stage design with an additional RoI head and a pooling operation to refine the initial proposals. The SETPREDICT model is based on the direct set prediction approach introduced in the DETR [38] section (Section 2.1.3). To accommodate different annotation types, the one- and two-stage models can either be trained with box-level supervision (BOX) or a combination of voxel-level and box-level supervision (MIX). This results in five potential models: 1STAGE-BOX (Retina Net [35], Section 4.2.6), 1STAGE-MIX (Retina U-Net [46], Section 4.2.6), 2STAGE-BOX (Faster R-CNN [36], Section 4.2.6), 2STAGE-MIX (Mask R-CNN [37], Section 4.2.6) and SETPREDICT (Deformable DETR [147], Section 4.2.6). To limit the number of training runs and save computational resources, a model proposal stage is introduced to select one representative from each model category, as further elaborated in Section 4.2.5.

4.2.3 Generalisation Process

nnDetection is a self-configuring medical object detection method which can be applied to new data sets without manual intervention. The entire design process, including data

fingerprint extraction and hyperparameter configuration, is automated and can be used by practitioners as well as domain experts to train powerful detection models. This design makes detection models available to the entire medical image computing community and provides strong baseline performance for future research.

To evaluate the generalisation capabilities of nnDetection and simulate its real-world adoption, experiments were conducted on two further data set pools, including 12 previously unseen detection data sets from the generalisation and benchmarking pool. As described in Section 4.2.6 minimal interventions to the training schedule of the SET-PREDICT model were necessary on two data sets to ensure proper convergence. These changes are now automated for future deployments of nnDetection. The remaining experiments are conducted without manual intervention.

4.2.4 Data Fingerprint

Medical images have a variety of different properties, like different image modalities and voxel spacings, which are not commonly found in other imaging domains. These properties have an impact on the configuration process of deep learning methods and need to be carefully incorporated into the design for optimal performance. The data fingerprint, originally introduced in nnU-Net [47], is a collection of such properties and extracts a compact representation of the data. However, the detection domain encapsulated properties different from those of the semantic segmentation domain, which requires changes to the composition of the fingerprint.

Medical images are acquired with different image modalities to highlight the clinically needed anatomical structures. They can represent different quantities like HU, resulting in different processing steps to make them suitable for neural networks. All images are represented by their intensity distribution, which captures the statistical properties of their voxel intensities. Since not all images capture the same body parts, the resulting scans have different image sizes, which need to be handled. These sizes can be collected and assembled to the image shape distribution of the data fingerprint. Images operate on discretized grids of the real world, where the spacing defines the distance between centres of voxels that the grid is made of. Furthermore, the collection of images and labels by themselves can be used to deduce empirical parameters, like the IoU threshold of the NMS operation, and reflect the training data by itself. Not all object structures require the same amount of spatial detail in their annotation, which results in different annotation types across the medical detection domain. Common styles include instance segmentations [11, 207, 13], bounding boxes [14, 25] or spherical representations [18, 43]. All detection tasks require annotated objects, which are always associated with two properties: object position and object size.

4.2.5 Rule-based Parameters

Rule-based parameters use the data fingerprint to adapt hyperparameters for a given data set. They are dynamically deduced, resulting in a custom configuration for each data set. These rules are explained in the following paragraphs (R1-R8):

[R1] Preprocessing - Target Spacing and Resampling

Medical images capture the physical world in a discretized grid where each grid cell can capture a varying extent of the physical world. This size is represented by the spacing of the images, which can be divided into in-plane and out-of-plane spacing. For most acquisition protocols, the out-of-plane spacing is higher than the in-plane spacing since volumetric data is captured in a slice-wise manner. Capturing more slices increases the acquisition time and, in the case of CT, also the radiation dose. Usually, the spacing of images varies within a data set, consequently resulting in varying fields of view of the information in the physical space. To provide the neural network with a constant physical field of view, all images are resampled to a unified target spacing.

Choosing an appropriate target spacing is essential to standardise the data set without removing critical information. When the target spacing is chosen too coarsely, small objects can vanish during the resampling step, making them undetectable for the method. Using a small target spacing results in images with many voxels, limiting the amount of contextual information for a given patch size and resulting in longer inference times due to the increased computational burden. nnDetection adopts the same rule as nnU-Net [47] to determine the target spacing of the full-resolution model by using the median spacing for each axis. If the largest spacing along any axes exceeds the smallest spacing by a factor of three, an adapted rule is triggered, which sets the spacing to the 10th percentile of the coarsest axis [47].

[R2] Preprocessing - Low-Resolution Models

Current 3D neural networks utilise a patch-wise training scheme due to restrictions in the VRAM of modern GPUs. On the other hand, some detection tasks contain target structures that extend significantly beyond currently possible patch sizes. This drastically limits the available contextual information in a single patch and complicates the inference process. During inference, the image is scanned via a sliding window scheme with overlapping patches, resulting in multiple predictions for each object. If an object extends across multiple patches, it is not sufficient to suppress duplicate predictions, but it becomes a necessity to stitch them together. This constitutes a difficult task, and *stitching artefacts* can occur during the prediction, resulting in duplicate predictions of the same object.

As pointed out in our submission to the MELA challenge, see Section 4.1.1, this can be partially circumvented by increasing the target spacing resulting in lower-resolution images. This enables the model to learn coarser information and increases the field of view of the physical space within a single patch. If the 99.5th percentile of the object sizes exceeds the current patch size, additional low-resolution versions of the data set are triggered until the condition is not fulfilled anymore. These versions are also referred to as resolution stages in this thesis. Each stage doubles the spacing along all axes, effectively halving the number of voxels along each axis.

[R3] Preprocessing - Intensity Normalisation

Normalising the inputs for neural networks is important to avoid numerical instabilities during training, since unnormalised inputs can yield vanishing or exploding gradients. A good normalisation scheme can also be used to attenuate selected properties of images, for example, by applying thresholding operations to remove unnecessary information.

Voxel intensities of CT images represent a physical quantity, describing the absorption of radiation in tissue, and are measured in HUs. As a result, the intensity values produced by different CT scanners are comparable to each other. Clinicians utilise this fact to define intensity windows which highlight clinically relevant regions in images. These intensity windows vary between tasks and can thus not be used directly to normalise the inputs in nnDetection. Instead, the 99.5-th percentile and the 0.5-th percentile are used as the upper and lower bound for clipping, respectively. The mean and standard deviation of the voxel intensities are collected across the entire data set and used to normalise each image with z-score normalisation. This normalisation scheme first subtracts the mean and then divides the intensities by the standard deviation. All other modalities use a z-score normalisation where mean and standard deviation are computed on a per-image basis. This rule was adopted from nnU-Net [47].

[R4] Training - Patch Size and Network Topology

Choosing the correct patch size for a given problem is essential to achieve the best possible results. It has complex interactions with many other hyperparameters and needs to balance the available contextual information for the model and compute resources. As a starting point, the patch size is initialized to the median shape of the data set. This represents the largest useful patch size for a given task as padding would need to be applied to the majority of images to support even larger inputs. Most volumetric data sets result in initial patch sizes which vastly exceed the available VRAM. Starting from there, the patch size is iteratively reduced while keeping the physical dimensions constant across all axes. This is repeated until the configured memory budget is met. The resulting patch size is anisotropic for anisotropic spacings, and isotropic patch sizes are used for

isotropic spacings. VRAM estimation is performed via a heuristic to keep the procedure deterministic and reproducible across all devices. This principle for determining the patch size was adopted from nnU-Net [47].

The network topology needs to be adapted depending on the configured patch size via two configuration parameters: (1) network depth and (2) kernel and pooling sizes. Deeper networks can capture a larger field of view in deep layers, which complements larger patch sizes and enables the detection of large objects. However, decreasing the size of the feature maps indefinitely can influence other operations, such as normalisation layers, which can only provide a robust statistical estimate for larger feature maps. To avoid this problem, the network is configured to retain a minimum feature map size of $4 \times 4 \times 4$. All models within the nnDetection framework use the same patch size to be comparable with each other. The default memory budget is allocated to roughly 12-16GB of VRAM. Second, the kernel and pooling sizes are configured in accordance with the patch size. For isotropic patches, the pooling and kernel sizes operate along all axes equally. But fusing information across very anisotropic axes can result in performance degradation. In this case, early kernel sizes are configured to be anisotropic, effectively operating in a two-dimensional mode, and information across slices is only fused later in the network.

Anchor-based detectors, like Retina U-Net [46], use predictions from multiple scales to detect objects of different sizes. High-resolution feature maps are better suited to detect small objects, while coarse feature maps capture large field of views to detect larger objects. The FPN is responsible for combining information from multiple scales and adjusting the number of channels for the shared detection head. For deep network configurations, the deepest four levels are used for the detection heads. When small patch sizes are used in combination with shallow networks, the FPN levels will automatically shift upwards and use higher resolution features.

[R5] Training - Number of Objects per Patch

All detection models within nnDetection require an upper limit of detections per patch. Choosing a good limit is essential to achieve robust performance: if the limit is set too high, many unnecessary predictions will be produced that need to be processed. In direct set prediction models, like DETR, this also influences the training convergence since the loss would include much more background samples. On the other hand, using a restrictive threshold will suppress correctly predicted objects.

Since the training and the inference procedure are conducted in a patch-wise manner, it is not sufficient to compute statistics across the entire image. It is necessary to estimate the number of objects that are usually contained in a single patch, a quantity that depends on the selected patch size and data set characteristics. In order to obtain a good estimate, the distance between the centre points of the objects is calculated and compared against the patch size. This measure is computed for every image in the data set, and the value

O_{img} is set to the 95th percentile across all images. The upper limit for anchor-based detectors is set to $4 * O_{\text{img}}$ with a minimum of 100 predictions. The number of predictions for DETR models is set to $3 * O_{\text{img}}$ with a minimum of 12 predictions. These values were determined during the development process and ensure that the detectors are able to predict a sufficient number of objects.

[R6] Training - Anchor Sizes

The core component of anchor-based detection models are the predefined anchors resembling potential bounding box positions. The selection of anchors influences both the training and inference characteristics of these models. As such, an inappropriate anchor composition will result in difficult optimisation problems during training since the models need to learn to compensate for the large disparities. This will impact decisions concerning the assignment heuristic and scaling of losses. However, choosing the correct anchors is a powerful tool for injecting prior knowledge about object sizes into the model and can be especially beneficial for smaller data sets.

Due to the large variation of object sizes across the axis, nnDetection uses 27 anchor boxes at each position. Three different sizes are defined along each axis, and the Euclidean product is used to combine all of the anchor sizes. The pooling strides are used to scale the anchor sizes from the initial level to the deeper levels, reflecting the increase in the field of view. Their initial size is determined via iterative optimisation with differential evolution, utilising the TwoPointsDE algorithm from the *nevergrad* library [208]. The IoU between the anchors and the ground truth object sizes is used as the optimisation criterion.

[R7] Training - Pseudo 2D Augmentation

Augmentation is an essential tool to increase the diversity of the training data set and avoid model overfitting. Section 4.1.1 outlined the importance of proper configuration of the augmentation pipeline and evaluation scheme to obtain optimal performance. All details regarding the augmentation pipeline are described in Section 4.2.6 as part of the fixed parameters. Nonetheless, some spatial augmentations operate along all dimensions equally, which implicitly assumes isotropic data. In the presence of anisotropic data, it can be harmful to augment the data extensively along the out-of-plane axis due to the occurrence of interpolation artefacts. To avoid this, rotation and scaling are only performed along the in-plane axis in this scenario.

[R8] Training - Model Proposals

The medical detection domain has a large variety of annotation types since they offer different benefits: (1) Instance segmentations offer the richest supervision signal where objects are delineated on the voxel level. On the other hand, they also require the highest annotation effort. (2) Bounding boxes offer a quicker annotation solution and encapsulate the position and size of the object. (3) Spherical annotations can be annotated the quickest but only capture the position and largest extent of the object. The annotation type does not solely depend on the available annotation budget, as not all medical detection tasks benefit from additional supervision. We observed this behaviour in an initial experiment in the context of the vessel occlusion study Section 4.1.2. If it is not possible to clearly delineate objects by their boundaries, the availability of segmentations does not boost the model's performance. Based on the available annotation type, various models have been proposed which can utilize the richer signals if available. To allow for a unified pre-processing and augmentation pipeline, box-level annotations are converted to (pseudo) voxel-wise labels during preparation even when the trained models do not depend on voxel-level supervision.

nnDetection encapsulates two types of models to cope with the diversity of annotation types: (1) BOX models are only trained with bounding box supervision and do not incorporate additional information in the form of segmentations, and (2) MIX models are trained with a mixture of box-level supervision and voxel-level supervision to leverage the richer training signal when available. This implicitly assumes that the appropriate annotation type was already determined by the clinician performing the annotations, and objects without clear boundaries are annotated with bounding boxes. Anchor-based models like Retina Net [35] and Faster R-CNN [36] have direct counterparts which can use additional voxel-level supervision, namely Retina U-Net [46] and Mask R-CNN [37]. Direct set prediction models are always trained with box-level supervision since they do not incorporate segmentations in their design. To limit the number of necessary training runs, the model proposal stage selects three models depending on the available annotation type: (1) if box-level annotations are available in the form of either bounding boxes or spheres, the 1STAGE-BOX, 2STAGE-BOX and SETPREDICT models are recommended for training and (2) if voxel-level annotations are available, the 1STAGE-MIX, 2STAGE-MIX and SETPREDICT models are proposed. This ensures that the appropriate model is chosen for the present task, utilising the available annotation to its full extent. The SETPREDICT model is trained in both instances since it provides great results when trained on larger data sets even when no voxel-level supervision is utilised.

4.2.6 Fixed Parameters

This parameter group encapsulates parameters that generalise robustly across detection tasks and are thus kept constant.

Preprocessing

The images are resampled via third-order B-spline interpolation as in [47]. Segmentation maps can only contain discrete values where each object is encoded as a separate label to retain the object-level information. In the case of higher-order resampling strategies, the labels need to be encoded as one-hot vectors and discretised after interpolation. However, this results in vastly varying memory requirements and compute times between segmentations with few and many objects. All segmentations are resampled via nearest neighbour interpolation in nnDetection to alleviate this concern.

Training - One-Stage Anchor-based Detection Blueprint

Two anchor-based one-stage detection models are available within nnDetection: Retina Net [35](1STAGE-BOX) and Retina U-Net [46] (1STAGE-MIX). If voxel-level annotations are available, the 1STAGE-MIX model is used to leverage this additional information. Otherwise, only the 1STAGE-BOX model is used to avoid conflicting information from pseudo-segmentation masks.

Architectural Design: Each network is divided into three components: encoder, decoder and detection head. The 1STAGE-MIX architecture uses an additional semantic segmentation head at the highest resolution to leverage voxel-level supervision. An architectural overview is provided in Figure 4.8.

The encoder network is responsible for extracting multi-scale features from the original image and passing them to the decoder network. It follows a simple CNN design where each level consists of two stacked convolutions followed by Instance Normalisation [189] and a Leaky ReLU non-linear activation function. The first convolution of each level is responsible for performing the downsampling operation, which is realised via strided convolutions. A FPN is used as the decoder architecture and is responsible for recombining coarse information with fine-grained information. Upsampling is performed via transposed convolutions, and features from different levels are combined using element-wise addition. If the 1STAGE-MIX architecture is used, the decoder is extended to the full image resolution, and higher levels consecutively reduce the number of feature maps that are not used by the detection head. The depth, kernel sizes, pooling sizes and detection head levels are determined via the configuration process outlined in Section 4.2.5. The detection head is shared across multiple levels, predicting objects across various scales and consists of a classification and a regression branch. Both branches share the same design with two convolutions followed by Group Normalisation [190] and Leaky ReLU non-linear activation functions. The outputs are generated via pointwise convolutions to produce confidence scores and regression deltas. The optional segmentation head uses a pointwise convolution to produce the segmentation outputs as well.

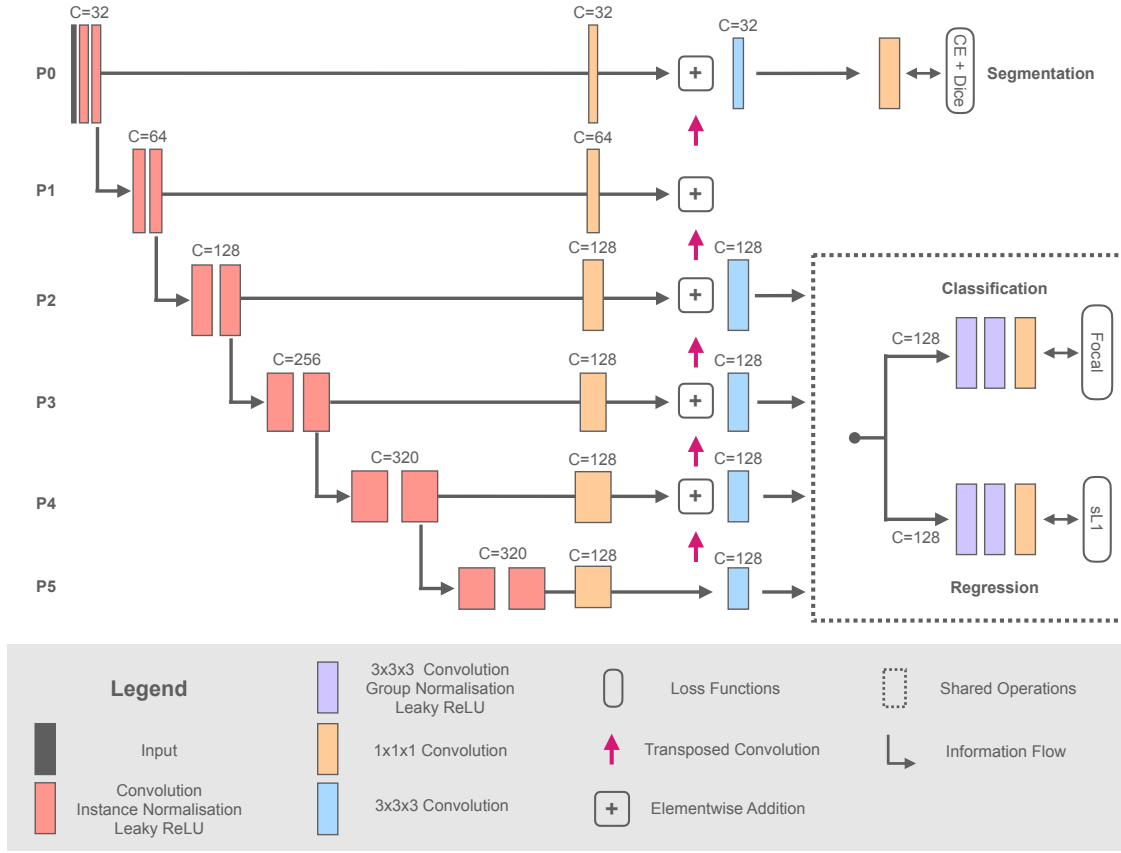


Figure 4.8: One-Stage Anchor-based Architectural Blueprint. Shows the Retina U-Net architecture (1STAGE-MIX) using object-level and voxel-level supervision. The encoder consists of stacked convolutions with Instance Normalisation and Leaky ReLU non-linear activation functions. The decoder is realised via a FPN with transposed convolutions. The detection head is shared across multiple levels and consists of a classification and regression branch. Semantic Segmentation is performed at the highest resolution level of the decoder and removed for the 1STAGE-BOX model. This figure is adapted from [82].

The anchor classification is trained via the focal loss (Section 2.1.2) with $\gamma = 1$ and $\alpha = 0.75$ (Equation (2.8)). The normalisation of the loss is performed by taking the mean across classes Y and sum across all anchors A to stabilise gradients, see Equation (4.3).

$$L_{cls} = \sum_A \frac{1}{|Y|} \sum_{y_i \in Y} L_{focal}(y_i, \hat{y}_i) \quad (4.3)$$

The smooth L1 (Section 2.1.1) is used to regress the regression targets for the positive anchors. The classification and regression losses are normalised by the moving average of the number of positive anchors. The bounding box deltas are encoded the same way

as in the original R-CNN publication [95] (see Section 2.1.1). The segmentation branch is trained via a combination of the Dice (batch dice version without smoothing factor) and CE loss. The full loss function can be written as Equation (4.4).

$$L_{1stage} = 0.3 * L_{cls} + L_{reg} + L_{seg,Dice} + L_{seg,CE} \quad (4.4)$$

Trainig Schedule: Both models are trained for 50 epochs with a batch size of 4 and 2500 training batches per epoch. SGD with Nesterov momentum of 0.99 is used to update the parameters of the model. The learning rate starts at $1e-6$ and is linearly increased to 0.01 over 4000 iterations. A PolyLR [47] schedule is used onwards. If numerical instabilities are encountered, training is automatically restarted with a momentum of 0.9. A complete overview of model-specific parameters is visualised in Figure 4.9.

Training - Two-Stage Anchor-based Detection Blueprint

Two-stage models use a RPN to generate an initial set of candidates and refine them via the RoI head. nnDetection encapsulates a 2STAGE-BOX model (Faster R-CNN [36]) and a 2STAGE-MIX model (Mask R-CNN [37]) which are dynamically selected based on the annotation type.

Architectural Design: The RPN model uses the same architectural blueprint as the 1STAGE models but replaces the focal loss (Section 2.1.2) with a combination of stochastic online HNM and the BCE loss. Available anchors are subsampled to balance the number of positive (one-third) and negative (two-thirds) matches. Since balanced sampling is not possible for patches without objects, the anchors are selected across the entire batch rather than individual images. A 3D version of the RoI Align (Section 2.1.1) operation is used to pool features from the selected feature level for the RoI head.

The RoI head consists of a classification, regression and segmentation branch. Feature maps are pooled to a size of $5 \times 5 \times 5$ for the regression and classification branch and processed by four convolutions. Each convolution is followed by a Group Normalisation [190] layer and Leaky ReLU non-linear activation function. Afterwards, an average pooling layer and a pointwise convolution are used to classify and regress the original proposals. The segmentation branch is only present for the 2STAGE-MIX architecture and produces binary segmentation masks for each object. It processes features of size $9 \times 9 \times 9$ and uses a transposed convolution to increase the output resolution of the mask to $18 \times 18 \times 18$. An architectural overview of the RoI head is shown in Figure 4.10.

The BCE loss is used to train the classification branch, and the smooth L1 loss [101] is used to refine the proposals. Stochastic online HNM balances the number of positive and negative proposals for the loss computation. Candidates with an IoU below 0.2 are considered negative examples, and candidates above 0.3 are considered positive examples.

Parameter	Dependencies	Description
Network Topology & FPN Levels & Patch Size	Median Image Shape, Target Spacing, GPU Memory Budget	The patch size is initialised with the median image shape after resampling and iteratively decreased until the memory requirements are met. The physical field of view of the patch is kept constant. The network topology defines the depth, kernel sizes and pooling sizes based on the current patch size. For deep configurations the last four levels are used for predicting objects. For shallow configurations the levels are shifted upwards.
Anchor Optimisation	Object Sizes, Target Spacing, Network Topology	Three anchor sizes per axes are used and combined with the Euclidean product to derive a set of 27 anchors per position. The sizes are optimised to achieve the highest IoU with reference object sizes.
Number of Predictions	Object Positions, Patch Size	Determines the number of objects located within the current patch size, denoted by O_{img} . Number of predictions is set to the 95th percentile multiplied by 4 and a minimum of 100.
Model Proposal	Annotation Type	The Retina U-Net (1STAGE-MIX) model is used in presence of segmentation annotations. For other annotation types, the Retina Net (1STAGE-BOX) architecture is used.
Optimiser & Learning Rate		The learning rate is linearly increased from 1e-6 to 1e-2 for the first 4000 iterations. A PolyLR schedule is used for the remaining training. SGD with Nesterov momentum of 0.99 is used to update the parameters. If training does not converge, an automatic restart with momentum 0.9 is triggered.
Architecture Template		Encoder: stacked convolutions with Instance Normalisation and Leaky ReLU non-linear activation functions. Decoder: FPN style decoder network with transposed convolutions to upsample features. Number of channels decreases for upper levels. Detection Head: classification and regression branch each consisting out of two convolutions with Group Normalisation and Leaky ReLU activations. The outputs are generated via pointwise convolutions. Segmentation Head: convolution to produce semantic segmentation. Only used for 1STAGE-MIX model.
Anchor Matching		The assignment between anchors and reference objects is performed via Adaptive Training Sample Selection (ATSS), using a dynamic IoU threshold for each object.
Loss Functions		Classification Branch: Focal loss with alpha set to 0.75 and gamma to 1. The classification loss is weighted with 0.3 inside the multi-task loss. Regression Branch: The smooth L1 loss is used to regress deltas which are encoded in the R-CNN style. Segmentation Branch is trained with the Cross-Entropy loss and Dice loss. It only distinguishes between foreground and background. Only used if 1STAGE-MIX model is used.
Empirical Parameter Optimization	Training Data	Inference parameters are derived by empirical optimisation on the training data. Specifically, the IoU threshold for NMS per model, IoU threshold for WBC, a lower limit for the confidence score, a lower limit for object sizes are derived.

Figure 4.9: One-Stage Anchor-based Model Configuration. Visualizes the configuration of the 1STAGE-MIX and 1STAGE-BOX models. Only model-specific configuration options are shown here; model-agnostic parameters are omitted to provide a better overview. Additional dependencies for the parametrization are shown in the second column, and additional details are provided in the third column. Fixed parameters are shown in light blue, rule based parameters are shown in orange and empirical parameters are shown in purple. This figure is adapted from [82].

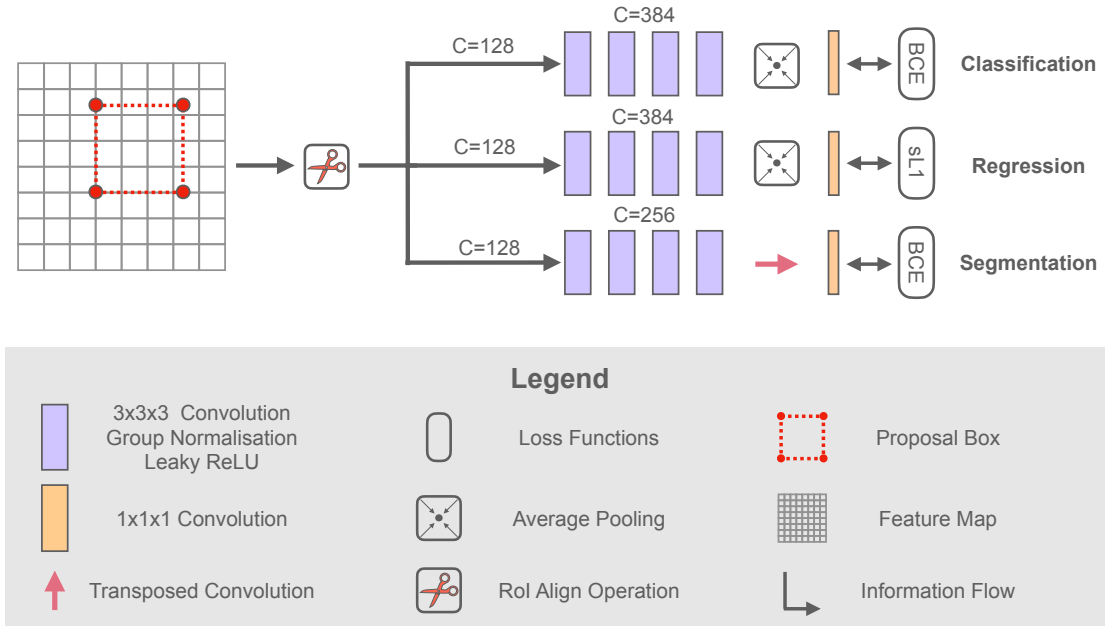


Figure 4.10: Two-Stage RoI Head Architecture. Shows the RoI head of the 2STAGE-MIX model using object-level and voxel-level supervision. The RPN is omitted for simplicity. The head comprises three branches responsible for classification, regression, and segmentation. The classification and regression branches follow the same design with four convolutions, an average pooling and a final point-wise convolution. The segmentation branch replaces the pooling operation with a transposed convolution to produce object masks. This figure is adapted from [82].

Region proposals between these two thresholds are ignored during loss computation. In contrast to the RPN, only one out of eight of the proposals contributing to the loss computation are positive, and seven out of eight are negative. To stabilise training in the beginning, ground truth bounding boxes are added as additional candidates to ensure that at least one positive proposal is present for foreground patches. The regression and the classification loss are normalised by the running mean over the number of selected positive proposals.

Trainig Schedule: Similar to the 1STAGE models, a training schedule with 50 epochs and 2500 batches per epoch is used. Each batch consists of 4 patches, and SGD with Nesterov momentum is used to update the parameters. The momentum term is set to 0.9, and the PolyLR learning rate schedule is used. An overview of all model-specific parameters is provided in Figure 4.11.

Parameter	Dependencies	Description
One-Stage Detector as RPN		Uses a 1STAGE detection model as the Region Proposal Network. This includes all corresponding rule-based, fixed and empirical parameters of the model.
Model Proposal	Annotation Type	The Retina U-Net 2 Stage Mask (2STAGE-MIX) model is used if voxel-level annotations are available. For other annotation types, the Retina Net 2 Stage (2STAGE-BOX) architecture is used.
Optimiser & Learning Rate		The learning rate is linearly increased from 1e-6 to 1e-2 for the first 4000 iterations. A PolyLR schedule is used for the remaining training. SGD with Nesterov momentum of 0.9 is used to update the parameters.
Architecture Template		<p>Rol Detection Head: classification and regression branch each consisting of four convolutions with Group Normalisation and Leaky ReLU activation functions. Outputs are generated with pointwise convolutions.</p> <p>Rol Mask Head: mask branch with four convolutions each with Group Normalisation and Leaky ReLU activation functions. A transposed convolution is used to upsample the features. The output is generated via a pointwise convolution. Only used for 2STAGE-MIX model.</p>
Anchor Matching		Rol: Proposals with IoU below 0.2 are considered negative and above 0.3 as positives. Proposals which an IoU between these two thresholds are ignored.
Loss Functions		<p>RPN Classification: Online Stochastic Hard negative mining (1/3 positive and 2/3 negative) in combination with the BCE loss.</p> <p>Rol Classification: The BCE loss is used to train the classification of the proposals. Online Stochastic Hard negative mining (1/8 positive and 7/8 negative) is use for subsampling proposals.</p> <p>Rol Regression: The smooth L1 loss is used to train the proposal refinement.</p> <p>Rol Segmentation: The mask prediction branch is trained via the BCE loss. Only used for 2STAGE-MIX model.</p>
Rol Pooling		<p>Detection Branch: Feature maps processed by the detection branch are pooled via Rol Align to a size of 5x5x5 .</p> <p>Mask Branch: Feature maps for the segmentation branch are pooled via Rol Align to a size of 9x9x9 . Only used for 2STAGE-MIX model.</p>

Figure 4.11: Two-Stage Anchor-based Model Configuration. Visualizes the configuration of the 2STAGE-MIX and 2STAGE-BOX models. Only model-specific configuration options are shown here; model-agnostic options are omitted to provide a better overview. RPN parameters are only shown if they differ from the 1STAGE models to provide an improved overview. Additional dependencies for the parametrization are shown in the second column, and additional details are provided in the third column. Fixed parameters are shown in light blue, rule based parameters are shown in orange and empirical parameters are shown in purple. This figure is adapted from [82].

Training - Direct Set Detection Blueprint

The initial study described in Section 4.1.3 introduced several DETR models which can be used to perform direct set prediction. nnDetection uses the Deformable DETR architecture, which yielded similar results to DINO DETR during the development experiments while having a simpler design. The following sections about nnDetection refer to the Deformable DETR model as SETPREDICT.

Architectural Design: DETR models only use an encoder to extract multi-scale features from images. To keep the results comparable the same encoder design as for the 1STAGE and 2STAGE models is used. Each level uses two convolutions with Instance Normalisation [189] and Leaky ReLU non-linear activation functions. Downsampling is performed via strided convolutions. The deepest four levels are used by the transformer modules for further processing. Absolute positional encoding is used to retain positional information when flattening the features into sequences. The output of each level is followed by a convolution to adjust the number of feature channels. Three transformer blocks, each with Multi-Scale Deformable Attention (Section 2.1.3), Layer Normalisation layers and a MLP are used within the transformer encoder. All positions of the feature maps are considered reference points and used within the transformer encoder. The transformer decoder consists of 6 blocks with Multi-Scale Deformable Attention, Self-Attention, Layer Normalisation and a MLP. It is used to process reference boxes via cross-attention with the help of the encoder features. The same two-stage DETR design is used as in the original publication [147] (Section 2.1.3). Each decoder block predicts a refined version of the reference boxes via non-shared detection heads. The classification branch is trained via the focal loss [35]. A combination of the L1 and GIoU loss [138] is used to train the regression branch. The architectural blueprint is visualised in Figure 4.12.

Trainig Schedule: The model is trained for 100 epochs with 2500 batches per epoch and a batch size of four. The AdamW [196] optimiser is used to perform the parameter updates, and the learning rate is set to $3e - 4$. AMSGrad [209] is enabled during the training. The PolyLR learning rate schedule is used to change the learning rate throughout the training. After initial application to the generalisation pool, convergence issues were noticed on two data sets. An automatic restart mechanism was added to the model, where the learning rate is reduced to $1e - 4$, and a warm-up period is used in the beginning. Model-specific configurations are summarised in Figure 4.13.

Training - Dataloading

Processing entire volumetric images with deep neural networks requires large amounts of VRAM, constituting a major limitation. To reduce the required memory footprint, typical 3D networks are trained with smaller patches, which are extracted from the whole image. The majority of medical detection tasks have target structures that only occupy

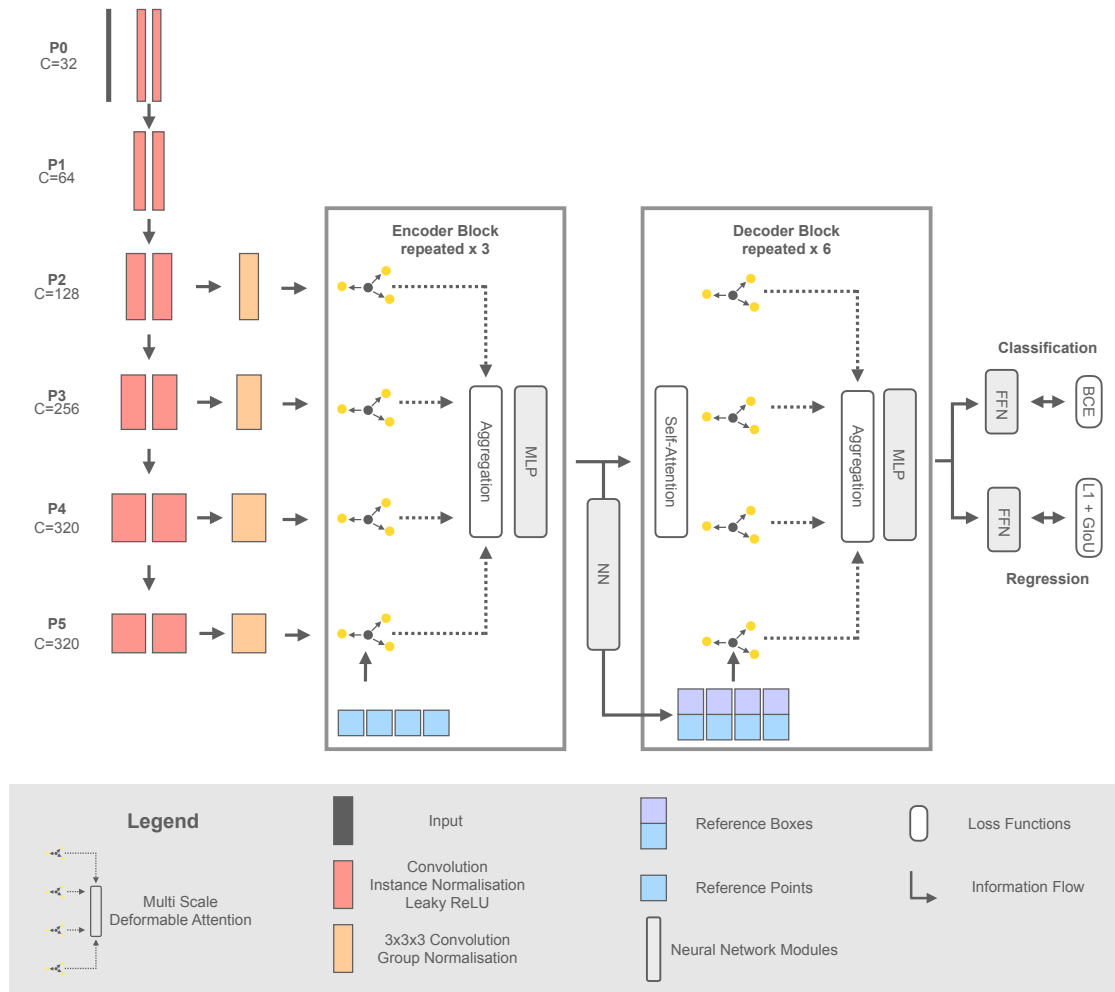


Figure 4.12: Direct Set Prediction Architectural Blueprint. Shows the Deformable DETR model (SETPREDICT) using object-level supervision. The transformer encoder and decoder leverage multi-scale deformable attention to process features and produce the embeddings used for predicting the objects. The decoder uses reference boxes to define reference points and modulate offsets. Two small networks are responsible for processing the transformer decoder output to produce confidence scores and bounding boxes. This figure is adapted from [82].

a small portion of the entire image, introducing a severe imbalance between patches that do not contain objects (background patches) and patches that contain at least one object (foreground patches). To counteract this imbalance, the data loading procedure in nnDetection automatically balances batches by enforcing that 50% of the sampled patches are foreground patches. All models are trained with a batch size of four to balance compute requirements and training stability.

Parameter	Data Fingerprint	Description
Network Topology & Patch Size	Median Image Shape, Target Spacing, GPU Memory Budget	<p>The patch size is initialised with the median image shape after resampling and iteratively decreased until the memory requirements are met. The physical field of view of the patch is kept constant.</p> <p>The network topology defines the depth, kernel sizes and pooling sizes based on the current patch size.</p>
Number of Predictions	Object Positions, Patch Size	Determines the number of objects located within the current patch size, denoted by O_{img} . Number of predictions is set to the 95-th percentile multiplied by 3 and a minimum of 12.
Model Proposal	Annotation Type	Uses Deformable DETR as a direct set prediction model denoted by SETPREDICT.
Optimiser & Learning Rate		Parameter updates are performed via the AdamW optimiser with a learning rate of $3e-4$ and a PolyLR learning rate schedule. In case of convergence issues, an automatic restart is initiated with a reduced learning rate of $1e-4$ and a learning rate warm up from $1e-6$ over 10.000 iterations.
Architecture Template		<p>Encoder: Stacked convolutions with Instance Normalisation and Leaky ReLU non-linear activation functions.</p> <p>Positional Embedding: Absolute positional embeddings based on sine and cosine functions are used to preserve location information of the features.</p> <p>Transformer: The transformer encoder consists of three transformer blocks and the decoder is assembled from six blocks. All blocks use multi-scale deformable attention to save compute. Two stage DETR and iterative bounding box refinement are utilised.</p> <p>Detection Heads: The classification branch consists of a single linear layer to map to the output logits. The regression branch consists of three linear layers where first two layers are followed by a ReLU activation function.</p>
Loss Functions		<p>Classification Branch: Focal loss with alpha set to 0.75 and gamma to 1.</p> <p>Regression Branch: A combination of the L1 loss and the GloU loss are used to train the regression branch. The L1 loss is weighted by 5 and the GloU loss by 2.</p> <p>Auxiliary Losses: Auxiliary losses are computed at each decoder block to refine the predictions.</p>
Empirical Parameter Optimization	Training Data	Inference parameters are derived by empirical optimisation on the training data. Specifically, the IoU threshold for NMS per model, IoU threshold for WBC, a lower limit for the confidence score, a lower limit for object sizes are derived.

Figure 4.13: Direct Set Prediction Model Configuration. Visualizes the configuration of the SETPREDICT model. Only model-specific configuration options are shown here; model-agnostic options are omitted to provide a better overview. Additional dependencies for the parametrization are shown in the second column, and additional details are provided in the third column. Fixed parameters are shown in light blue, rule based parameters are shown in orange and empirical parameters are shown in purple. This figure is adapted from [82].

Some medical detection tasks are not limited to a single foreground class but present multiple classes. This introduces another potential imbalance during training which needs to be considered. Furthermore, not all patients may include objects at all. This requires additional consideration when selecting patients during data loading. To address both concerns, nnDetection performs the selection process on the object-level, where each foreground class is assigned the same sampling probability. For background patches, a random patient and position are sampled.

To ensure a diverse sampling behaviour of foreground objects, an additional offset is introduced for the selection of the centre point of foreground patches:

- When the patch size exceeds the image size, the image is centred in the middle and padded on both sides.
- If the selected foreground object exceeds 70% of the patch size, a random point within the bounding box of the object is selected as the centre.
- In the remaining cases, the object centre is initially used as the patch centre and an additional offset is introduced to vary the position during training. Since the entire object should remain within the patch, the maximum offset is determined by the object- and patch size. A magnitude parameter $M = 0.7$ is introduced as a hyperparameter to adjust the severity of the translation.

In summary, this data loading scheme ensures a balanced sampling of objects, foreground classes and background patches.

Training - Augmentation

Data augmentation is essential for avoiding overfitting and making neural networks robust against image perturbations. nnDetection uses the volumetric augmentation framework Batchgenerators [210] due to its optimised processing pipeline and a broad range of available transformations. The pipeline follows a similar design as nnU-Net [47] with changes in the interpolation granularity and removal of the low-resolution augmentation. All augmentations and their parameterizations are described as follows:

- **[A1] Spatial Augmentation:** This augmentation applies spatial distortions to the images to make neural networks robust against rotations and changes in object size. A random rotation is applied with 20% probability where the rotation angle for each axis is sampled from a uniform distribution $U(-30^\circ, 30^\circ)$. Additional scaling is applied with a probability of 20% and sampled within $U(0.7, 1.4)$. The resulting deformation map is cropped around the centre before applying the interpolation. Linear interpolation is used for the image data, and nearest neighbour interpolation is used for the labels. Lower-order interpolation methods ensure

high throughput during the training and reduce the risk of Central Processing Unit (CPU) bottlenecks.

- **[A2] Pseudo 2D Transformation:** As introduced in Section 4.2.5, applying augmentations along anisotropic axes can lead to severe artefacts and decreased performance. This augmentation is only present if the rule-based parameter is triggered, which ensures that spatial augmentations like rotation, scaling, and mirroring are only applied along the in-plane axes. Furthermore, in this case the rotation angles are sampled within $U(-180^\circ, 180^\circ)$.
- **[A3] Gaussian Noise Transformation:** Additive Gaussian noise is added to the images with a probability of 10%.
- **[A4] Gaussian Blur Transformation:** Gaussian blurring is applied to the images with a probability of 15% where each channel is augmented with a probability of 50%. The standard deviation of the Gaussian kernel is sampled from $U(0.5, 1)$.
- **[A5] Brightness Transformation:** With a probability of 15% a multiplicative brightness augmentation is applied to the image. The scale range is sampled from $U(0.75, 1.25)$.
- **[A6] Contrast Transformation:** The contrast augmentation is executed with a probability of 15% and a magnitude sampled from $U(0.75, 1.25)$.
- **[A7] Gamma Transformation:** Gamma transformation is another intensity augmentation which is applied with a probability of 30% and gamma being sampled from $U(0.7, 1.5)$. An inverted gamma augmentation is applied with a probability of 10%. The original mean and standard deviation of the image are retained in both augmentations.

The augmentation pipeline A[1-7] is applied consecutively to the training images. The online validation images are not augmented.

Inference - Pipeline

Since training and inference are conducted in a patch-wise fashion, it is necessary to post-process the predictions from all models. Figure 4.14 shows an overview of the utilised inference pipeline.

The first step of the inference procedure is the prediction with a sliding window scheme to obtain predictions across the entire image. Objects located at the border of the patches contain less contextual information, which may lead to suboptimal detection performance at the edges of the patch. By using a sliding window scheme with 50% patch overlap, each object can be predicted near the centre of the patch. Additionally, based on the distance of the prediction to the patch centre, an additional weighting factor P_{weight} is

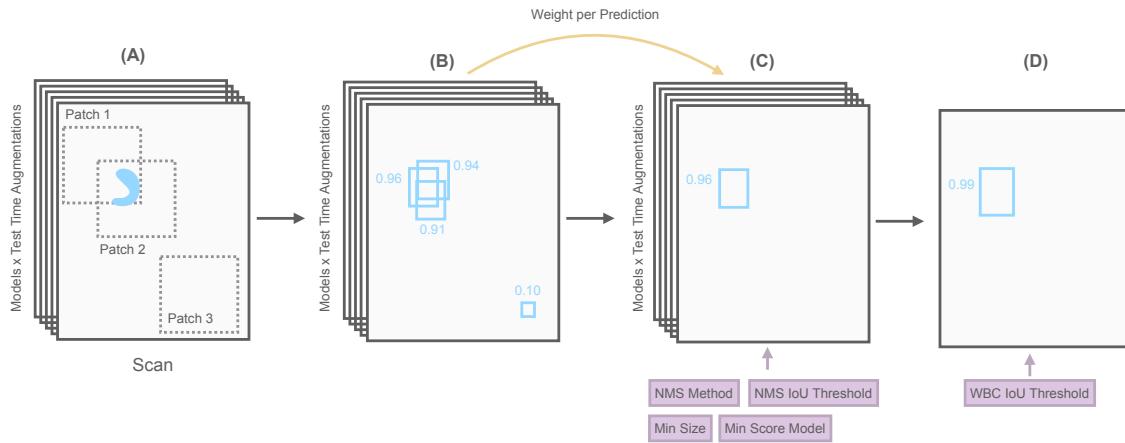


Figure 4.14: Overview of the Inference Pipeline. The inference pipeline can be divided into four steps: (A) a sliding window inference scheme is employed to predict the entire image by all models and test time augmentation transformations. For simplicity, only a small subset of the windows is shown. (B) All predictions are collected and placed correctly into the image space, resulting in multiple predictions per object due to overlapping patches. (C) NMS is used to suppress duplicate predictions, and predictions with small objects or low confidence are removed. (D) Predictions from cross-validation models and different test time augmentations are ensembled and filtered by an optional minimal confidence score.

computed, which can be used during later processing steps. Test time augmentation with mirroring along all axis combinations is used to improve the detection performance. This process is performed for each model from the cross-validation phase. During the second processing step, the predictions of a single model are collected, and NMS is used to suppress duplicate predictions resulting from overlapping patches. Empirical parameter optimisation is used to find the optimal NMS method and IoU, see Section 4.2.7. Two NMS methods are available, one representing the typical functionality and one which uses the previously computed weights P_{weight} to scale the confidence scores of the predictions. The second variant, called *weighted NMS*, assigns higher importance to predictions which are located close to the patch centre. Furthermore, small objects, as well as low-scoring predictions, are removed. During the final step, the predictions across the models and test time augmentations are ensembled via Weighted Box Clustering [46], which performs an averaging across spatial coordinates and confidence scores while incorporating the number of expected predictions.

4.2.7 Empirical Parameters

Some design decisions can not be automatically derived by rules but require adjustment between data sets. Empirical parameters represent these and include the model ensembling step and the parameterization of the inference pipeline.

Inference Pipeline Parametrization

The inference pipeline requires several hyperparameters to suppress duplicates and group similar predictions from different models. These parameters are difficult to estimate from the data characteristics alone but can be efficiently determined by empirically evaluating their performance on the cross-validation predictions. To keep the number of possible parameter combinations reasonable, they are optimised in an iterative fashion starting from a general starting configuration. All parameter ranges, and their respective starting values are shown in Table 4.3.

Table 4.3: Overview of empirical inference parameters. Shows an overview of all empirically optimised inference parameters, including the value range for which they are optimised and the initial configuration. Table reproduced from [82].

Parameter	Range	Starting Value
NMS IoU	$[1 \times 10^{-5}, 0.1, 0.2, 0.3, 0.4, 0.5]$	0.1
NMS Method	["NMS", "wNMS"]	"wNMS"
WBC IoU	$[1 \times 10^{-5}, 0.1, 0.2, 0.3, 0.4, 0.5]$	0.5
Min Size	$[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6]$	$1e-2$
Min Score	$[1 \times 10^{-2}, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0]$	0.0

No additional training runs are needed to perform this empirical parameter optimisation since the network output can be saved once and reused to process the output with different parameterizations.

Training - Model Selection and Ensembling

nnDetection triggers multiple models for training to determine the optimal target spacing and model composition for inference.

Determine best resolution: For data sets with large objects, multiple resolution stages are generated to allow for the efficient detection of large objects. These may span across numerous patches in fine-grained image acquisition protocols. To save upon computational resources, it is usually sufficient to train a single detector type on the different resolution stages, for example, 1STAGE-MIX or 1STAGE-BOX, which offer the shortest training times. Based on the cross-validation results, the best resolution can be determined for the remaining models.

Heterogeneous model ensembling: Section 5.1.3 presented promising results for DETR based detection models showing the potential impact of utilising different detection mechanisms. However, as will be presented in Section 5.2, none of the individual detector types is superior across all tasks and selecting the correct model or model ensemble is critical for achieving the best possible performance. As a consequence, the model proposal stage ([R8] Section 4.2.5) selects one one-stage model, one two-stage model and one direct set prediction model for training. The best individual model or model ensemble is determined based on the cross-validation results and selected for inference on the test set. Our experiments on the test sets of the development pool showed that the best performance can be achieved by using NMS to combine the predictions, as shown in Figure B.4.

4.2.8 nnU-Net Baselines

Current work uses semantic segmentation based approaches for object-level evaluation, either for object detection [64, 46, 18, 211] or instance segmentation [11, 212, 213, 214, 215, 216, 217]. To represent this practice, we build two baselines based on the highly competitive "3d fullres" model of the nnU-Net framework. nnU-Net [47] is a self-configuring semantic segmentation method which can be applied to unseen data sets without manual intervention. It is the only other method which offers out-of-the-box adaptability and represents a state-of-the-art segmentation method.

nnU-Net Basic: represents a common approach for building instances from segmentations. The output probabilities are converted to class maps via the 'argmax' operation, and connected component analysis is used to build object clusters. A confidence score per object is computed by aggregating the values of the probability maps within the objects. All clusters with less than five voxels are removed from the predictions.

nnU-Net Plus: leverages an empirical optimisation scheme to determine task-specific postprocessing parameters. Instead of using the "argmax" operation, an adjustable probability threshold is used to derive the class map. Tasks with more than one foreground class pose an especially difficult problem since objects may be partially predicted as different classes. Besides the approach described for nnU-Net Basic, nnU-Net Plus offers a second clustering mode called "voted". In this scenario, connected component analysis is applied to a binarised version of the class map, and majority voting is used to determine the class of each object. The minimal number of voxels per object, the probability threshold and the aggregation method of the confidence scores are optimised empirically on the cross-validation predictions. An overview of the parameters is shown in Table 4.4.

Task specific changes: During our experiments, convergence problems of the nnU-Net models were observed on three tasks. The smoothing coefficients were removed from the Dice loss for D09 and D10 to ensure proper convergence of the models. This design

Table 4.4: Overview of empirical parameters for nnU-Net Plus. Shows the empirical parameters and their ranges during the optimisation process. Table reproduced from [82].

Parameter	Range
Cluster Mode	["connected", "voted"]
Min. Voxels per Object	[0, 5, 10, 15, 20]
Min. Confidence Score per Object	[0, 0.1, 0.2, 0.3, 0.4, 0.5]
Confidence Score Aggregation	["max", "median", "mean", "95th percentile"]

avoids potential local minima in the loss landscape. D16 required additional design modifications since it contains difficult-to-detect objects, and many scans were acquired from negative patients. The modifications include: (1) patient balancing during the data loading procedure, (2) an offset was added to the object locations as an additional regularisation effect, and (3) the AdamW [196] optimiser with an initial learning rate of $1e - 4$.

CHAPTER 5

Experiments and Results

This chapter presents the results from both the manually and automatically configured detection pipelines, with each section including details about the experimental design and observed results. The findings are interpreted in the scope of their respective study, and interpretations in the broader context are deferred to Chapter 6. The structure of the experiments follows Chapter 4.

Section 5.1.1 shows the effectiveness of large patch size training, adjustments to the augmentation pipeline to reflect the target metric and ensembling of multiple models to boost model performance in competitive settings. Results for our clinical evaluation in the context of vessel occlusion detection are presented in Section 5.1.2. The analysis is performed on the object-level and patient-level to reflect on both diagnostic requirements. A comparison of two CE- and FDA-approved commercial solutions reveals the usefulness of our method. Section 5.1.3 provides the results of the first application of DETR models to lesion and aneurysm detection tasks in volumetric images. It demonstrates the abilities of these models across four data sets against a state-of-the-art anchor-based detection baseline. These results provide the necessary empirical evidence to answer RQ1 Section 1.2.1.

The extensive analysis of our self-configuring medical object detection model is presented in Section 5.2. We demonstrate the advantages of our design across 22 data sets, including the generalisation to previously unseen image modalities, anatomical regions and object structures. Further comparison against current cutting-edge task-specific models is conducted on two benchmarking data sets where nnDetection sets a new record on the

PN9 data set [14]. These results summarise the utility, effectiveness and adaptability of models which are developed under RQ2 Section 1.2.2.

The design of manual detection pipelines is based on [182, 183, 184]. The design of self-configuring medical object detection methods is based on [185, 82].

5.1 Task Specific Design of Object Detection Methods

This section describes three manually configured detection methods to (1) detect mediastinal lesions in CT images, see Section 5.1.1 (2) detect vessel occlusions in CTA images, see Section 5.1.2 and (3) leveraging DETR models in the scope of lesion and aneurysm detection, see Section 5.1.3.

5.1.1 Detecting Mediastinal Lesions in CT Images

Experimental Setup

The official training and validation data sets were combined into a unified training cohort of 880 CT images to perform five-fold cross-validation. This approach generates one model per fold, five models in total, which were used as an ensemble during the inference phase. The primary evaluation metric employed is the FROC score, assessed at $[1/8, 1/4, 1/2, 1, 2, 4, 8]$ FPPI. A prediction was considered correct if it exceeded an IoU threshold of 0.3. Furthermore, the cross-validation experiments were evaluated at an additional IoU threshold of 0.1 and with respect to the mAP metric.

Benchmarking of Mediastinal Lesion Detection Models

Since multiple submissions per day to the leaderboard were allowed, we opted for four submissions in total. Table 5.1 shows cross-validation and leaderboard results from all submitted models.

The baseline model demonstrates a high FROC score of 0.984 during cross-validation, underscoring its good performance on the underlying task. Increasing the patch size during training leads to improvements in the FROC score at both IoU thresholds, although a slight decrease in the AP score is observed at an IoU threshold of 0.1. Adding the Aug B scheme to the training pipeline further enhances performance across all metrics, yielding the best results for coarse detection tasks evaluated at an IoU threshold of 0.1. Moreover, ensembling the two large patch size models, M2 and M3, achieves superior performance at an IoU threshold of 0.3. This improvement directly translates to the

Table 5.1: Shows mAP and FROC results of MELA models. Each row contains the results for a single model, which was evaluated in the cross-validation experiments and submitted to the leaderboard. The best results, obtained from the ensemble model, at an IoU threshold of 0.3 for both FROC and mAP translate directly to the leaderboard results. For coarser IoU values, a single model with LP and Aug B training achieves the best results. Table reproduced from [182].

Model	Config	AP CV		FROC CV		FROC Leaderboard
		IoU 0.1	IoU 0.3	IoU 0.1	IoU 0.3	IoU 0.3
M1	Baseline	0.970	0.961	0.984	0.976	0.9824
M2	LP	0.968	0.962	0.986	0.981	0.9851
M3	Aug B, LP	0.980	0.974	0.992	0.986	0.9851
M2 + M3	Ensemble	0.978	0.976	0.992	0.988	0.9897

challenge leaderboard. The best-performing model achieved an FROC score of 0.9897 on the challenge leaderboard, resulting in the third rank in the MELA challenge.

The FROC curves for the baseline model and the ensemble from the cross-validation experiments are depicted in Figure 5.1.

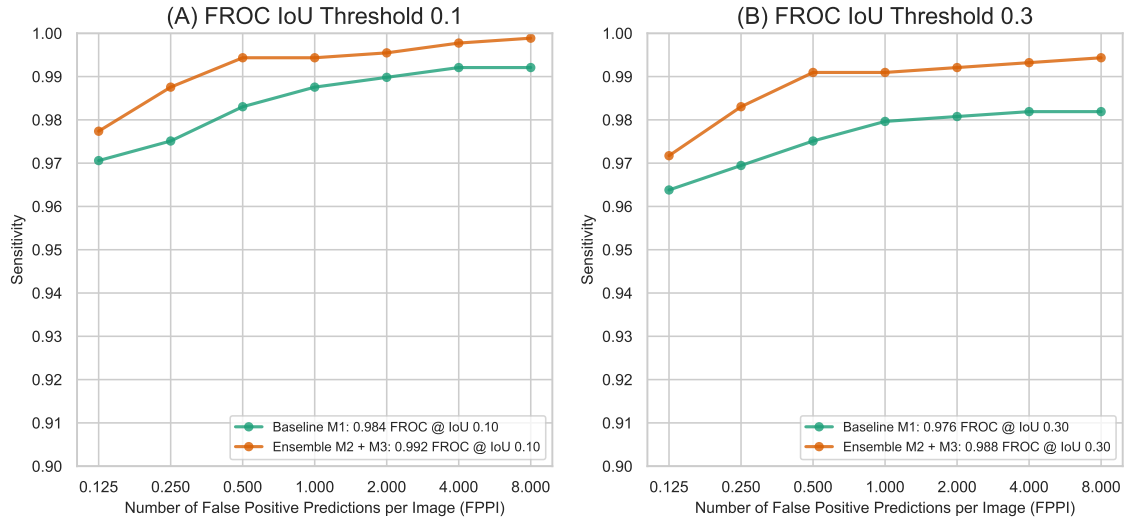


Figure 5.1: Cross-validation results for MELA baseline and final ensemble. (A) The left plot presents the FROC curves for the baseline and ensemble models, evaluated at an IoU threshold of 0.1. (B) The right plot displays the FROC curves for both models, evaluated at an IoU threshold of 0.3.

At an average false positive rate of one per eight images (0.125 FPPI), the baseline model achieves a sensitivity over 97% at the coarse IoU threshold and over 96% at the fine IoU threshold. The ensemble model further improves upon these results, outperforming the

baseline model across all false positive per image thresholds, and achieves a sensitivity above 99% when evaluated at an IoU threshold of 0.1.

5.1.2 Detecting Vessel Occlusions in CTA Images

Experimental Setup

Data Splits: Five-fold cross-validation was used to train an ensemble of Retina Net models on the training split of the UKHD data. The individual data splits were generated randomly with stratification at the patient-level to ensure a similar prevalence of sub-categories in the individual folds. The least frequent class in the data set was selected as the patient-level label for stratification. The UKHD test set served as an internal test set containing unseen images of the same distribution as the training data. Additionally, two external cohorts, FAST and UKB, were collected in a pseudo-prospective manner to demonstrate the generalization capability of the proposed method. The images were acquired from different hospitals and included distribution shifts in the acquisition phase and scanner parameters (for example, prevalence, slice thickness and reconstruction kernel).

Evaluation and Statistical Analysis: The proposed method is evaluated on two granularity levels: (1) **object-level** evaluation is used to assess the number of correctly detected vessel occlusions with respect to the number of false positive predictions and (2) **patient-level** evaluation reflects the ability of the model to differentiate patients with and without vessel occlusions. Bootstrapping with 1000 iterations was applied to derive 95% confidence intervals for the evaluation metrics.

Predictions from the object-level are aggregated to the patient-level by determining the maximum confidence score from all predictions. This evaluation scheme does not require the accurate localization of occlusions, meaning that a correct patient-level prediction could potentially arise from an incorrectly positioned prediction. Patients who presented at least one vessel occlusion were considered occlusion-positive. The primary evaluation metric for patient-level performance is the AUROC. Additional results evaluated at the selected confidence threshold can be found in Section B.1.

For object-level performance, the FROC is used, which averages the sensitivity at pre-defined FPPI thresholds. To adhere to previous studies [43, 14, 173, 11, 12, 19], the FPPI thresholds were set to $1/8$, $1/4$, $1/2$, 1, 2, 4, 8. A prediction was considered correct if the IoU exceeds a threshold of 0.1. This threshold was adopted from other studies [46, 185] reflecting the underlying diagnostic task, which depends on the coarse localization of occlusions. Duplicate predictions of the same vessel occlusion were considered false positive predictions. The clinical problem was formulated as a binary detection task without explicitly differentiating between different types of vessel occlusions. Since false

positive predictions can not be assigned to vessel segments, subgroup analysis on the object-level only considers the sensitivity per group, while all false positives are included in the calculations.

The external cohorts, FAST and UKB, were additionally annotated with HGS labels to extend the evaluation. The annotation of the HGS follows the protocol as the vessel occlusions and are considered if over 70% of the vessel lumen were occluded. When HGS was included, a patient was considered positive if they presented with at least one vessel occlusion or HGS.

Furthermore, two FDA-cleared and CE-marked commercial software solutions, referred to as CS1 and CS2, are included to compare against currently available solutions. The real names of the solutions can not be disclosed at any point in time. No further details regarding the internal design are available due to their proprietary nature. These solutions were compared to the proposed method regarding sensitivity, specificity, PPV, and NPV. A prediction was considered correct if it was located on the correct vessel without requiring precise localization of the occlusion. All patients were evaluated, but only vessel segments supported by the commercial solutions were considered: the internal carotid artery (ICA) and the M1 segment of the middle cerebral artery for large vessel occlusions (LVOs), and in the M2- and M2-segment of the middle cerebral artery for medium vessel occlusions (MeVOs). Statistical significance for sensitivity and specificity was determined using McNemar's test, while a comparison of relative predictive values for PPV and NPV was conducted using the "rpv.test" function from R's DTComPair package.

Benchmarking of Vessel Occlusion Detection Model

Patient and Object Level Performance across Test Sets: The ROC curve for the patient-level results and the FROC curve for the object-level performance across the three patient cohorts are illustrated in Figure 5.2. The proposed method achieved an AUROC of 0.96 [0.95, 0.98] and an FROC score of 0.79 [0.73, 0.84] on the internal UKHD test set. At 0.5 FPPI, an object-level sensitivity of 0.74 [0.67, 0.81] was achieved, which increased to 0.79 [0.73, 0.85] at 1 FPPI. At the determined confidence threshold of 0.647 a sensitivity of 0.73 [0.67, 0.79] was achieved, as illustrated in Table B.4. At this working point, the patient-level sensitivity was 0.94 [0.90, 0.97] (161/172), and specificity was 0.83 [0.77, 0.88] (142/172), see Table B.7. In the first external test set, the FAST cohort, the performance decreased slightly, yielding an AUROC of 0.90 [0.84, 0.94] and an FROC score of 0.75 [0.65, 0.85]. At 0.5 FPPI, an object-level sensitivity of 0.76 [0.64, 0.86] was obtained, increasing to 0.79 [0.68, 0.89] at 1 FPPI. A sensitivity of 0.72 [0.61, 0.83] was evaluated at the determined threshold, as shown in Table B.5. On the patient-level a sensitivity of 0.87 [0.77, 0.95] (45/52) and specificity of 0.77 [0.72, 0.82] (212/274) was observed, see Table B.8. Evaluation on the UKB cohort resulted in an AUROC of 0.85

[0.79, 0.91] and a FROC score of 0.74 [0.66, 0.82]. At 0.5 FPPI, a sensitivity of 0.73 [0.63, 0.82] and at 1 FPPI a sensitivity of 0.76 [0.67, 0.85] was achieved. A sensitivity of 0.71 [0.60, 0.80] was observed at the confidence threshold, as shown in Table B.6. Patient-level sensitivity was 0.81 [0.71, 0.90] (65/80), and specificity was 0.81 [0.75, 0.85] (196/243), see Table B.9.

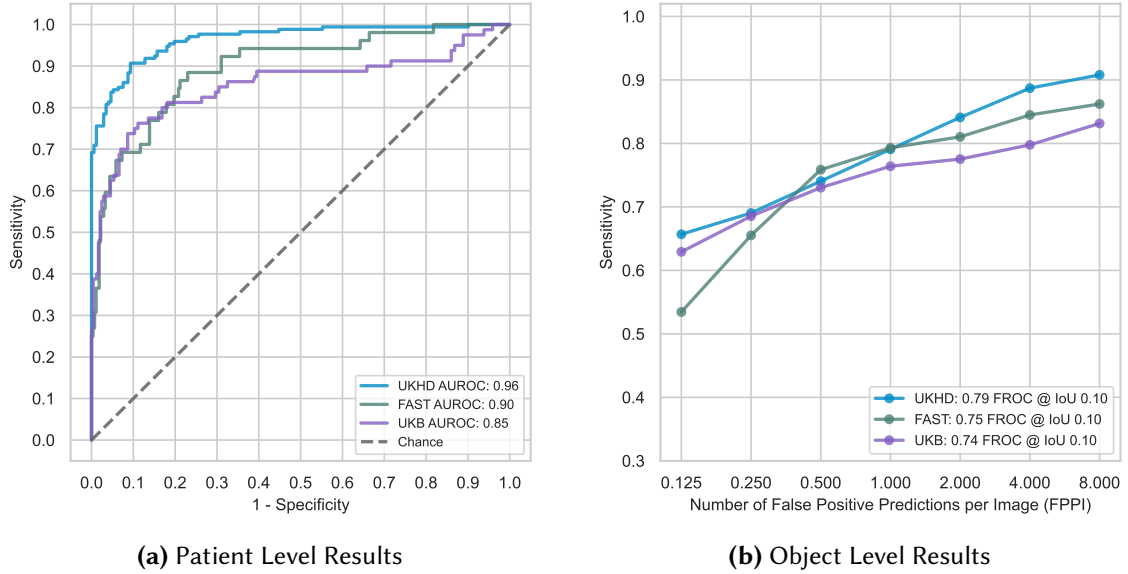


Figure 5.2: Patient and Object level performance of vessel occlusion detection model across Heidelberg, FAST and UKB cohorts. (a) Patient Level Results: The ROC curves show the patient-level performance of the detection model on the Heidelberg, FAST, and UKB cohorts. The AUROC values are 0.96, 0.90, and 0.85 for the Heidelberg, FAST, and UKB cohorts, respectively. **(b) Object Level Results:** The Free-response Receiver Operating Characteristic (FROC) curves illustrate the object-level performance of the model, averaged across seven predefined FPPI thresholds. The model achieves FROC scores of 0.79, 0.75, and 0.74 at an IoU threshold of 0.10 for the Heidelberg, FAST, and Bonn cohorts, respectively. Figure is adapted from [183].

In summary, the best performance was observed on the internal UKHD test set. Since the training data was sourced from the same hospital and scanner type, this test set exhibits the smallest distribution shift. Additionally, the UKHD CTA scan had the smallest slice thickness without acquisition phase shifts, see Table B.1, which may enhance the detection performance of smaller occlusions. When generalizing to the heterogeneous external cohorts, a performance decline was noted, although patient-level performance still remained above an AUROC of 0.85 in both cohorts. Even though all images are resampled to the same spacing before being processed by the neural network, both data sets contain images with greater slice thicknesses, potentially resulting in images that look different from the training set. Moreover, there is a significant shift in the acquisition phases between the Heidelberg test set and the external test sets. While the majority

of the UKHD test set was acquired in the Early Arterial Phase, the majority of FAST scans were acquired in the Peak Arterial and Equilibrium Phases, and the majority of UKB cases were acquired in the Equilibrium and Peak Venous Phase, see Table B.2. The additional venous overlay in these images leads to an increase in false positive predictions, see Table B.10.

Object Level Performance on Sub-Categories: When further subdividing the vessel occlusion classes, as shown in Figure 5.3 for the Heidelberg test set, the highest performance is reached for LVOs in the anterior circulation with an object-level sensitivity of 0.81 [0.74, 0.88] at 0.5 FPPI and a sensitivity of 0.85 [0.78, 0.91] at 1 FPPI.

For MeVOs, a performance drop to a sensitivity of 0.65 [0.53, 0.77] at 0.5 FPPI and a sensitivity of 0.71 [0.60, 0.82] at 1 FPPI is observed. The most challenging occlusions are located in the posterior circulation, where a sensitivity of 0.50 [0.31, 0.7] at 0.5 FPPI and 0.59 [0.4, 0.78] at 1 FPPI is achieved. The performance across the individual subcategories is proportional to their prevalence in the test and training sets, indicating the potential for future extensions of the data set to cover a larger amount of rare occlusion types.

At 1 FPPI, the sensitivity observed across various vascular segments are as follows: 0.20 in the Common carotid artery, 0.79 for Carotid bifurcations, 0.69 in the Internal carotid artery, 0.90 in the Carotid T, 0.96 in the Middle cerebral artery (M1 segment), 0.82 in the Middle cerebral artery (M2 segment), 0.33 in the Middle cerebral artery (M3/4 segment), 0.00 in the Vertebral artery (V1-3 segment), 0.25 in the Vertebral artery (V4 segment), 0.85 in the Basilar artery, 1.00 in the Anterior cerebral artery (A1 segment), 0.36 in the Anterior cerebral artery (A2/3 segment), 1.00 in the Posterior cerebral artery (P1 segment) and 0.00 in the Posterior cerebral artery (P2/3 segment). The majority of vessel occlusions in the Heidelberg test set are located in the M1 segment of the Middle cerebral artery, where 92% of them are correctly detected. In the smaller and more difficult M2 segment 82% of this artery, a sensitivity of 0.82% is reached.

Patient Level Performance with High-grade Stenosis (HGS): During the qualitative assessment of predictions in the FAST and UKB cohorts, multiple false positive predictions were observed at HGS locations. When including HGS as additional detection targets in these cohorts, the patient-level performance improves, as illustrated in Figure 5.4.

In the FAST cohort, patient-level AUROC increased from 0.90 [0.84, 0.94] to 0.92 [0.88, 0.96], and on the UKB cohort, the performance increased from 0.85 [0.79, 0.91] to 0.88 [0.83, 0.93]. HGS, which were previously considered false positive predictions, are now classified as true positive predictions, leading to a general uplift in performance. At the selected confidence cutoff the patient-level specificity increased from 0.77 [0.72, 0.82] to 0.85 [0.80, 0.90] in the FAST cohort and from 0.81 [0.75, 0.85] to 0.88 [0.83, 0.92] in the UKB cohort, see Tables B.8 and B.9. This change in specificity is achieved while retaining the same sensitivity on the FAST cohort and a small sensitivity shift by 1% on the UKB cohort. These results show that although no HGS were annotated in the training data set, the network learned to detect vessels with fading contrasts.

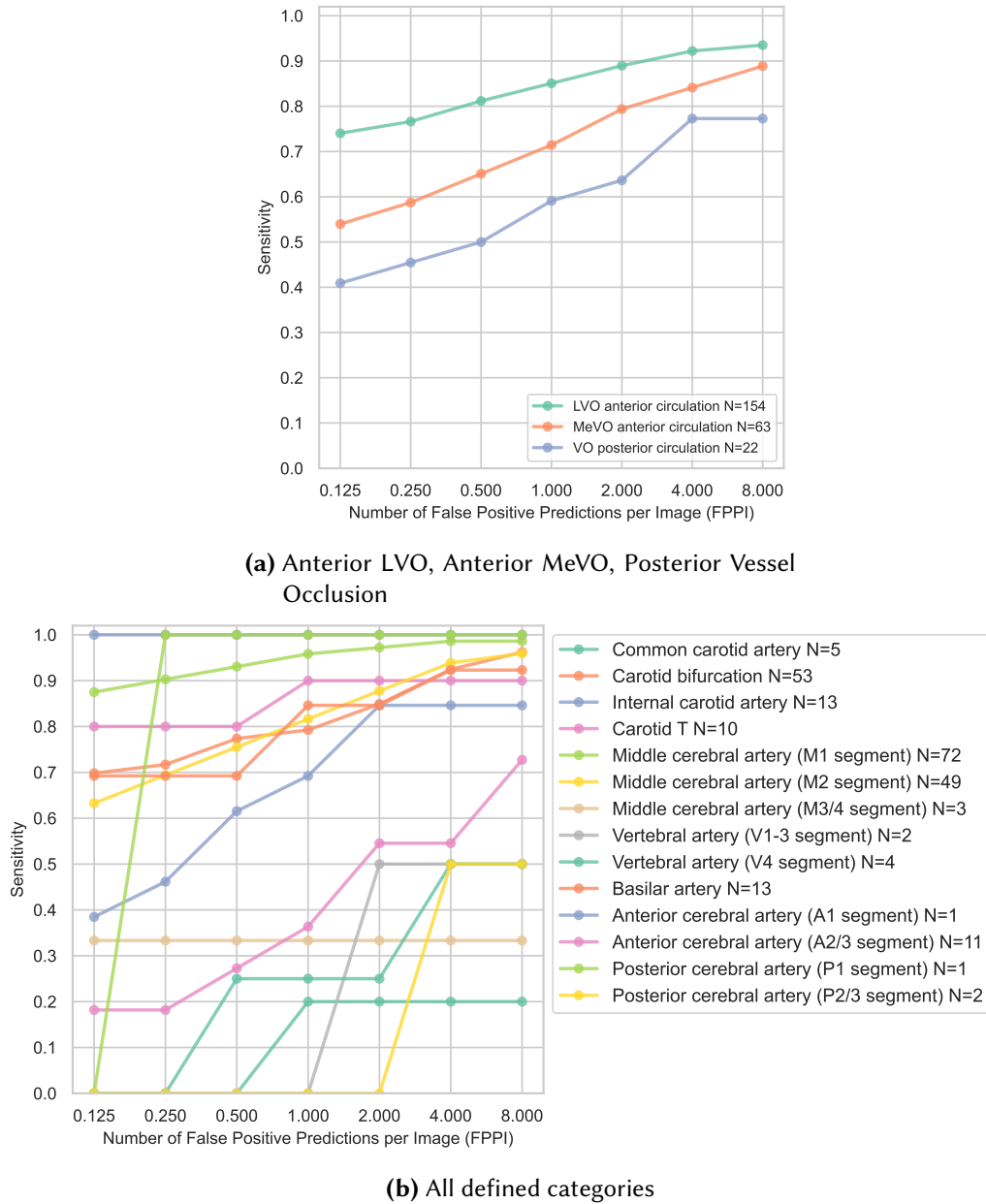


Figure 5.3: Object Level Performance Across Vessel Occlusion Subgroups on the UKHD Test Set. (a) *High-level Categorization*: The highest detection performance is observed for LVOs in the anterior circulation, followed by medium performance for MeVOs in the anterior circulation. A further decline in performance is seen for vessel occlusions in the posterior circulation. The model’s performance correlates with the prevalence of the subcategories. (b) *Fine-grained Categorization*: Performance is shown for the fine-grained categorization of occlusion types, with larger vessel occlusions being easier to detect than smaller ones. This figure is adapted from [183].

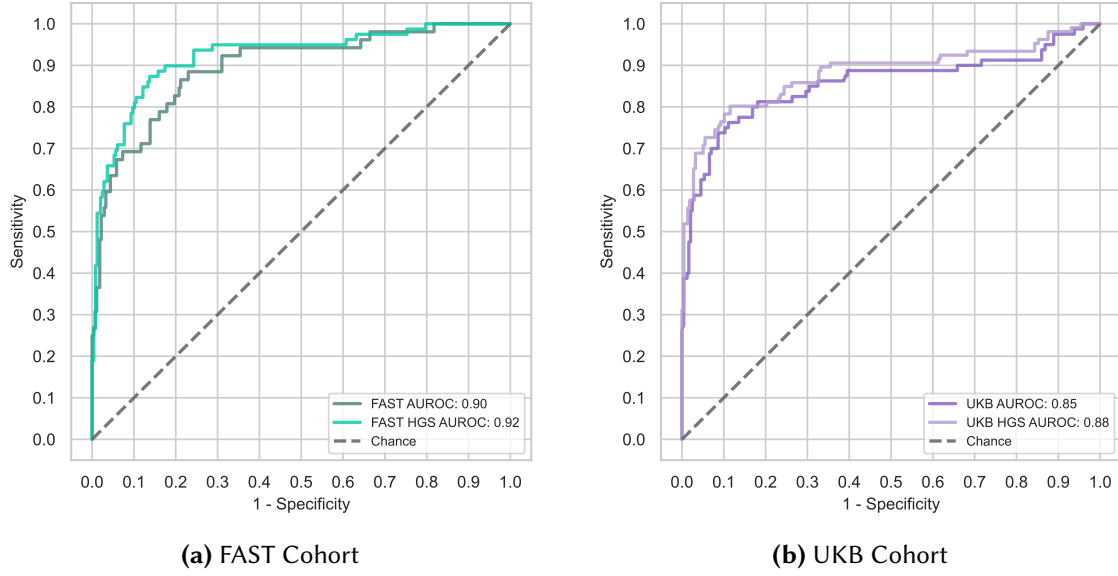


Figure 5.4: Patient-Level Performance on External Cohorts with and without Inclusion of HGS. (a) *FAST Cohort*: The model achieves an AUROC of 0.90 when only evaluating vessel occlusions. The performance increases to an AUROC of 0.92 when HGS are also included in the evaluation. (b) *UKB Cohort*: The model achieves an AUROC of 0.85 when evaluating on vessel occlusions alone. When HGS is included, the AUROC improves to 0.88. This improvement stems from detected HGS, which are not considered false positives anymore. This figure is adapted from [183].

Benchmarking against Commercially Available Solutions: In the UKB cohort, HD-CTA was compared against two commercially available software solutions, as shown in Table 5.2. The proposed method achieved a significantly higher sensitivity of 0.83 [0.72–0.91], compared to 0.38 [0.27–0.51] for CS1 and 0.45 [0.32–0.57] for CS2. The performance difference is particularly evident for MeVOs where HD-CTA reaches a sensitivity of 0.70 [0.50–0.86] while the other methods reach at most 0.19 [0.06–0.38]. Importantly, this increase in sensitivity is observed while achieving a better specificity, 0.87 [0.82–0.90] for HD-CTA, 0.78 [0.73–0.83] for CS1 and 0.84 [0.79–0.88] for CS2. These results show the capabilities of our proposed method, which achieves a significant gain in sensitivity without introducing more false positive predictions.

Due to contractual restrictions, benchmarking in the FAST cohort was limited to a single commercially available solution (CS2). In this cohort, HD-CTA achieved a sensitivity of 0.92 [0.79–0.98], surpassing CS2 with a sensitivity of 0.67 [0.50–0.81] while also demonstrating a higher specificity. Although the performance difference between the compared solutions for LVOs was not significant, HD-CTA showed a significantly better performance for MeVOs.

Table 5.2: Performance Comparison of HD-CTA, CS1 and CS2 on UKB Cohort. Comparison of sensitivity, specificity, PPV, and NPV between HD-CTA, CS1 and CS2 across overall occlusions, LVOs, and MeVOs in the UKB data set. Statistical significance was measured via McNemar’s two-tailed test for sensitivity and specificity, and a comparison of relative predictive values was used for PPV and NPV (rpv.test function of R’s DTComPair package, two-tailed). No correction for multiple comparisons was conducted. P values which are considered significant are bold. Table reproduced from [183].

Software	Sensitivity	Specificity	PPV	NPV
Overall (Occlusions n=65/323)				
HD-CTA (ours)	0.83 [0.72–0.91]	0.87 [0.82–0.90]	0.61 [0.50–0.71]	0.95 [0.92–0.98]
CS 1	0.38 [0.27–0.51]	0.78 [0.73–0.83]	0.30 [0.21–0.42]	0.84 [0.78–0.88]
<i>p value</i>	p < 0.001	p = 0.007	p < 0.001	p < 0.001
CS 2	0.45 [0.32–0.57]	0.84 [0.79–0.88]	0.41 [0.29–0.53]	0.86 [0.81–0.90]
<i>p value</i>	p < 0.001	p = 0.310	p < 0.001	p < 0.001
LVO only (ICA, M1 – n=38)				
HD-CTA (ours)	0.92 [0.79–0.98]	0.87 [0.82–0.90]	0.50 [0.38–0.62]	0.99 [0.96–1.00]
CS 1	0.61 [0.43–0.76]	0.78 [0.73–0.83]	0.29 [0.19–0.40]	0.93 [0.89–0.96]
<i>p value</i>	p = 0.001	p = 0.007	p < 0.001	p < 0.001
CS 2	0.63 [0.46–0.78]	0.84 [0.79–0.88]	0.36 [0.25–0.49]	0.94 [0.90–0.97]
<i>p value</i>	p = 0.002	p = 0.310	p = 0.020	p = 0.002
MeVO only (M2, M3 – n=27)				
HD-CTA (ours)	0.70 [0.50–0.86]	0.87 [0.82–0.90]	0.35 [0.23–0.49]	0.97 [0.93–0.99]
CS 1	0.07 [0.01–0.24]	0.78 [0.73–0.83]	0.03 [0.00–0.12]	0.89 [0.84–0.93]
<i>p value</i>	p < 0.001	p = 0.007	p < 0.001	p < 0.001
CS 2	0.19 [0.06–0.38]	0.84 [0.79–0.88]	0.11 [0.04–0.23]	0.91 [0.86–0.94]
<i>p value</i>	p < 0.001	p = 0.310	p = 0.002	p = 0.002

Inference time measurements on the test set of the Heidelberg cohort showed that the median inference time for HD-CTA is below 2 minutes, specifically 103 s (IQR 83–142 s), which is essential for this time-sensitive application. The distribution of the measured inference time is depicted in Figure 5.5. The primary bottleneck of the pipeline was identified as the pre-processing step, which took 83 s (IQR 67–113 s), rather than the inference step itself, which, when executed on multiple GPUs, took 20 s (IQR 16–28 s). Future work could explore further optimisation of this method by experimenting with other resampling techniques and computer vision libraries.

Further Results: Table B.3 and Table B.7 provide further results from the cross-validation experiments. The FROC score was 0.85 [0.82, 0.87] with a sensitivity of 0.83 [0.80, 0.86] at 0.5 FPPI and 0.87 [0.84, 0.89] at 1 FPPI. 90% [87%–93%] of the LVOs and 84% [77%, 89%] of the MeVOs in the anterior circulation were observed during cross-validation. The patient-level AUROC of 0.96 [0.95, 0.97] was similar to the Heidelberg test set despite

Table 5.3: Performance Comparison of HD-CTA and CS2 on FAST Cohort. Comparison of sensitivity, specificity, PPV, and NPV between HD-CTA and CS2 across overall occlusions, LVOs, and MeVOs in the FAST data set. It was not possible to evaluate CS1 on this cohort due to contractual restrictions. Statistical significance was measured via McNemar’s two-tailed test for sensitivity and specificity, and a comparison of relative predictive values was used for PPV and NPV (rpv.test function of R’s DTComPair package, two-tailed). No correction for multiple comparisons was conducted. P values which are considered significant are bold. Table reproduced from [183].

Software	Sensitivity	Specificity	PPV	NPV
Overall (Occlusions n=39/320)				
HD-CTA (ours)	0.92 [0.79–0.98]	0.85 [0.80–0.89]	0.46 [0.35–0.58]	0.99 [0.96–1.00]
CS 2	0.67 [0.50–0.81]	0.82 [0.77–0.86]	0.34 [0.23–0.45]	0.95 [0.91–0.97]
<i>p-value</i>	p=0.003	p=0.298	p=0.021	p=0.003
LVO only (ICA, M1 – n=26)				
HD-CTA (ours)	0.92 [0.75–0.99]	0.85 [0.80–0.89]	0.36 [0.25–0.49]	0.99 [0.97–1.00]
CS 2	0.85 [0.65–0.96]	0.82 [0.77–0.86]	0.30 [0.20–0.42]	0.98 [0.96–0.97]
<i>p-value</i>	p=0.321	p=0.30	p=0.174	p=0.296
MeVO only (M2, M3 – n=13)				
HD-CTA (ours)	0.92 [0.64–1.00]	0.85 [0.80–0.89]	0.22 [0.12–0.36]	1.00 [0.98–1.00]
CS 2	0.31 [0.09–0.61]	0.82 [0.77–0.86]	0.07 [0.02–0.18]	0.96 [0.93–0.98]
<i>p-value</i>	p=0.004	p=0.298	p=0.006	p=0.004

changes in the distribution of positive and control patients between the training and test sets.

Public Web Interface: The proposed method is available via a public web interface at <https://stroke.ccibonn.ai>, which allows the upload of CTA scans and provides the user with the results via a web-based DICOM viewer. The methodological design of the deployed algorithms follows Section 4.1.2 but was extended with additional utilities to process DICOM data and run computer-aided de-identification (e.g. defacing) of the images. Users can voluntarily donate their data for future development of this method.

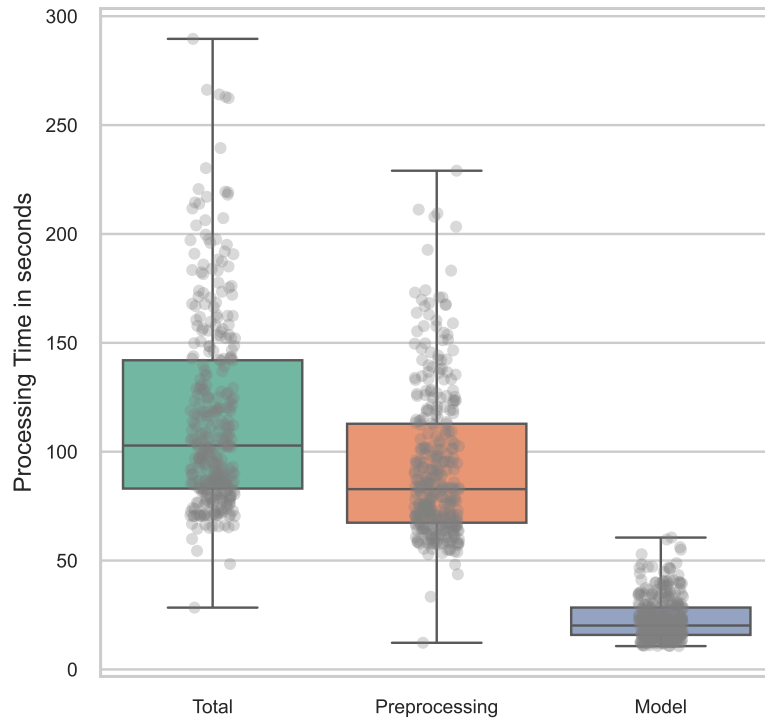


Figure 5.5: Inference Time on Heidelberg Test Set. Box plots showing the distribution of total processing time, pre-processing time, and model inference time in seconds. The centre line represents the median value, while the box spans from the first to the third quartile, and the whiskers extend up to 1.5 times the interquartile range. The scatter points represent individual data samples. This figure is adapted from [183].

5.1.3 Exploring Detection Transformers for Medical Object Detection

Experimental Setup and Evaluation

The selected data sets do not provide official leaderboards, so we performed five cross-validation experiments. Object-level performance is evaluated via the mAP at an IoU threshold of 0.1 and 0.5, which is directly targeted towards coarse and precise localisation tasks. Furthermore, FROC curves with an IoU threshold of 0.1 are provided at 1/8, 1/4, 1/2, 1, 2, 4, 8 FPPI to show an intuitive impression of the performance. 95% confidence intervals are determined via bootstrapping with 1000 iterations where samples are drawn randomly on the patient level.

Benchmarking of Detection Transformers

The performance comparison for the coarse localisation task evaluated at an IoU threshold of 0.1 can be found in Figure 5.6 (a) and table B.11. The original DETR model shows the worst performance of the compared models across all data sets while requiring the longest training schedule. Conditional DETR improves the results for all data sets and uses a shortened training schedule. Compared to Retina U-Net [185] it still shows a performance deficit on three out of four data sets. DINO DETR shows the best performance across all data sets, also outperforming the anchor-based baseline model. Simultaneously, it also uses the shortest training schedule among all of the DETR models. When the IoU threshold is increased to 0.5, as shown in Figure 5.6 (b) and table B.12, all models show reduced performance since predictions which do not fulfil the localisation criterion are now considered false positives. The relative performance gap between DETR and Conditional DETR clearly widens compared to the other two models across all data sets. Figure 5.7 visualizes the performance of the models across seven FPPI working points. DETR and Conditional DETR show decreased performance compared to Retina U-Net across all working points for CADA, KiTS19 and LIDC. On RibFrac, only DETR shows reduced performance, and Conditional DETR achieves the same result as the anchor-based model. DINO DETR demonstrates the best results on LIDC and RibFrac, while Retina U-Net achieves the best results on CADA and KiTS19.

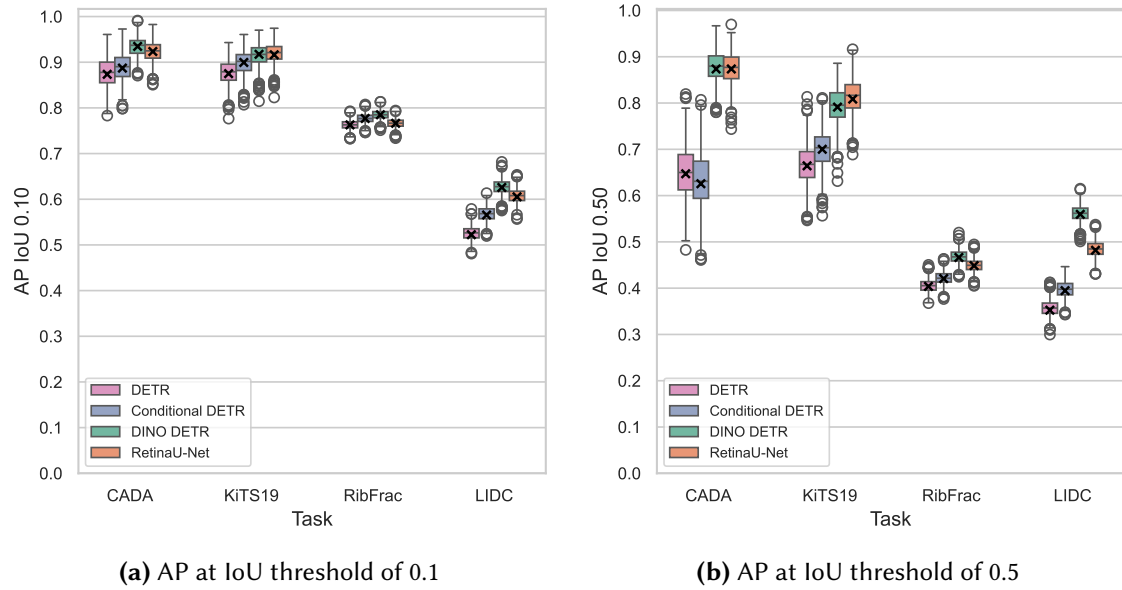


Figure 5.6: mean Average Precision of Detection Transformers on four data sets. Object-level detection performance of detection transformer models on CADA [17], RibFrac [11, 12], KiTS21 [15] and LIDC [13] is shown. Bootstrapping with 1000 iterations was applied to analyse the stability of the metrics. The whiskers span to the 1.5 IQRs of the lower and upper quartile while the boxes extend to the lower and upper quartile. Values outside of these ranges are depicted as points. DETR and Conditional DETR perform worse than the anchor-based baseline on both IoU thresholds while requiring significantly longer training schedules. The performance deficit between these methods to DINO DETR and Retina U-Net widens on all data sets when the IoU threshold is increased. At an IoU threshold of 0.1, DINO DETR shows the best performance across all data sets. Retina U-Net shows similar or better performance on CADA and KiTS19 when the IoU threshold is increased to 0.5.

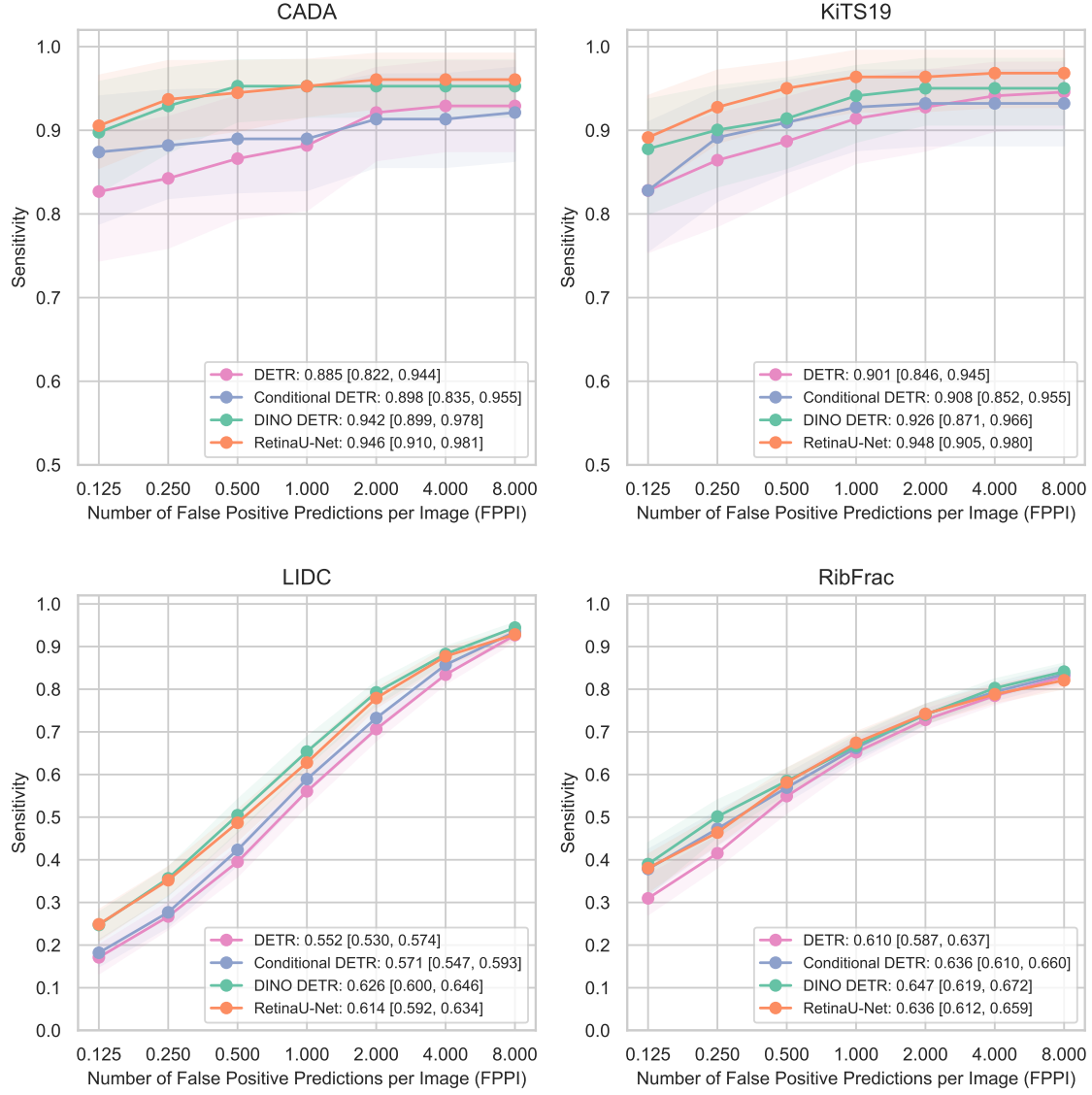


Figure 5.7: FROC curves of Detection Transformers at IoU threshold of 0.1. Shows object-level detection performance of detection transformer models on CADA [17], RibFrac [11, 12], KiTS21 [15] and LIDC [13]. Bootstrapping with 1000 iterations is applied to analyse the stability of the metrics and provide confidence intervals for the curves. DETR and Conditional DETR show the worst performance across all data sets. DINO DETR provides the best performance on RibFrac. Retina U-Net outperforms DINO DETR on CADA and KiTS19.

5.2 Self-Configuring Design of Medical Object Detection Methods

This section presents the results of our self-configuring object detection method nnDetection described in Section 4.2. The evaluation is divided into two parts: (1) The generalisation pool is used to compare against the only other available self-configuring method, named nnU-Net [47], on 9 previously unseen detection tasks. Two baselines, nnU-Net Basic and nnU-Net Plus, are used to represent state-of-the-art semantic segmentation-based detection models. The data sets introduce new object structures, image modalities, and anatomical regions. Furthermore, they are annotated with different annotation types, reflecting the true heterogeneity of the medical detection domain. (2) The benchmarking pool, consisting of three additional data sets, is used to compare nnDetection against current state-of-the-art task-specific models. Evaluation is performed on LUNA16 [43], PN9 [14] and CTA-A [170].

5.2.1 Experimental Setup and Evaluation

The generalisation pool is evaluated with multiple metrics to benchmark different aspects of the proposed method. Ranking-based metrics are used over threshold-based metrics since the working point might vary between tasks, and the primary evaluation metrics aim to develop a general-purpose detection method. The primary metric is the mAP metric at an IoU threshold of 0.1 to focus on the diagnostic performance of the developed detection method. In this scenario, the exact delineation of the target object is less important, and the main objective is to identify the coarse location correctly. To complement this evaluation, ablation experiments with a higher IoU threshold of 0.3 are presented, focusing on the delineation of the objects. The mAP metric does not restrict the number of false positive predictions, which can be exploited by providing a large number of low-scoring predictions aiming to detect the remaining objects and boosting performance in clinically irrelevant areas [149]. The FROC score, evaluated at 1/8, 1/4, 1/2, 1, 2, 4, 8 FPPI, is used as a secondary metric to circumvent this shortcoming. Since all tasks have various numbers of objects and varying difficulties, a ranking-based evaluation scheme is adopted that assigns the same weight to each task. Bootstrapping with 1000 iterations is used to compute the ranking histograms and determine the performance statistics of the methods.

Additional comparative evaluation metrics are used to compare against the strong baseline model nnU-Net Plus. The average relative improvement (in %) is computed for each task with respect to the baseline and averaged across the tasks. The number of improved data sets is measured based on the mAP metric at an IoU threshold of 0.1.

Tasks in the benchmarking pool are evaluated with the official evaluation script to provide a fair comparison against prior work. LUNA16 [43] uses the FROC score averaged over 1/8, 1/4, 1/2, 1, 2, 4, 8 FPPI. Predictions are considered positive if their centre point is located within the radius of the ground truth object. PN9 [14] uses the same evaluation scheme. Ceballos-Arroyo et al. [83] use the CTA-A [170] data set and evaluate the detection performance at an IoU threshold of 0.3. The FROC metric with the same FPPI values is used as in the other studies. We use the same evaluation script with ground truth object coordinates extracted from the unprocessed labels.

The SETPREDICT model did not converge on VALDO-M [20, 21, 22, 23, 24] and PN9 [14] in its original configuration and adjustments to the learning rate were needed to ensure proper convergence, please refer to Section 4.2.6.

5.2.2 Benchmarking on the Generalisation Pool

Evaluation of the generalisation pool consisting of 9 unseen data sets shows that nnDetection achieves the best performance across various metrics. An overview of the aggregated metrics is provided in Figure 5.8. The two segmentation models, nnU-Net Basic and nnU-Net Plus, represent the only other available self-configuring methods and are used as baselines. Further ablation experiments are conducted by evaluating the performance of each model within the nnDetection framework. Rankings for each task are visualised in Figure 5.9, and further details are provided in the following sections.

Comparison against segmentation baselines

nnU-Net Basic, representing the basic post-processing scheme for semantic segmentation methods, ranks the worst among all methods with an average mAP of 59.66. The kidney tumour detection task in D11 is the only task where the model achieves ranks among the best methods. nnU-Net Plus shows better performance on the majority of tasks and achieves an average mAP of 67.68. It is only outperformed by nnU-Net Basic on D14 and D17, which have bounding box annotations and it achieves the same performance on D11. On average, it shows a relative performance improvement of 13.9% over nnU-Net Basic, which shows the importance of proper post-processing parameterization for these methods. All object detection methods in the nnDetection framework show improved performance over nnU-Net Plus, reaching average mAP scores between 72.28 and 74.56. The best individual model across all tasks is the 2STAGE-BOX model, with a mean rank of 2.34. The best performance is achieved by nnDetection with a mean rank of 1.48 and an average mAP score of 74.99. It achieves an average relative performance improvement of 13.2% over the nnU-Net Plus baseline and outperforms it on 7 out of 9 tasks.

5 Experiments and Results

		Mean Rank	Median Rank	Avg. mAP IoU 0.1	Avg. Relative Improv. %	Improved Datasets
		lower better ↓	lower better ↓	higher better ↑	higher better ↑	higher better ↑
Baselines						
nnU-Net Basic		6.03	7	59.66	-13.9	2
nnU-Net Plus		4.51	6	67.68	0.0	0
Ablations						
SETPREDICT		3.43	4	72.28	9.0	6
2STAGE-MIX		2.63	3	74.12	11.9	6
2STAGE-BOX		2.34	2	74.56	12.3	7
1STAGE-MIX		2.94	3	74.10	11.7	7
1STAGE-BOX		3.37	4	73.64	10.8	7
Method						
nnDetection (ours)		1.48	1	74.99	13.2	7

Figure 5.8: Aggregated results from the test sets of the generalisation pool. Shows the performance of the proposed method against two baseline models and five ablation models. Mean and median ranks were computed via bootstrapping with 1000 iterations and collected across all data sets. mAP at IoU 0.10 was computed per data set and averaged in a subsequent step. The relative improvement over the nnU-Net Plus baseline was computed for each data set and averaged afterwards. The improved data sets show the number of data sets where the mAP at IoU 0.10 improved over the nnU-Net Plus baseline. Averaging was performed as a subsequent step to ensure equal weighting of each task. nnDetection (ours) shows the best performance across all metrics. This figure is adapted from [82].

Comparison against ablation models

Further analysis of the results for individual data sets (Figure 5.9) reveals that none of the individual detection models can achieve the best result across all tasks. This shows that different models have different strengths and weaknesses, and selecting an appropriate subset is important. The SETPREDICT model has the lowest performance among the detection-based approaches with an average mAP of 72.28. The 1STAGE models show improved performance with an average mAP of 74.10 for the 1STAGE-MIX model and 73.64 for the 1STAGE-BOX model. The best individual performance is achieved by the 2STAGE models with 74.12 average mAP for the 2STAGE-MIX model and 74.56 for the 2STAGE-BOX model. Ensembling the dynamically selected models achieves the best average performance and is represented by nnDetection.

Variation in Metrics: Changes in the evaluation metric can lead to differences in the rankings shown in Figure 5.10. Exchanging the mAP metric for the FROC metric leads to minor variations in the rankings where the 1STAGE-MIX model outperforms the 2STAGE models. The relative ranking order remains unchanged otherwise. nnDetection remains the leading method even when exchanging the metric.

Detection applications are not limited to diagnostic scenarios, and more fine-grained delineation of the objects might be needed for some tasks. Especially upcoming interactive segmentation models, like SAM [218], offer the potential to convert bounding box predictions into high-quality segmentations. However, more precise bounding boxes might be needed for them to limit the ambiguity in the target structures. To reflect this need, the IoU threshold is increased from 0.1 to 0.3, putting a larger emphasis on the exact delineation of the objects. As shown in Figure 5.10, the rankings change more drastically. Notably, the different detector types remain clustered while the 1STAGE methods are overtaken by the SETPREDICT model. Additional supervision from voxel-level information improved the delineation performance of the 1STAGE and 2STAGE models, indicating that these models can learn to better delineate the boundaries. The lead of nnDetection over the ablation models is reduced, but it remains the overall best-performing method.

Variation in Annotation Type: The generalisation pool is not limited to a single annotation type but includes instance segmentations, bounding boxes and spherical annotations. Different annotation types offer varying localisation granularity, which influences the model's performance. Figure 5.11 visualizes changes in the rankings when evaluating subsets of the generalisation pool. Spherical annotations are considered equivalent to bounding boxes and are thus considered as box-level tasks. The 1STAGE-BOX and SETPREDICT models gain ranks when only considering tasks with box-level annotations. All three top-performing methods are only based on methods utilising box-level supervision. Especially, the 2STAGE-MIX model shows reduced performance, potentially due to the tight integration of voxel-level supervision during training for the RPN and RoI head. nnDetection utilises ensembles from models with box-level supervision and achieves the best rank. When only evaluating tasks with voxel-level annotations, the trend changes, and models that can utilise the additional information gain performance compared to their counterparts. While the 2STAGE models still outperform the other detector types, the MIX models outperform the BOX models. nnDetection selects an ensemble from the 1STAGE-MIX, 2STAGE-MIX, and SETPREDICT models in these scenarios and archives the best overall result.

5 Experiments and Results

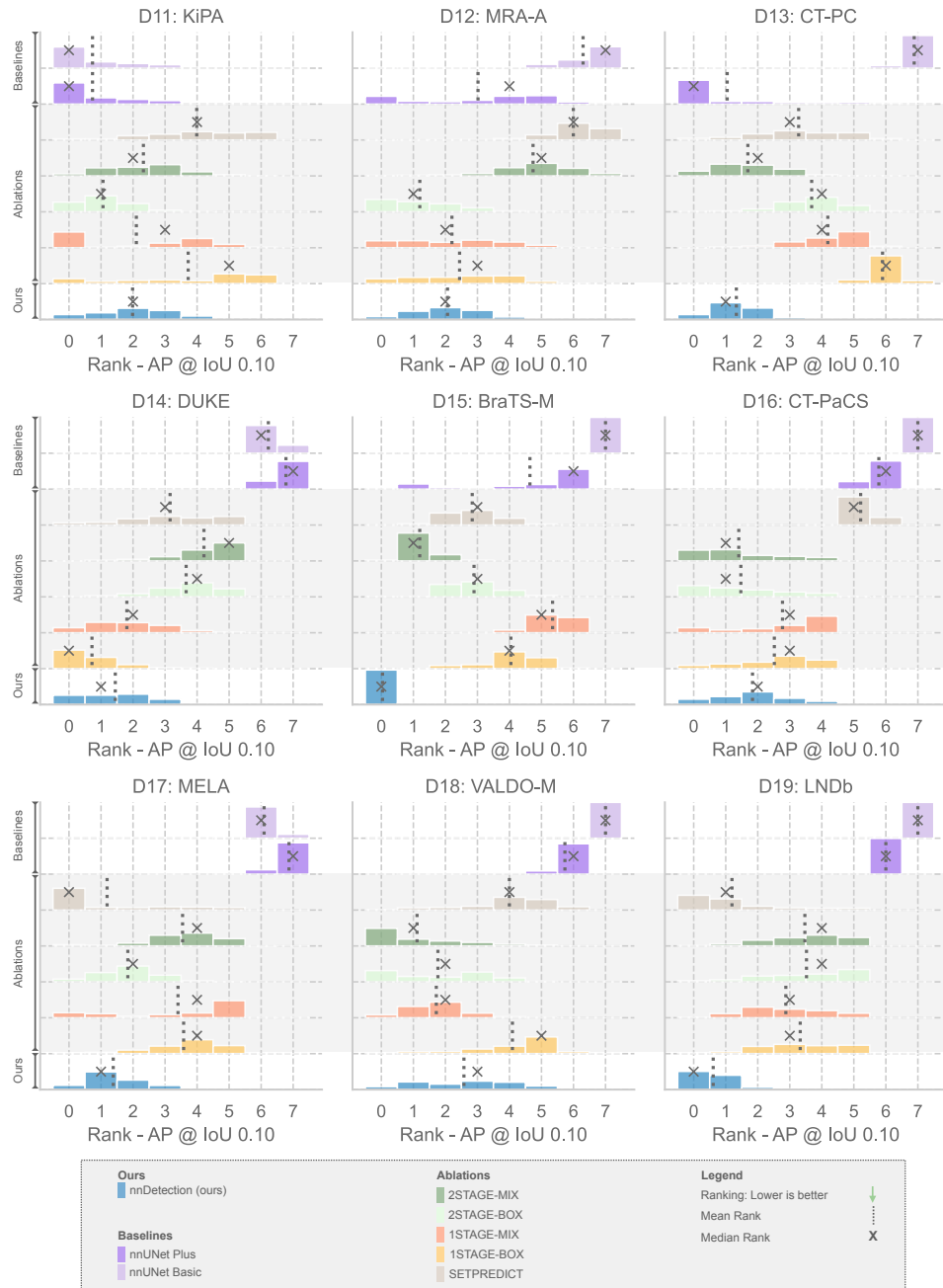


Figure 5.9: Ranking histogram for each test set of the generalisation pool. Shows the ranking of nnDetection, two baselines and five ablation models on the test sets of the generalisation pool. Bootstrapping with 1000 iterations was used with the mAP metric at an IoU threshold of 0.1. Lower rankings correspond to better performance. The dotted line and cross denote the mean and median rank for each method and task, respectively. None of the methods can outperform all competing methods across all data sets. nnDetection performs among the best methods for all data sets. This figure is adapted from [82].

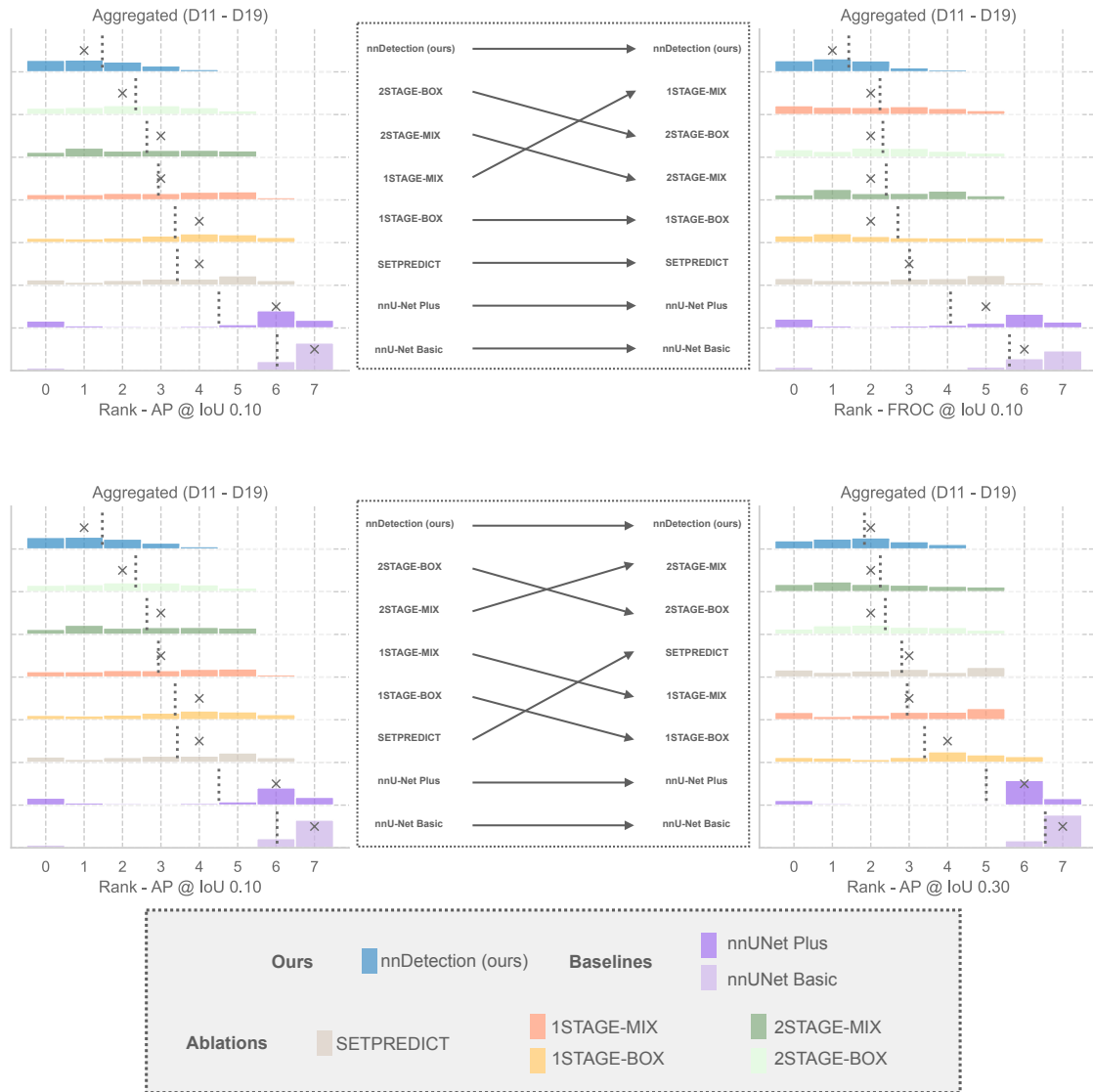


Figure 5.10: Performance comparison with different metrics on the test sets of the generalisation pool. Shows the aggregated rankings of nnDetection, two baselines and five ablation models on the test sets of the generalisation pool. Bootstrapping with 1000 iterations and varying metrics were used to compute the rankings. Lower rankings indicate better performance. The left side denotes the default metric mAP at an IoU threshold of 0.1. The upper row shows changes in the ranking if the FROC metric is used at the same IoU threshold. The lower row shows performance changes when the IoU threshold is increased from 0.1 to 0.3. nnDetection (ours) shows the best performance across all metrics. This figure is adapted from [82].

5 Experiments and Results

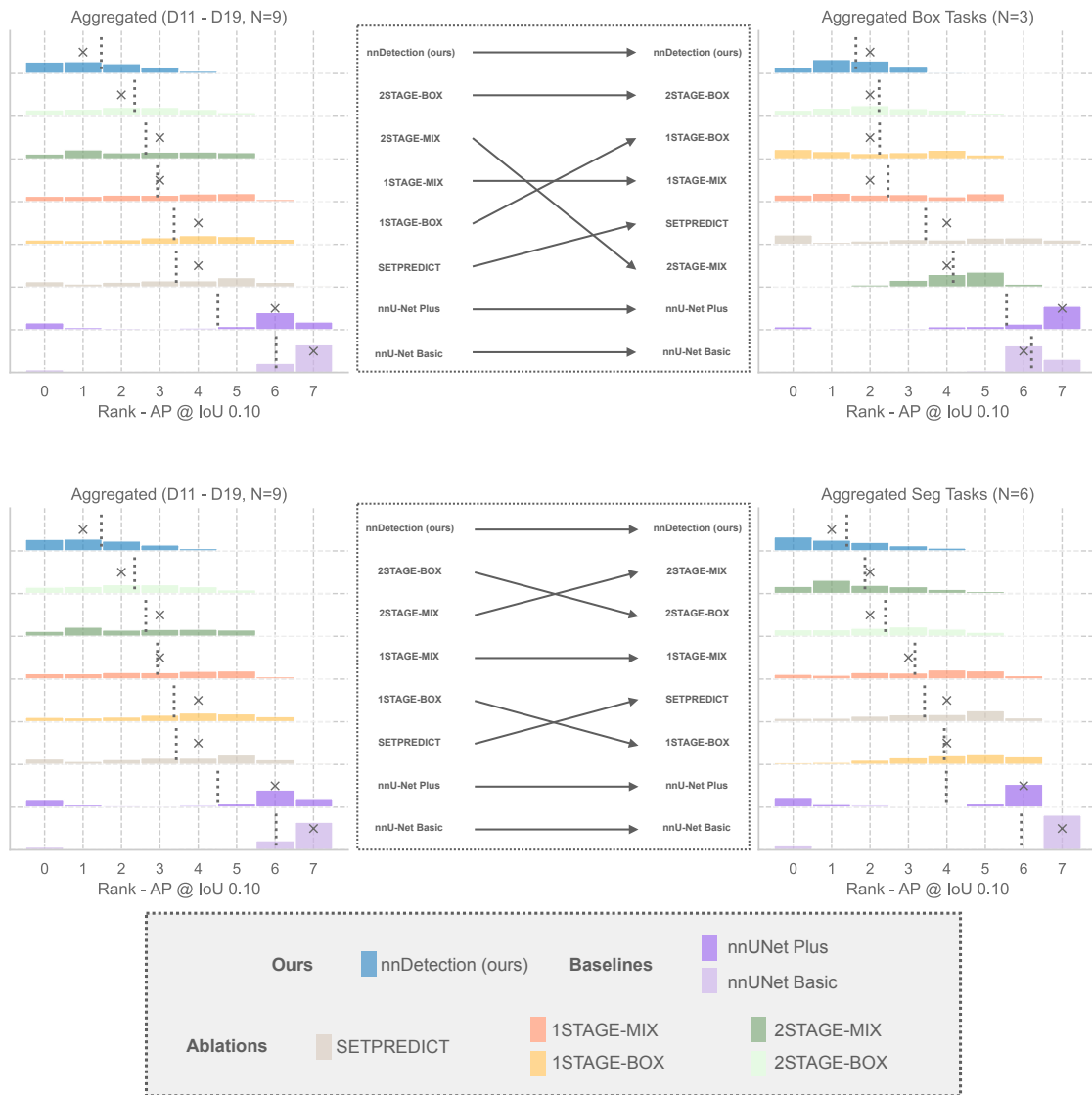


Figure 5.11: Performance comparison with different metrics on the test sets of the generalisation pool. Shows the aggregated rankings of nnDetection, two baselines and five ablation models on the test sets of the generalisation pool. Bootstrapping with 1000 iterations was used with the mAP metric at an IoU threshold of 0.1. Lower rankings indicate better performance. The left side shows the aggregated rankings across all data sets. The upper row shows changes in the results when only data sets with box-level annotations are considered. The lower row visualises differences in the rankings when only data sets with voxel-level annotations are considered. nnDetection (ours) shows the best performance across both annotation types. This figure is adapted from [82].

5.2.3 Benchmarking against Task-specific Methods

Due to the limited availability of generalising detection models, the generalisation pool can only be evaluated with a limited number of models. The benchmarking pools consist of data sets that are used to compare detection methods from prior work.

Benchmarking on LUNA16

This benchmark data set is a subset of the LIDC [13] data set from the development pool of nnDetection but underwent an additional filtering step by the authors of the challenge [43]. The annotations were reduced to spherical labels, and only nodules that were annotated by at least three out of four radiologists were considered positive. The remaining nodule locations are marked as irrelevant, and predictions that are located near them are ignored during the evaluation [43]. Due to the changes in image and annotation characteristics, nnDetection configures a different pipeline, and thus, we consider this as a separate task.

LUNA16 [43] is used by many prior works and constitutes one of the most widely used medical detection data sets to date. The data set is officially divided into 10 subsets, which should be used in a cross-validation fashion. Some publications specifically mention their split, while others only refer to the cross-validation scheme. However, different versions of the split can be used with either eight training folds, one validation fold and one testing fold ('8-1-1' split) or a more classical nine training folds and one testing fold setup ('9-0-1' split) can be utilised for the experiments. The latter will result in overly positive results due to the absence of a dedicated test set. Since many methods did not open source their exact training and evaluation setup, it remains unclear which setup was used by prior work. To provide the best possible comparison, nnDetection is evaluated in both scenarios. Our study includes 18 baseline models, including Liao et al. [78] Harsono et al. [79] Dou et al. [65] Tang et al. [77] Li and Fan [72] Lu et al. [68] Mei et al. [70] Wang et al. [67] Song et al. [74] Gong et al. [174] Ding et al. [66] Luo et al. [69] Khosravan and Bagci [76] Cao et al. [64] Zhu et al. [75] and Liu et al. [73]. A subset of these methods was benchmarked on the LUNA16 data set as part of the study from [70]. Figure 5.12 shows the FROC score for all methods and the FROC curves of the best performing methods. All nnDetection models rank among the best methods, only being outperformed by Liu et al. with an additional FPR stage. The ensembling strategy does not significantly improve the results in these scenarios, likely due to the small validation sets imposed by the official split and the additional labels that are ignored in the official evaluation script but not during training in nnDetection.

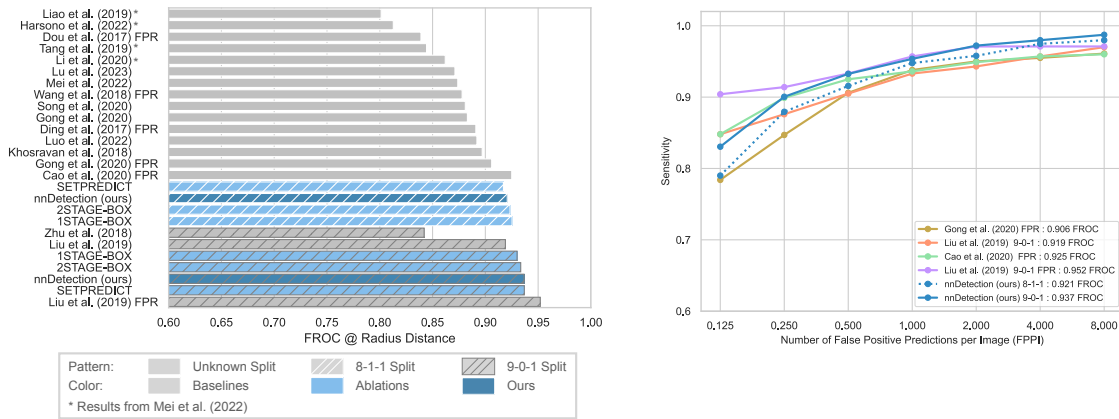


Figure 5.12: Benchmarking results against task-specific models on LUNA16. Shows results of nnDetection (ours) against 16 task-specific models and 3 ablation models. Experiments were conducted with an '8-1-1' and '9-0-1' split. Performance was measured with the official evaluation script using the FROC metric. Predictions were considered positive if the centre point was within the radius of the ground truth annotation. The right side shows the FROC curves of nnDetection and the three best-performing methods. A subset of the baseline results was taken from [70]. This figure is adapted from [82].

Benchmarking on PN9

The PN9 [14] data set is a comparably new benchmark consisting of over 8,000 images and 40,000 object annotations. Official splits are provided for the training, validation and test subset. nnDetection is compared against 11 baseline models, including Lin et al. [35] Ren et al. [36] Liu et al. [118] Liao et al. [78] Zhu et al. [75] Harsono et al. [79] Tang et al. [77] Li and Fan [72] Mei et al. [70] and Xu et al. [71]. The majority of baseline results were taken from [70]. The cross-validation experiments were conducted on the training split, and the validation split was not used at all in this scenario. An additional experiment was conducted with a single SETPREDICT model, which was trained on the training split and empirical parameters determined on the validation split. No further model ensembling is used in this scenario, and the model is denoted as 'SETPREDICT (single)'. This experiment is intended to compare the performance of a single SETPREDICT with previous works which do not utilise model ensembling.

Figure 5.13 provides an overview of all FROC scores and the FROC curve for the best performing methods. All nnDetection models outperform the previous state-of-the-art task-specific models on the test set. Notably, the SETPREDICT (single) model shows better single model performance than the baselines while using significantly fewer computational resources (SANet [70] uses 4 GPUs, and LSSANet [71] uses 3 GPUs). nnDetection establishes a new state-of-the-art on the PN9 test set.

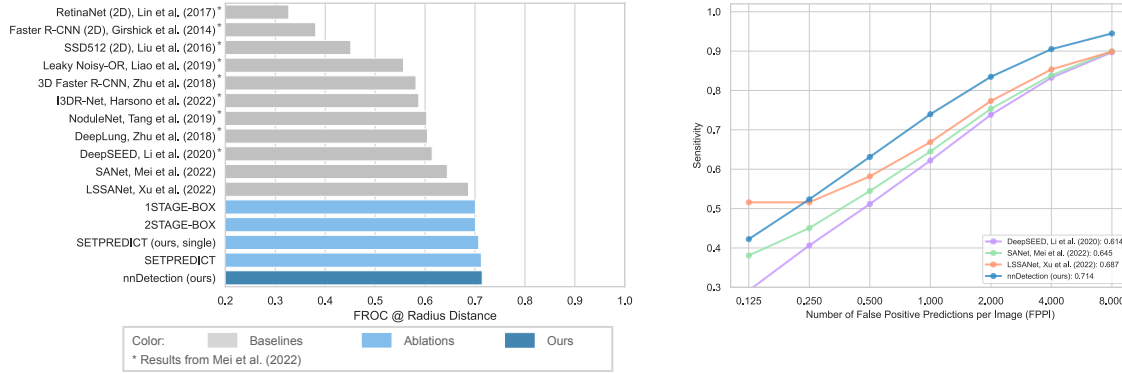


Figure 5.13: Benchmarking results against task-specific models on PN9. Shows results of nnDetection (ours) against 11 task-specific models and 3 ablation models. Experiments were conducted using the official training, validation, and testing splits. Performance was measured with the official evaluation script using the FROC metric. Predictions were considered positive if the centre point was within the radius of the ground truth annotation. An additional ablation experiment was conducted where a single SETPREDICT model was trained and evaluated. The right side shows the FROC curves of nnDetection and the three best-performing methods. The majority of baseline results were taken from [70]. This figure is adapted from [82].

Benchmarking on CTA-A

Detecting aneurysms in CTA scans is a difficult task since the images need to be acquired with fine details, and aneurysms only occupy a tiny portion of the entire scan. The CTA-A data set has two test splits: one internal split, which contains scans from a similar distribution as the training data, and one external test data set. nnDetection models were trained in a cross-validation fashion on the training split and applied to both test sets. The baseline models were established in Ceballos-Arroyo et al. and include Xie et al. [219] Luo et al. [69] and Ceballos-Arroyo et al. [83]. The method from Ceballos-Arroyo et al. [83] is based on the Deformable DETR architecture. Figure 5.14 summarises the results across the two test sets and shows the FROC of the best-performing methods. All nnDetection models outperform the previous state-of-the-art on the internal test set, and 2 out of 3 individual models show better performance on the external test set. nnDetection achieves the best result on the external test set. The SETPREDICT model configured by nnDetection achieves similar or better results than the baseline while requiring fewer computational resources (the baseline model was trained on a node with 4×24 GB VRAM GPUs). Furthermore, no prior information in the form of vessel segmentations is used by our method.

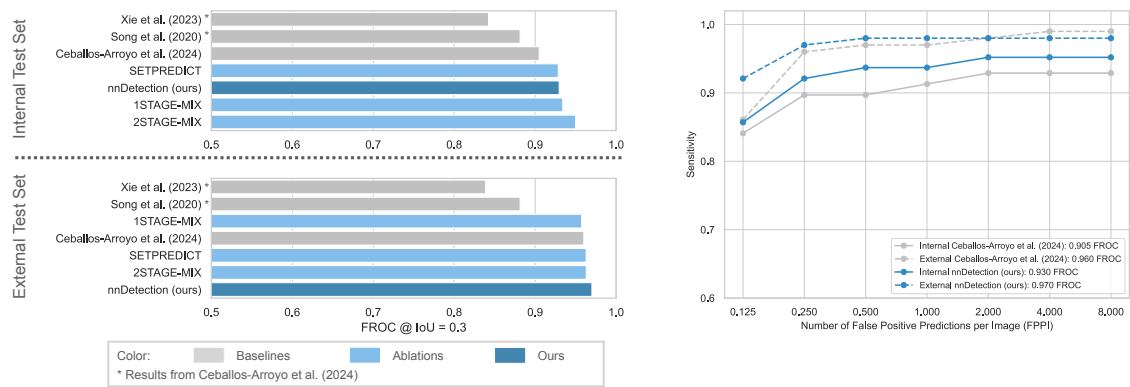


Figure 5.14: Benchmarking results against task-specific models on CTA-A. Shows results of nnDetection (ours) against 3 task-specific models and 3 ablation models on the internal and external test sets. Performance was measured with the publicly available evaluation script using the FROC metric. Predictions were considered positive if their IoU exceeded 0.3. The right side shows the FROC curves of nnDetection and the best-performing task-specific method. Baseline results were taken from [83]. This figure is adapted from [82].

CHAPTER 6

Discussion

This chapter places the results from Chapter 5 into the broader context of the medical image computing domain. In the first part, insights about the manual configuration of detection models are discussed based on the case studies of mediastinal lesion detection, vessel occlusion detection and detection with direct set prediction models. The second part discusses the impact and implications of self-configuring detection models on current development and evaluation practices.

The design of manual detection pipelines is based on [182, 183, 184]. The design of self-configuring medical object detection methods is based on [185, 82].

6.1 Task Specific Design of Object Detection Methods

6.1.1 Detecting Mediastinal Lesions in CT Images

In Section 4.1.1, we outlined our approach for detecting mediastinal lesions for the MELA challenge 2022. Our method relies on multiple Retina U-Net one-stage detection models trained with different configurations to fully leverage available computational hardware resources and adapt to the challenge’s evaluation scheme. We emphasised the refinement of the model configuration rather than introducing novel architectures or detection concepts. Notably, the baseline model already demonstrated excellent detection

performance, achieving a FROC score of 0.9824 on the leaderboard, leaving limited room for further improvement.

Given the large size of the presented mediastinal lesions in the data set, we identified potential stitching artefacts during inference, likely caused by duplicate predictions of the same object from different patches, as a central problem of the baseline. To mitigate this, we used a large patch size of $[192 \times 192 \times 192]$, which reduces the number of sliding window inference patches and thus reduces the probability of introducing such artefacts. Furthermore, the official challenge evaluation is performed at a high IoU threshold of 0.3, which emphasises the need for precise lesion delineation. To incorporate this design decision into our training pipeline, we reduced the spatial augmentation's rotation angles to minimize imprecise bounding boxes. Interestingly, this approach also enhances the coarse localization performance at an IoU threshold of 0.1, yielding the best single-model performance across all cross-validation metrics. Ensembling predictions from multiple models further improves performance at the higher IoU threshold.

Despite these improvements, detecting smaller lesions remains a big challenge, even for the ensembled models. Since the IoU decreases cubically in three-dimensional detection tasks, achieving high IoU values for small lesions is particularly difficult, as minor deviations can already have a substantial impact. Future research could explore more sophisticated methods for accurately delineating such lesions. Visual inspection of cross-validation results also indicated that the developed model can detect suspicious regions located in the lung near the mediastinum. Although considered false positives in the scope of this challenge, they still pose relevant clinical findings. Qualitative results are displayed in Figure 6.1.

In summary, we developed a simple yet highly effective model for detecting mediastinal lesions in CT images. By leveraging a large patch size and a different augmentation scheme, we achieved improvements across the entire FROC curve. The final model achieved a promising FROC score of 0.9897 on the challenge leaderboard, ultimately ranking third in the overall competition.

6.1.2 Detecting Vessel Occlusions in CTA Images

Our proposed method, as outlined in Section 4.1.2, can label any number of occlusions without restrictions on vessel size or anatomical location. This is accomplished through a generic object detection algorithm that directly predicts the position of the vessel occlusion rather than relying on manual heuristics to determine its position. The method operates directly on high-resolution CTA scans with minimal pre-processing, thereby circumventing any dependence on vessel segmentations or maximum intensity projections. The proposed method was developed on a single institutional homogeneous data set collected from the Heidelberg University clinic, where it achieved an AUROC of 0.96 [0.95,

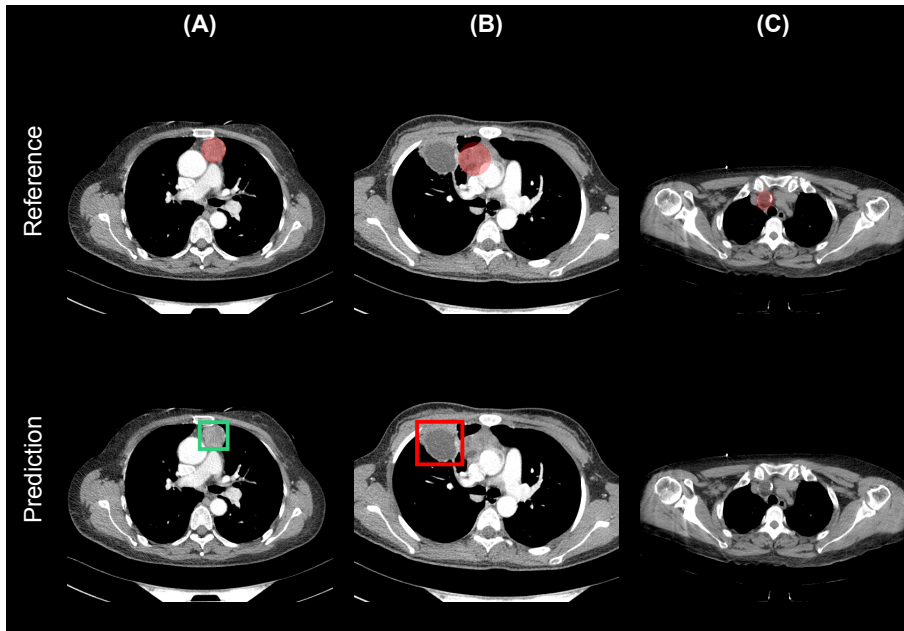


Figure 6.1: Qualitative results of cross-validation results on MELA data set. The first row contains the CT images, and the second row contains the predictions and ground truth annotations. Predictions are shown as bounding boxes with the associated predicted class and confidence score. Ground truth objects are represented by their spherical segmentation. (A) Shows a true positive prediction (green). (B) Depicts a prediction of a suspicious region in the lung (red) and a missed mediastinal lesion. (C) A false negative lesion. This figure is adapted from [182].

0.98] and patient-level sensitivity of 0.94 [0.90, 0.97]. Further evaluations were conducted in two pseudo-prospectively collected cohorts with heterogeneous data incorporating new vendors, scanner models, reconstruction kernels and shifts in the acquisition phases. In the first external cohort, called FAST, we observed slightly degraded results with an AUROC of 0.90 [0.84, 0.94] and patient-level sensitivity of 0.87 [0.77, 0.95]. The second external cohort, from the University Clinic Bonn, showed an AUROC of 0.85 [0.79, 0.91] and patient-level sensitivity of 0.81 [0.71, 0.90]. Given that patients flagged by an automated system will undergo additional clinical review by clinicians, our experiments focused on achieving high sensitivity and NPV. Missing occlusions could lead to reduced prioritisation of patients in clinical practice, which can have detrimental effects in the case of time-sensitive applications like the diagnosis of ischemic stroke. Qualitative inspection of the results showed that the developed model is able to correctly detect HGS, which were not present in the training data set but also constitute clinically relevant findings. Benchmarking against two CE-marked and FDA-approved software solutions demonstrated significantly better sensitivity while being on par or superior in specificity. The largest performance differences were observed for the more difficult-to-detect MeVOs (M2- and M3-segment occlusions).

The reported performance metrics for commercial solutions in the literature vary considerably depending on the cohort and method used for analysis. Specifically, sensitivity/specificity of 82%/90% have been reported for Viz-LVO (Viz.ai), 84%/96% for e-CTA (Brainomix), 93–96%/78% for StrokeViewer (Nico-LAB), 73%/98% for AUTOSTroke LVO (Canon) and 96%/98% for RAPID-LVO (RapidAI) [159, 161, 162, 158, 157]. These solutions are primarily aimed towards the detection of LVOs in the anterior circulation, with reported performance for MeVOs being lower [159, 161]. In contrast, our approach is not limited to individual anatomical regions and can detect vessel occlusion in arbitrary positions, including the posterior circulation and extracranial occlusions. We have demonstrated high patient-level sensitivity across all cohorts exceeding 80%. Commercial software solutions performed below-reported values in literature, potentially due to the difficulty of the external data sets, which comprised heterogeneous data from four hospitals, several scanner types and acquisition phases. The presence of venous overlay in the external cohorts can easily introduce additional false positive predictions and pose an especially difficult problem. Existing deep learning-based approaches to localize vessel occlusions exist in literature but are combined with manual heuristics to derive information about the presence and localization of LVOs. One potential heuristic is based on the identification of vessel asymmetries [220]. These formulations might explain the drastic performance drop since not all occlusions in peripheral vascular territories, such as the M2 segment, are accompanied by a drop in vascular density and can present high interpersonal variability [162, 220, 154, 151]. Our developed approach, which does not depend on such priors, offers a potentially superior task formulation compared to current commercial tools.

Further analysis of the results revealed the promising capability of detecting HGS in CTA scans, suggesting that the developed network is able to detect a reduction in vessel contrast even without a complete blockage. Although not the primary focus of this study, localizing HGS can be an important extension of future solutions since they represent clinically relevant findings, potentially requiring further follow-up treatment by clinicians. Further work will be necessary to reliably detect HGS in CTA scans by collecting additional data and re-training the algorithm.

Qualitative evaluation of the results in the FAST and UKB cohort showed that many false positive predictions made by our method were located on small veins. This is likely attributed to a systematic shift towards delayed acquisition phases, which introduces a venous overlay. In such cases, veins tend to lose contrast, resembling the appearance of contrast loss in arteries. This phenomenon not only challenges automated detection but also complicates manual diagnosis by trained clinicians and thus highlights the need for high-quality acquisition protocols.

Our study on vessel occlusion detection has several limitations. (1) Although we collected a large data set, it still includes significant class imbalance, posing challenges in terms of model training and evaluation of rare occlusion types. Decentralised studies can facilitate

the collection of larger data sets covering a higher number of rare pathologies, which can be used to train more powerful algorithms in the future. The included online web platform, with its crowd-sourcing capabilities, might be a first step in this direction. This scheme could be extended to include other vessel pathologies, such as HGS, to train more general models in the future. (2) The median processing time of our proposed method was 103s, allowing processing of high-resolution CTA scans below 2 minutes. However, further optimisation of the preprocessing pipeline can still significantly reduce the inference time, as our analysis showed that the preprocessing of the images takes up to 80% of the total time. Exploring alternative preprocessing techniques and computer vision libraries could alleviate the current bottleneck. (3) Our study only included a subset of the commercially available software solutions. Expanding this analysis to cover a broader range of solutions, as well as potentially publicly available tools in the future, could provide additional insights into the advantages and disadvantages of current methods. By making our solution available via a web platform, we hope to take the first step in the right direction. (4) Finally, this study only included cohorts which were manually cleaned from artefacts and image corruptions. A potential clinical deployment of this algorithm would require automated checks to ensure sufficient image quality.

In summary, we presented a generic detection method to localize vessel occlusions in CTA scans, which is not limited to any particular anatomical region and does not depend on extensive manual heuristics. Applying the model to two heterogeneous external cohorts revealed promising results and the capability to detect HGS. Benchmarking against two CE-marked and FDA-approved software solutions showed substantially better performance by our proposed method. We made the presented model available via a web platform, which can be found at <https://stroke.ccibonn.ai>.

6.1.3 Exploring Detection Transformers for Medical Object Detection

This study presents the first usage of DETR models for medical object detection tasks on lesions and aneurysms. The empirical results show that the original DETR model requires long training schedules while achieving inferior results compared to state-of-the-art anchor-based models. Introducing conditional (cross-)attention partially helps with long convergence times and provides a slight performance boost. The most complex DETR model, DINO DETR, converges significantly faster and provides the best performance across all models for most tasks. Compared to anchor-based detectors, it does not require manual heuristics for anchors, anchor matching or Non-Maximum Suppression. The training length and number of queries used to control the number of predictions are two of the very few detector-specific parameters of the model.

The DETR and Conditional DETR models require up to four times the number of epochs than the anchor-based detector Retina U-Net [185]. Even though the time per step of

DETR models is shorter than for Retina U-Net, the resulting training times still pose a significant disadvantage. DINO DETR uses additional mechanisms to effectively speed up the training convergence, like deformable multi-scale attention [39, 147], resulting in training times which are comparable to the anchor-based model.

DINO DETR provides the best results of the analysed DETR models, which shows a direct trade-off between model complexity and performance. Simpler models, not relying on multi-scale deformable attention, provide worse results on data sets with smaller objects like CADA and LIDC, while the more complex design of DINO DETR shows clear convergence and performance advantages. DETR and Conditional DETR can use higher resolution feature maps to leverage more fine-grained information but require significantly more VRAM due to the quadratic memory and compute increase of the attention operation with increasing sequence lengths. DINO DETR processes the last three feature maps, resulting in features which are four times more fine-grained along each axis than the features processed by other DETR models. We hypothesize that this is the reason for its improved performance and convergence speed. On the other hand, DINO DETR introduces additional hyperparameters which require expert knowledge and computational resources to be adjusted accurately.

Currently, state-of-the-art medical detection models [46] rely on additional segmentation supervision to achieve the best possible performance. This provides them with additional supervision but leads to long annotation times. DETR models are trained with bounding box supervision, which enables the annotation of large-scale data sets due to the decreased annotation time per object.

Since this study aims to assess the feasibility of DETR models for medical object detection, it is still limited in some aspects: (1) Due to VRAM limitations of modern GPUs, all 3D detection models are trained in a patch wise fashion. During inference, a sliding window scheme is used to avoid border artefacts but this introduces duplicate predictions of the same objects. As a result, additional post-processing steps, like NMS, are still used and require the correct adjustment of IoU thresholds. (2) While this study analysed three different DETR models, many additional extensions are available that promise further improvements in performance and convergence speed. Furthermore, additional ablation experiments will be needed in the future to understand the most impactful design decisions of the DINO DETR model in the medical domain. (3) Our analysis only included four medical detection tasks which is larger than most studies but does not cover the full breadth of the detection domain. Finally, (4) the DETR models were manually tuned for these experiments which limits their potential user base to experts with profound knowledge.

6.2 Self-Configuring Design of Medical Object Detection Methods

Section 4.2 presented the design of nnDetection, the first self-configuring medical object detection method. The empirical evaluation showcases the versatility of the developed method which is able to generalise to unseen modalities, anatomical regions and object structures. Additionally, it extends beyond a single annotation type and incorporates models for training with box-level and mixed supervision. Its design incorporates one-stage, two-stage and direct-set prediction models to generalise across a wide spectrum of volumetric detection tasks. All models are part of a unified framework that offers automated model proposals, heterogeneous model ensembling, and configuration via rule-based, fixed, and empirical parameters. The results presented in Section 5.2 indicate that a single detection model is not able to achieve optimal performance across all detection tasks, and ensembling a diverse set of detector types can alleviate this shortcoming. The presented method’s development and evaluation included 22 data sets, establishing a robust set of detection tasks in the domain. nnDetection outperforms strong segmentation-based methods on the generalisation pool, consisting of 9 previously unseen data sets with an average relative improvement of 13%. Further benchmarking on three data sets against task-specific models shows the benefits of our robust and general design, which achieves new state-of-the-art performance.

A preliminary version of nnDetection was published in 2021 and has collected over 500 stars on Github and nearly 100 forks. The method presented in this thesis is the result of several hundred experiments on the development pool and 5 years of continuous efforts to find the best possible detection models. nnDetection was used as the foundation for several top ranking submissions to international challenges, including ADAM 2020 [44] (1st rank detection track), MELA 2022 [19, 182] (3rd rank), TDSC-ABUS 2023 [221] (2nd rank detection track) and INSTED 2024 [222] (1st rank). By developing an even more general method and including a diverse set of models for ablation experiments, we hope to further accelerate the adoption of detection methods in the community.

Isensee et al. [81] performed a standardised evaluation of recent semantic segmentation methods and observed several pitfalls in the domain. Among them are (P1) the suitable configuration of baseline methods, (P2) the appropriate selection of high-quality data sets as well as a sufficient quantity of tasks and (P3) different reporting practices between publications. During this study, we have seen the emergence of these pitfalls in the detection domain as well. Baseline methods for the PN9 [14] data set include a 3D Faster R-CNN model which ranked among the worst methods in [70]. The 2STAGE-BOX model of nnDetection follows the same design but is able to outperform all existing solutions without relying on other bells and whistles like squeeze-and-excitation blocks [177, 74, 72], dual path blocks [175, 75], slice grouped non-local modules [70] or long short slice-aware modules [71]. This highlights the importance of proper configuration of the

baseline methods and shows the impact of the self-configuring capabilities of nnDetection for future work. Prior work focuses on the development and evaluation of individual detection tasks [83, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 174, 78, 18, 169] which leads to task-specific design choices and limited generalisation of the design. The presented results showcase that evaluation across multiple tasks is essential to guarantee robust decision-making when developing and evaluating methods. The data pools establish a diverse development and evaluation suite that future studies can use to develop general object detection methods on a robust set of tasks. Finally, we also observed potential differences in reporting strategies for the LUNA16 [43] data set, impacting the experimental design and evaluation of our method. By using accessible data sets and making our code for preparing them publicly available, we hope to set the foundation for the next generation of object detection benchmarks.

Currently, all models within nnDetection are trained from scratch for each new task, which may be a limiting factor for small data sets. Foundation models, which learn general representations from a large corpus of data, represent a different paradigm where the model only requires fine-tuning on downstream tasks. In contrast to our approach, the pre-training usually depends on massive amounts of computing resources, so the resulting model is kept constant across different tasks. Exploring these models as future work offers great potential, especially in combination with detection models which do not incorporate extensive prior knowledge like DETR based direct set prediction models. nnDetection offers a great development platform for these use cases by providing an extensive number of baseline models, reference implementations for different detector categories and easy access to many data sets for downstream evaluation. Furthermore, foundation models like the Segment Anything Model (SAM) can also extend the capabilities of nnDetection by processing the produced bounding box to generate instance segmentations, which allow for the voxel-wise delineation of objects.

While the presented results offer a wide range of tasks and generalisation scenarios, there are still some limitations to this study: (1) Our method focuses on volumetric data, which poses an especially difficult problem due to imbalanced data and large images that require careful design of the entire processing pipeline. Nevertheless, the impact of 3D images on patient care is immense due to the relevance and broad applicability of sequences like MRI and CT. (2) All models within the nnDetection framework are designed for GPUs with 11-16GB of VRAM to make the models available to a large community. However, this also limits the capacity of the models and adding further extensions to scale to larger memory budgets can significantly enhance the performance on large data sets like PN9 [14]. (3) The presented generalisation and benchmarking pools are built upon public data and evaluation is performed via the official scripts. This ensures the comparability of studies based on the available data and evaluation parameters but does not impose any limits on the number of test set evaluations. Since the entire data set is publicly available, hyperparameter tuning on the test set is theoretically possible.

In summary, the first self-configuring medical object detection method was presented, and 22 data sets were used during its development and evaluation. It follows in nnU-Net's footsteps and systematizes the configuration process by categorising parameters into rule-based, fixed and empirical. A unified framework leverages one-stage, two-stage and direct-set prediction models in a heterogeneous ensemble. The resulting method can handle commonly used annotation types and generalises to new volumetric detection tasks without manual intervention.

CHAPTER 7

Conclusion

7.1 Summary

This thesis made significant contributions to the field of medical object detection and paved the way for the democratization of deep learning-based detection models. Identifying objects in medical images presents an essential clinical task for radiologists and computer vision algorithms. This thesis made contributions as outline by the research questions from Section 1.2. RQ 1.1 involved the deployment of detection models to an international challenge. This showcased the feasibility of object detection models in the medical domain and highlighted the importance of proper configuration. However, curated data sets only cover a limited number of applications, and generalising them to other hospitals poses a difficult problem during deployment in clinical settings. RQ 1.2 resulted in the development of a specialized model for vessel occlusion detection in CTA images and showed promising generalisation performance to external cohorts. RQ 1.3 investigated the feasibility of direct set prediction approaches via the transformer architecture for medical detection tasks.

All model of RQ1 were manually configured for the underlying task, limiting their usability to a small number of experts. Self-configuring models require a different development paradigm but offer out-of-the-box applicability to previously unseen tasks. RQ 2.1 investigated the availability of detection data sets for building models that can generalise across tasks. Finally, RQ 2.2 established a systematic configuration process for volumetric detection models.

7.1.1 Manual Design of Detection Methods

The first part of this thesis focused on the feasibility of detection models for medical tasks and their manual configuration. For RQ 1.1, a detection model was developed to identify mediastinal lesions in CT images and submitted to an international challenge with great success. Central components of the approach include generating pseudo masks to provide auxiliary supervision during training, scaling the patch size to use the full extent of the available hardware resources, adjusting the augmentation scheme to reflect the task's requirements, and ensembling multiple models. The submitted model achieved a FROC score of 0.9897 on the MELA leaderboard and ranked third in the challenge. A corresponding publication was published as part of the challenge proceedings *Lesion Segmentation in Surgical and Diagnostic Applications* and I orally presented the solution at the virtual workshop.

The clinical deployment of algorithms is not limited to curated data sets and usually incorporates more difficult tasks, like the generalisation to external hospitals. Under the consideration of task-specific requirements, the design of a detection model for vessel occlusions was presented as part of RQ 1.2. The configuration included scaling to the available hardware resources, selecting an appropriate model for the object structure, reducing the number of anchors to match the diversity in object sizes, and decreasing the inference time. The developed model offers a robust design that only involves a minimal set of preprocessing operations, enabling the application to arbitrary occlusion types. The evaluation was performed on a single internal data set and two external cohorts, which included a heterogeneous set of scans. The detection performance correlated with the number of occlusions per category. Scans in the internal test set can be processed with a median processing time of less than 2 minutes. Furthermore, the model was able to detect HGS, which were not annotated in the training data set but constitute a relevant clinical finding. A comparison against two CE- and FDA-approved commercially available software solutions revealed significantly better performance of our model. The performance differences were especially pronounced on more difficult-to-detect structures like MeVOs. Our findings were published in *Nature Communications*, and the model is publicly available as part of a web service with crowdsourcing capabilities.

Many detection methods rely on an anchor-based design, which requires multiple manually defined heuristics, such as anchor size and density, additional post-processing for deduplication, and assignment rules. Direct set prediction models offer an alternative formulation and can train detection models end-to-end. RQ 1.3 investigated the feasibility of these models for diagnostic tasks in the medical domain. Section 5.1.3 presented empirical evidence demonstrating the usefulness of DETR based models. Simpler models like DETR and Conditional DETR are not able to produce competitive results, but more complex models like DINO DETR outperformed a strong anchor-based baseline model on four detection tasks. This work was presented as a full paper at the *German Conference on Medical Image Computing (BVM)* and received the *third rank for the best paper award*.

7.1.2 Self-Configuring Design of Detection Methods

Manually configuring deep learning models poses a difficult problem due to the required expert knowledge, available time and needed computational resources. However, moving away from the current development practice requires two key components: a large number of data sets to cover many different tasks (RQ 2.1) and a different development concept (RQ 2.2).

The medical domain encompasses many different modalities, anatomical regions, and object structures, which need to be reflected throughout the model’s development and evaluation cycle. Additional difficulty is added to the detection domain due to different annotation styles, which are tailored towards the clinical task and available annotation budget. Three data set pools were compiled to address these requirements: the development pool, which consists of 10 data sets; the generalisation pool, which consists of 9 unseen data sets; and the benchmarking pool, which consists of 3 additional data sets. A total of 22 data sets are assembled, which are annotated with instance segmentations, bounding boxes, or spherical annotations. Due to the prevalence of semantic segmentation methods in the domain, multiple clinically relevant data sets were reformulated as detection tasks. The generalisation pool includes previously unseen characteristics like anatomical regions and object structures compared to the development pool. Prior work utilised individual data sets to develop task-specific models, which were assembled into the benchmarking pool, including two widely used data sets LUNA16 [43] and PN9 [14].

Given enough data sets, developing a general detection model required robust performance across multiple tasks. During the development process, the pipeline parameters were divided into rule-based, fixed and empirical groups and adjusted throughout many experiments. Several insights from RQ 1 about the manual design of detection methods were incorporated into the design to ensure robust performance across the development pool. The resulting method, named nnDetection, provides a unified configuration system for one-stage, two-stage and direct-set prediction models. Robust generalisation across tasks is achieved by using a heterogeneous ensemble of different detector types. An heterogeneous set of models is used for ensembling based on the available annotation type to incorporate as much information as possible.

nnDetection showed notable gains over commonly used segmentation-based baselines and multiple ablation models across the generalisation pool. It remained the best-performing method even when varying the metric and performing analysis on annotation type-specific sub-groups. nnDetection sets new state-of-the-art performance on the benchmarking pool, outperforming task-specific models and showing the capabilities of self-configuring methods. A preliminary version was published at *The Medical Image Computing and Computer Assisted Intervention (MICCAI)* conference and included a public code release which has gathered over 500 Github stars and more than 100 forks. An abstract of the method was accepted at the *German Conference on Medical*

Image Computing (BVM), where I presented it as an oral presentation and received the *Best Presentation Award*. Furthermore, solutions based on nnDetection ranked within the top three solutions in several international challenges: ADAM 2020 [44, 223] (1st rank detection track, team "mibaumgartner"), MELA 2022 [224, 182] (3rd rank, team "mibaumgartner"), TDSC-ABUS 2023 [225] (2nd rank detection track, team "Deadluck") and INSTED 2024 [226, 222] (1st rank, team "MIC"). nnDetection, as described in this thesis, is currently in preparation for submission to a high impact journal.

7.2 Outlook

This thesis sets the potential foundation for future research on medical object detection and provides an easy-to-use entry point to the domain. Nevertheless, several directions can be explored in the future to further advance the field.

Extension to Instance Segmentation

Producing voxel-level predictions for each object via deep learning-based models combines the object detection domain with the semantic segmentation domain. This is called Instance segmentation and is necessary to provide exact delineations of individual objects to allow reasoning on the voxel and object levels simultaneously. It is an essential tool for performing clinical measurements, such as assessing the volume of lesions, which rely on information from both levels. Since algorithmic solutions of the object detection and semantic segmentation domain follow different model designs, methods from both domains can be adopted to solve instance segmentation tasks. Bottom-up methods, extend segmentation models by grouping their voxel-level output into regions via heuristics [227, 228, 229, 230]. Top-down methods first generate region proposals and create binary segmentation masks as a secondary step; these include methods presented in this thesis like Mask R-CNN [37]. They first follow an object-centric approach and predict the voxel-level mask as an additional output in a subsequent step. Direct set prediction models, like DETR, offer new possibilities in this domain by merging object-level reasoning with voxel-level information. The Mask (2) Former architecture [231, 232] combines these two aspects and achieves promising performance, for instance segmentation and semantic segmentation in the natural image processing domain. While these methods start to emerge in the medical domain [233] they follow a task-specific development and evaluation scheme. Since the range of applications for instance segmentation is equally broad as for object detection and semantic segmentation, focussing on a generalising design is essential to establish it as a cornerstone in the medical domain and provide these capabilities to a broad audience.

Using an instance segmentation method might only be one direction to explore in this context: foundation models like the Segment Anything Model (SAM) [218] are continuously gaining traction for interactive segmentation scenarios. These models can be prompted via points or bounding boxes to provide segmentations for the respective entity. Given a model capable of producing a segmentation from an initial bounding box prompt, the instance segmentation task is effectively reduced to a detection task. Using the divide and conquer principle, specialized models can be optimised to provide the best possible detection and segmentation separately and combined in a stage-wise pipeline.

The work presented in this thesis follows an object-centric pipeline and evaluation scheme, which sets the foundation for both potential model designs. Furthermore, many of the data sets which are part of the development and evaluation strategy of nnDetection are annotated via instance segmentations. The availability of a diverse pool of data sets for this task significantly reduces the burden of entry for future research.

Is the future static or dynamic?

This thesis presented a self-configuring design for medical object detection models to achieve generalisation across many data sets. This approach aims to find the optimal configuration for any given task, which is characterised by certain properties. In this scenario, no additional data is required, and all models can be trained from scratch, ultimately yielding a **dynamic** expert model for each task. On the other hand, foundation models gain traction by popularising a new development paradigm. Instead of creating a model for each task separately, these models aim to solve many tasks without adaptation. Training foundation models requires large amounts of data and computational resources, yielding a general model which can leverage synergies between tasks. They are able to generate a general representation of the data, which results in a **static** design for all tasks. On a high level, two major approaches can be considered when developing this type of models: supervised and self-supervised models.

Supervised models are trained on massive annotated data sets to learn general representations. These data sets are often created with human-in-the-loop annotation schemes, where model predictions are refined, and the training is performed iteratively. SAM [218] represents one of these foundation models primarily targeted towards the natural image processing domain, and initial efforts are being developed for the medical domain like TotalSegmentator [234, 235] and AbdomenAtlas [236]. The large number of individually annotated medical data sets offers an alternative to individual data sets. Many partially annotated data sets can be pooled together to train a single model, solving all tasks simultaneously [237, 238, 239, 240] and leading to a generalizing model.

Self-supervised models do not rely on annotations, which allows them to be trained on an even larger number of images. The objective of the training varies with different pre-training styles with contrastive pre-training [241, 242], student-teacher models [243, 244,

245] and masked auto encoders [128, 246, 109] representing popular options. Contrastive pre-training aims to generate similar representations for similar images and dissimilar representations for the rest. The construction of similar and dissimilar images varies between approaches and is often driven by augmentations. Student-teacher networks follow a similar scheme. The teacher has more information available than the student, but the student is tasked with reconstructing the teacher’s output. Masked auto encoders formulate self-supervised learning as a reconstruction task by masking out large parts of the image and forcing the network to reconstruct them. The initial paper popularising this approach [128] on large-scale data sets used the transformer architecture, which can leverage additional speed-ups during training since the masking step can be performed directly on the tokens. Extensions to convolutional neural networks [246] have become available and allow scaling these networks, too.

Broad access to foundation models has the potential to revolutionise the medical image computing domain by offering easy-to-use feature extractors which can be embedded into arbitrary downstream models. It remains to be seen whether future models will be dynamic, static, or even a mixture of these, but the incorporation of large-scale data into the model design will substantially impact model robustness and generalisation. Our work, nnDetection offers a well-suited platform for future models, independent of their design since it allows for easy integration of arbitrary feature extractors and seamless rollout to many detection data sets.

7.3 Closing

In conclusion, this thesis presented the development and deployment of volumetric object detection methods across various medical applications. It presented methods that achieved state-of-the-art results in international challenges, analysed the feasibility of new methods for medical tasks, and developed custom solutions for clinically relevant tasks. All of the gathered knowledge was distilled into a unified method called nnDetection, which democratizes the availability of detection methods. The preliminary code release has already attracted many users, and we hope that it will help fuel the next generation of research in this domain.

APPENDIX A

Own Contributions and Publications

A.1 Own Contributions

The following section outlines my contributions to the research questions examined in this thesis.

RQ 1.1: Are detection methods competitive in international benchmarks?

This work describes my submission to the MELA challenge and was published in the challenge proceedings *Lesion Segmentation in Surgical and Diagnostic Applications*. The method **ranked third** in the overall competition. Furthermore, my solution was presented orally at the virtual challenge workshop. I was responsible for developing and submitting the models as well as writing the manuscript.

RQ1.2: Can detection models offer a clinical value over existing solutions for vessel occlusion recognition?

This work was published in *Nature Communications*. My contributions to this work were the development of the model, performing object-level and patient-level evaluation, creating an initial prototype of the web platform, and contributing to the manuscript's writing process.

RQ1.3: Are direct set prediction models beneficial for medical object detection?

This work resulted from Marc Kevin Ickler's master's thesis, which I supervised. My contributions include conceptualisation of the research question, providing the base

framework for the model integration, regular feedback and strategic planing during weekly meetings and writing parts of the submitted manuscript. I presented the work orally at the *German Conference on Medical Image Computing (BVM)* and it received the *third rank for the best paper award*.

RQ2: How can object detection methods be configured automatically?

The preliminary version of this work was published at the *The Medical Image Computing and Computer Assisted Intervention (MICCAI)* conference and includes a public code release. The method, named nnDetection, was used by myself and teams of the MIC department to participate in several challenges: ADAM 2020 (1st rank detection track, team "mibaumgartner"), MELA 2022 (3rd rank, team "mibaumgartner"), TDSC-ABUS 2023 (2nd rank detection track, team "Deadluck") and INSTED 2024 (1st rank, team "MIC"). nnDetection, as described in this thesis, is currently in submission.

My contributions to this work was the development and maintenance of the entire framework, identifying suitable detection tasks, analysing and identifying shortcomings of current models, finding innovative solutions to improve model performance, leading the detection based challenge participations and writing the manuscript.

Supervised Theses

I supervised the thesis from Marc Kevin Ickler titled *Taming Detection Transformers for Medical Object Detection*, which investigated the feasibility of Detection Transformer models in the medical detection domain. The results of this work were published at the German Conference on Medical Image Computing (BVM) and received the *third rank for the best paper award*.

A.2 Own Publications

This section includes all publications that I have first-authored or co-authored, including poster presentations, oral presentations, journal publications, and pre-prints.

First Author Publications

1. Michael Baumgartner, Paul F Jäger, Fabian Isensee, and Klaus H Maier-Hein. "nnDetection: a self-configuring method for medical object detection". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part V* 24. Springer, 2021, pp. 530–539. ISBN: 3030872394
2. Michael Baumgartner, Marc K. Ickler, Paul F. Jäger, Fabian Isensee, Constantin Ulrich, Tassilo Wald, Julius Holzschuh, Balint Kovacs, Partha Ghosh, for the ALFA study, and Klaus H. Maier-Hein. "nnDetection: A Self-configuring Method for Volumetric 3D Object Detection". In: *In preparation* (2025)

3. Gianluca Brugnara, Michael Baumgartner, Edwin David Scholze, Katerina Deike-Hofmann, Klaus Kades, Jonas Scherer, Stefan Denner, Hagen Meredig, Aditya Rastogi, Mustafa Ahmed Mahmutoglu, Christian Ulfert, Ulf Neuberger, Silvia Schönenberger, Kai Schlamp, Zeynep Bendella, Thomas Pinetz, Carsten Schmeel, Wolfgang Wick, Peter A Ringleb, Ralf Floca, Markus Möhlenbruch, Alexander Radbruch, Martin Bendszus, Klaus Maier-Hein, and Philipp Vollmuth. “Deep-learning based detection of vessel occlusions on CT-angiography in patients with suspected acute ischemic stroke”. In: *Nature Communications* 14 (1 2023), p. 4938. ISSN: 2041-1723. DOI: 10.1038/s41467-023-40564-8. URL: <https://doi.org/10.1038/s41467-023-40564-8>
4. Fabian Isensee, Tassilo Wald, Constantin Ulrich, Michael Baumgartner, Saikat Roy, Klaus Maier-Hein, and Paul F. Jäger. “nnU-Net Revisited: A Call for Rigorous Validation in 3D Medical Image Segmentation”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*. Ed. by Marius George Linguraru, Qi Dou, Aasa Feragen, Stamatia Giannarou, Ben Glocker, Karim Lekadir, and Julia A. Schnabel. Cham: Springer Nature Switzerland, 2024, pp. 488–498. ISBN: 978-3-031-72114-4
5. Marc K Ickler, Michael Baumgartner, Saikat Roy, Tassilo Wald, and Klaus H Maier-Hein. “Taming Detection Transformers for Medical Object Detection”. In: *BVM Workshop*. Springer, 2023, pp. 183–188
6. Michael Baumgartner, Peter M Full, and Klaus H Maier-Hein. “Accurate Detection of Mediastinal Lesions with nnDetection”. In: Springer, 2022, pp. 79–85
7. Michael Baumgartner, P Jaeger, Fabian Isensee, and Klaus H Maier-Hein. “Retina U-Net for aneurysm detection in MR images”. In: *Automatic Detection and Segmentation Challenge (ADAM)* (2020)

Co-Author Publications

1. Lena Maier-Hein, Annika Reinke, Patrick Godau, Minu D Tizabi, Florian Buetner, Evangelia Christodoulou, Ben Glocker, Fabian Isensee, Jens Kleesiek, Michal Kozubek, et al. “Metrics reloaded: recommendations for image analysis validation”. In: *Nature methods* 21 (2 2024), pp. 195–212. ISSN: 1548-7091
2. Annika Reinke, Minu D Tizabi, Michael Baumgartner, Matthias Eisenmann, Doreen Heckmann-Nötzel, A Emre Kavur, Tim Rädtsch, Carole H Sudre, Laura Acion, Michela Antonelli, et al. “Understanding metric-related pitfalls in image analysis validation”. In: *Nature methods* 21.2 (2024), pp. 182–194
3. Constantin Ulrich, Fabian Isensee, Tassilo Wald, Maximilian Zenk, Michael Baumgartner, and Klaus H Maier-Hein. “MultiTalent: A Multi-dataset Approach to Medical Image Segmentation”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*. Ed. by Hayit Greenspan, Anant Madabhushi,

- Parvin Mousavi, Septimiu Salcudean, James Duncan, Tanveer Syeda-Mahmood, and Russell Taylor. Springer Nature Switzerland, 2023, pp. 648–658. ISBN: 978-3-031-43898-1. URL: <https://arxiv.org/abs/2303.14444>
4. Kimberley M Timmins, Irene C van der Schaaf, Edwin Bennink, Ynte M Ruigrok, Xingle An, Michael Baumgartner, Pascal Bourdon, Riccardo De Feo, Tommaso Di Noto, Florian Dubost, et al. “Comparing methods of detecting and segmenting unruptured intracranial aneurysms on TOF-MRAS: The ADAM challenge”. In: *NeuroImage* 238 (2021), p. 118216
 5. Amine Yamlahi, Patrick Godau, Thuy Nuong Tran, Lucas-Raphael Müller, Tim Adler, Minu Dietlinde Tizabi, Michael Baumgartner, Paul Jäger, and Lena Maier-Hein. “Heterogeneous model ensemble for polyp detection and tracking in colonoscopy”. In: *EndoCV@ ISBI* (2022)
 6. Viktoria Palm, Tobias Norajitra, Oyunbileg von Stackelberg, Claus P Heussel, Stephan Skornitzke, Oliver Weinheimer, Taisiya Kopytova, Andre Klein, Silvia D Almeida, Michael Baumgartner, et al. “AI-Supported Comprehensive Detection and Quantification of Biomarkers of Subclinical Widespread Diseases at Chest CT for Preventive Medicine”. In: *Healthcare*. Vol. 10. 11. MDPI. 2022, p. 2166
 7. M Jorge Cardoso, Wenqi Li, Richard Brown, Nic Ma, Eric Kerfoot, Yiheng Wang, Benjamin Murrey, Andriy Myronenko, Can Zhao, Dong Yang, et al. “MONAI: An open-source framework for deep learning in healthcare”. In: *arXiv preprint arXiv:2211.02701* (2022)
 8. Saikat Roy, Gregor Koehler, Michael Baumgartner, Constantin Ulrich, Jens Petersen, Fabian Isensee, and Klaus Maier-Hein. “Transformer Utilization in Medical Image Segmentation Networks”. In: *arXiv preprint arXiv:2304.04225* (2023)
 9. T Weikert, PF Jaeger, S Yang, M Baumgartner, HC Breit, DJ Winkel, G Sommer, B Stieltjes, W Thaiss, J Bremerich, et al. “Automated lung cancer assessment on 18F-PET/CT using Retina U-Net and anatomical region segmentation”. In: *European Radiology* 33.6 (2023), pp. 4270–4279
 10. Saikat Roy, Gregor Koehler, Constantin Ulrich, Michael Baumgartner, Jens Petersen, Fabian Isensee, Paul F Jaeger, and Klaus H Maier-Hein. “Mednext: transformer-driven scaling of convnets for medical image segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer Nature Switzerland Cham. 2023, pp. 405–415
 11. Julius Holzschuh, David Zimmerer, Constantin Ulrich, Michael Baumgartner, Gregor Koehler, Rainer Stiefelhagen, and Klaus Maier-Hein. “Combining Anomaly Detection and Supervised Learning for Medical Image Segmentation”. In: *Medical Imaging with Deep Learning, short paper track*. 2023

12. Tassilo Wald, Constantin Ulrich, Fabian Isensee, David Zimmerer, Gregor Koehler, Michael Baumgartner, and Klaus H Maier-Hein. “Exploring new ways: Enforcing representational dissimilarity to learn new features and reduce error consistency”. In: *arXiv preprint arXiv:2307.02516* (2023)
13. Dimitrios Bounias, Michael Baumgartner, Peter Neher, Balint Kovacs, Ralf Floca, Paul F Jaeger, Lorenz Kapsner, Jessica Eberle, Dominique Hadler, Frederik Laun, et al. “Risk-adjusted Training and Evaluation for Medical Object Detection in Breast Cancer MRI”. in: *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*
14. Balint Kovacs, Nils Netzer, Michael Baumgartner, Carolin Eith, Dimitrios Bounias, Clara Meinzer, Paul F Jäger, Kevin S Zhang, Ralf Floca, Adrian Schrader, et al. “Anatomy-Informed Data Augmentation for Enhanced Prostate Cancer Detection”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer Nature Switzerland Cham. 2023, pp. 531–540
15. Balint Kovacs, Nils Netzer, Michael Baumgartner, Adrian Schrader, Fabian Isensee, Cedric Weißer, Ivo Wolf, Magdalena Görtz, Paul F Jaeger, Victoria Schütz, et al. “Addressing image misalignments in multi-parametric prostate MRI for enhanced computer-aided diagnosis of prostate cancer”. In: *Scientific Reports* 13.1 (2023), p. 19805
16. Dimitrios Bounias, Michael Baumgartner, Peter Neher, Balint Kovacs, Ralf Floca, Paul F Jaeger, Lorenz A Kapsner, Jessica Eberle, Dominique Hadler, Frederik Laun, et al. “Object Detection for Breast Diffusion-weighted Imaging”. In: *BVM Workshop*. Springer. 2024, pp. 334–334
17. Constantin Ulrich, Fabian Isensee, Tassilo Wald, Maximilian Zenk, Michael Baumgartner, and Klaus H Maier-Hein. “SEG. A. 2023 Segmentation of the Aorta: Technical Report”. In: ()
18. Jeremias Traub, Till J Bungert, Carsten T Lüth, Michael Baumgartner, Klaus H Maier-Hein, Lena Maier-Hein, and Paul F Jaeger. “Overcoming Common Flaws in the Evaluation of Selective Classification Systems”. In: *arXiv preprint arXiv:2407.01032* (2024)
19. Saikat Roy, Tassilo Wald, Michael Baumgartner, Constantin Ulrich, Gregor Koehler, David Zimmerer, Fabian Isensee, and Klaus Maier-Hein. “Lost in Transformation: Current roadblocks for Transformers in 3D medical image segmentation”. In: ()
20. Andrés Martínez Mora, Michael Baumgartner, Gianluca Brugnara, Maximilian Zenk, Yannick Kirchhoff, Aditya Rastogi, Alexander Radbruch, Martin Bendszus, Clara I Sánchez, Philipp Vollmuth, et al. “Curriculum-learning for Vessel Occlusion Detection in Multi-site Brain CT Angiographies”. In: *Medical Imaging with Deep Learning*. 2024

21. Constantin Ulrich, Catherine Knobloch, Julius C Holzschuh, Tassilo Wald, Maximilian R Rokuss, Maximilian Zenk, Maximilian Fischer, Michael Baumgartner, Fabian Isensee, and Klaus H Maier-Hein. “Mitigating False Predictions in Unreasonable Body Regions”. In: *International Workshop on Machine Learning in Medical Imaging*. Springer Nature Switzerland Cham. 2024, pp. 22–31
22. Mustafa Ahmed Mahmutoglu, Aditya Rastogi, Marianne Schell, Martha Foltyn-Dumitru, Michael Baumgartner, Klaus Hermann Maier-Hein, Katerina Deike-Hofmann, Alexander Radbruch, Martin Bendszus, Gianluca Brugnara, et al. “Deep learning-based defacing tool for CT angiography: CTA-DEFACE”. in: *European Radiology Experimental* 8.1 (2024), p. 111
23. Tassilo Wald, Constantin Ulrich, Gregor Köhler, David Zimmerer, Stefan Denner, Michael Baumgartner, Fabian Isensee, Priyank Jaini, and Klaus H Maier-Hein. “Decoupling Semantic Similarity from Spatial Alignment for Neural Networks”. In: *arXiv preprint arXiv:2410.23107* (2024)
24. Pedro RAS Bassi, Wenxuan Li, Yucheng Tang, Fabian Isensee, Zifu Wang, Jieneng Chen, Yu-Cheng Chou, Yannick Kirchoff, Maximilian Rokuss, Michael Baumgartner, Ziyang Huang, et al. “Touchstone Benchmark: Are We on the Right Way for Evaluating AI Algorithms for Medical Segmentation?” In: *arXiv preprint arXiv:2411.03670* (2024)

APPENDIX B

Additional Results

B.1 Detecting Vessel Occlusions in CTA Images

Table B.1: Scanner, Convolution Kernel and Slice Thickness Details Across Vessel Occlusion Cohorts. Table reproduced from [183].

Scanner Model	Heidelberg Cohort (n=1179)	FAST Cohort (n=327)	UKB Cohort (n=323)
Siemens SOMATOM Definition AS	1136 (96%)	124 (38%)	0 (0%)
Siemens SOMATOM X.cite	0 (0%)	36 (11%)	0 (0%)
Siemens Sensation 16	31 (3%)	0 (0%)	0 (0%)
Siemens Sensation 40	0 (0%)	167 (52%)	0 (0%)
Siemens Sensation Open	7 (1%)	0 (0%)	0 (0%)
Siemens SOMATOM Definition Flash	4 (0%)	0 (0%)	0 (0%)
Siemens SOMATOM Definition	1 (0%)	0 (0%)	0 (0%)
Philips IQon – Spectral CT	0 (0%)	0 (0%)	323 (100%)
Convolution Kernel			
B26f	746 (63%)	0 (0%)	0 (0%)
I30f	382 (32%)	0 (0%)	0 (0%)
H30f	31 (3%)	85 (26%)	0 (0%)
Hv40f	0 (0%)	1 (3%)	0 (0%)
B25f	0 (0%)	42 (13%)	0 (0%)
B30f	7 (1%)	0 (0%)	0 (0%)
B31f	0 (0%)	40 (12%)	0 (0%)
Bv40f	0 (0%)	35 (11%)	0 (0%)
D26f	4 (0%)	0 (0%)	0 (0%)
I26f	4 (0%)	124 (38%)	0 (0%)
B46f	3 (0%)	0 (0%)	0 (0%)
B10f	1 (0%)	0 (0%)	0 (0%)
B20f	1 (0%)	0 (0%)	0 (0%)
B	0 (0%)	0 (0%)	323 (100%)
Slice Thickness			
0.75 mm	1110 (94%)	4 (1%)	0 (0%)
0.80 mm	0 (0%)	40 (12%)	0 (0%)
0.6 mm	60 (5%)	0 (0%)	0 (0%)
1.0 mm	8 (1%)	209 (64%)	323 (100%)
1.2 mm	0 (0%)	38 (12%)	0 (0%)
1.5 mm	0 (0%)	36 (11%)	0 (0%)
2.0 mm	1 (0%)	0 (0%)	0 (0%)
Slice Thickness, median [IQR]	0.75 [0.75-0.75]	1.00 [1.00-1.00]	1.00 [1.00-1.00]

Table B.2: Acquisition Phase Across Vessel Occlusion Test Sets. Table reproduced from [183].

	Acquisition phase of the CT angiography in the test sets				
	Early Arterial	Peak Arterial	Equilibrium	Peak Venous	Late Venous
HD cohort	n = 258/344 (75%)	n = 39/344 (11%)	n = 40/344 (12%)	n = 4/344 (1%)	n = 3/344 (1%)
FAST cohort	n = 42/327 (13%)	n = 134/327 (41%)	n = 114/327 (35%)	n = 33/327 (11%)	n = 4/327 (1%)
UKB cohort	n = 32/323 (10%)	n = 73/323 (23%)	n = 133/323 (41%)	n = 65/323 (20%)	n = 21/323 (7%)
p-values	<0.001	<0.001	<0.001	<0.001	<0.001

Table B.3: Object-level results in the training set (cross-validation) of the Heidelberg cohort. Abbreviations: VO=vessel occlusion; LVO=large vessel occlusion, MeVO=medium vessel occlusion, FROC=free-response operating characteristic, S=sensitivity, FFPI=Number of false positives per image, VO = vessel occlusion, n=Number of VO. Table reproduced from [183].

Cohort	FROC	S@0.5FPPI	S@1FPPI	S@Thresh
VO (n = 727)	0.85 [0.82, 0.87]	0.83 [0.80, 0.86]	0.87 [0.84, 0.89]	0.85 [0.82, 0.88]
Anterior Circulation				
LVO (n = 456)	-	0.89 [0.85, 0.91]	0.90 [0.87, 0.93]	0.90 [0.87, 0.92]
MeVO (n = 200)	-	0.77 [0.70, 0.83]	0.84 [0.77, 0.89]	0.81 [0.74, 0.86]
Posterior Circulation				
VO (n = 71)	-	0.68 [0.56, 0.78]	0.73 [0.63, 0.84]	0.70 [0.60, 0.80]

Table B.4: Object-level results in the test set of the Heidelberg cohort. Abbreviations: VO=vessel occlusion; LVO: large vessel occlusion, MeVO: medium vessel occlusion; FROC=free-response operating characteristic; S=sensitivity; FFPI=Number of false positives per image; VO = vessel occlusion, n=Number of VO. Table reproduced from [183].

Cohort	FROC	S@0.5FPPI	S@1FPPI	S@Thresh
All (n = 239)	0.79 [0.73, 0.84]	0.74 [0.67, 0.81]	0.79 [0.73, 0.85]	0.73 [0.67, 0.79]
Anterior Circulation				
LVO (n = 154)	-	0.81 [0.74, 0.88]	0.85 [0.78, 0.91]	0.81 [0.74, 0.87]
MeVO (n = 63)	-	0.65 [0.53, 0.77]	0.71 [0.60, 0.82]	0.63 [0.51, 0.75]
Posterior Circulation				
VO (n = 22)	-	0.50 [0.31, 0.7]	0.59 [0.4, 0.78]	0.50 [0.31, 0.7]

Table B.5: Object-level results in the test set of the FAST cohort. Abbreviations: VO=vessel occlusion, LVO: large vessel occlusion, MeVO: medium vessel occlusion, FROC=free-response operating characteristic, S=sensitivity, FFPI=Number of false positives per image, VO = vessel occlusion, n=Number of VO. Table reproduced from [183].

Metrics	FROC	S@0.5FPPI	S@1FPPI	S@Thresh
Full data set				
VO (n = 58)	0.75 [0.65, 0.85]	0.76 [0.64, 0.86]	0.79 [0.68, 0.89]	0.72 [0.61, 0.83]
Anterior Circulation				
LVO (n = 31)	-	0.84 [0.70, 0.96]	0.84 [0.70, 0.96]	0.77 [0.62, 0.91]
MeVO (n = 15)	-	0.87 [0.67, 1.00]	0.87 [0.67, 1.00]	0.87 [0.67, 1.00]
Posterior Circulation				
VO (n = 12)	-	0.42 [0.11, 0.71]	0.58 [0.25, 0.86]	0.42 [0.11, 0.71]

Table B.6: Object-level results in the test set of the UKB cohort. Abbreviations: VO=vessel occlusion, LVO: large vessel occlusion, MeVO: medium vessel occlusion, FROC=free-response operating characteristic, S=sensitivity, FFPI=Number of false positives per image, VO = vessel occlusion, n=Number of VO. Table reproduced from [183].

Cohort n=Num VO	FROC	S@0.5FPPI	S@1FPPI	S@Thresh
VO (n = 89)	0.74 [0.66, 0.82]	0.73 [0.63, 0.82]	0.76 [0.67, 0.85]	0.71 [0.60, 0.80]
Anterior Circulation				
LVO (n = 41)	-	0.83 [0.70, 0.94]	0.88 [0.78, 0.96]	0.83 [0.70, 0.94]
MeVO (n = 29)	-	0.69 [0.50, 0.86]	0.72 [0.55, 0.88]	0.66 [0.48, 0.83]
Posterior Circulation				
VO (n = 19)	-	0.58 [0.33, 0.82]	0.58 [0.35, 0.83]	0.53 [0.29, 0.78]

Table B.7: Patient-level performance of HD-CTA on Heidelberg cohort. Abbreviations: AUROC = Area Under Receiver Operating Characteristic, PPV = Positive Predictive Values, NPV = Negative Predictive Value, VO = Vessel Occlusion. Table reproduced from [183].

Cohort	AUROC	Sensitivity	Specificity	PPV	NPV
Cross-Validation					
VO (n = 628) / Controls (n = 207)	0.96 [0.95, 0.97]	601/628; 0.96 [0.94, 0.97]	159/207; 0.77 [0.71, 0.82]	601/649; 0.93 [0.91, 0.95]	159/186; 0.85 [0.80, 0.90]
Test Set					
VO (n = 172) / Controls (n = 172)	0.96 [0.95, 0.98]	161/172; 0.94 [0.90, 0.97]	142/172; 0.83 [0.77, 0.88]	161/192; 0.84 [0.79, 0.89]	142/153; 0.93 [0.88, 0.96]

Table B.8: Patient-level performance of HD-CTA on FAST cohort. Abbreviations: AUROC = Area Under Receiver Operating Characteristic, PPV = Positive Predictive Values, NPV = Negative Predictive Value, VO = Vessel Occlusion. Table reproduced from [183].

Cohort n = Num Patients	AUROC	Sensitivity	Specificity	PPV	NPV
VO (n = 52)/Controls (n = 274)	0.90 [0.84, 0.94]	45/52; 0.87 [0.77, 0.95]	212/274; 0.77 [0.72, 0.82]	45/107; 0.42 [0.32, 0.51]	212/219; 0.97 [0.94, 0.99]
VO + HGS (n = 79)/Controls (n = 247)	0.92 [0.88, 0.96]	69/79; 0.87 [0.79, 0.94]	209/247; 0.85 [0.80, 0.90]	69/107; 0.64 [0.55, 0.74]	209/219; 0.95 [0.92, 0.98]

Table B.9: Patient-level performance of HD-CTA on UKB cohort. Abbreviations: AUROC = Area Under Receiver Operating Characteristic, PPV = Positive Predictive Values, NPV = Negative Predictive Value, VO = Vessel Occlusion. Table reproduced from [183].

Cohort n = Num Patients	AUROC	Sensitivity	Specificity	PPV	NPV
VO (n = 80)/Controls (n = 243)	0.85 [0.79, 0.91]	65/80; 0.81 [0.71, 0.90]	196/243; 0.81 [0.75, 0.85]	65/112; 0.58 [0.49, 0.67]	196/211; 0.93 [0.89, 0.96]
VO + HGS (n = 106)/Controls (n = 217)	0.88 [0.83, 0.93]	85/106; 0.80 [0.72, 0.87]	190/217; 0.88 [0.83, 0.92]	85/112; 0.76 [0.68, 0.83]	190/211; 0.90 [0.86, 0.94]

Table B.10: Number of false positives on object level in FAST and UKB cohort shown for different acquisition phases. Pearson's chi-squared test was used for comparing the distribution. P-values considered significant are shown in bold. Table reproduced from [183].

	Early Arterial	Peak Arterial	Equilibrium	Peak Venous	Late Venous	p-value
FAST data set						
False Positives [n=48]	3/48 (6%)	18/48 (38%)	25/48 (52%)	2/48 (4%)	0/48 (0%)	p<0.001
False Positives on veins [n=26]	0/26 (0%)	11/26 (42%)	14/26 (54%)	1/26 (4%)	0/26 (0%)	p<0.001
UKB data set						
False Positives [n=44]	1/44 (2%)	6/44 (14%)	17/44 (39%)	12/44 (27%)	7/44 (16%)	p<0.001
False Positives on veins [n=27]	1/27 (4%)	3/27 (11%)	10/27 (37%)	8/27 (30%)	4/27 (15%)	p<0.001

B.2 Exploring Detection Transformers for Medical Object Detection

Table B.11: mean Average Precision at an IoU threshold of 0.1 for DETR models and Retina U-Net across four data sets.

	CADA	KiTS19	LIDC	RibFrac
DETR	0.874	0.875	0.522	0.763
Conditional DETR	0.887	0.899	0.565	0.777
DINO DETR	0.935	0.917	0.626	0.785
RetinaU-Net V1	0.923	0.916	0.605	0.766

Table B.12: mean Average Precision at an IoU threshold of 0.5 for DETR models and Retina U-Net across four data sets.

	CADA	KiTS19	LIDC	RibFrac
DETR	0.647	0.664	0.352	0.404
Conditional DETR	0.626	0.700	0.395	0.421
DINO DETR	0.874	0.791	0.560	0.466
RetinaU-Net V1	0.874	0.808	0.482	0.449

B.3 Self-Configuring Design of Medical Object Detection Methods

B.3.1 Additional Results

B Additional Results

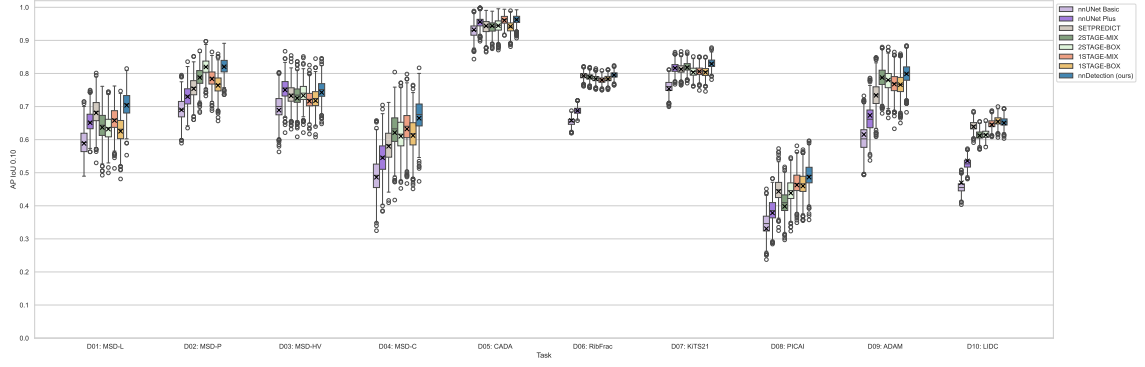


Figure B.1: Cross-validation performance on development pool. Shows the distribution of the detection performance with the mAP metric at an IoU threshold of 0.1. Bootstrapping with 1000 iterations was performed to generate the distribution. This figure is adapted from [82].

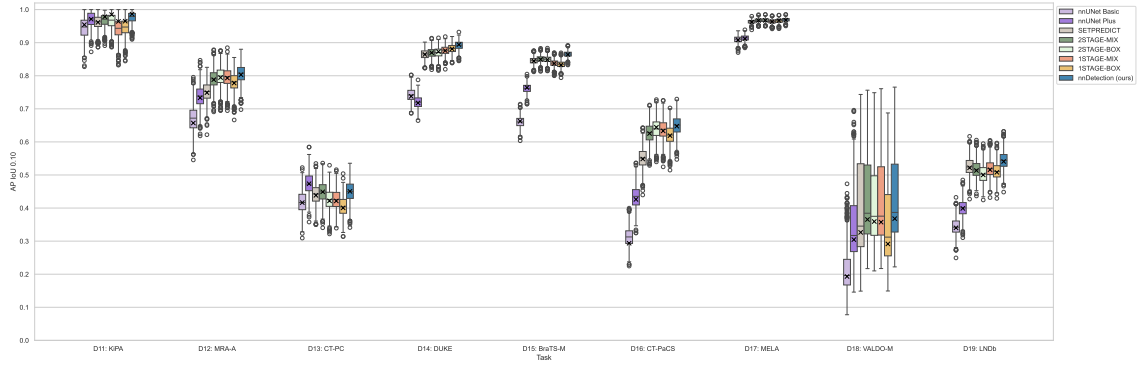


Figure B.2: Cross-validation performance on generalisation pool. Shows the distribution of the detection performance with the mAP metric at an IoU threshold of 0.1. Bootstrapping with 1000 iterations was performed to generate the distribution. This figure is adapted from [82].

B.3 Self-Configuring Design of Medical Object Detection Methods

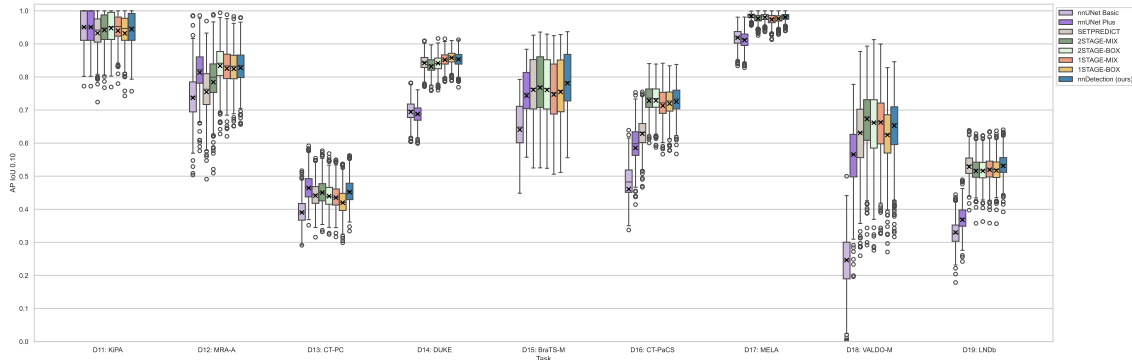


Figure B.3: Test set performance on generalisation pool. Shows the distribution of the detection performance with the mAP metric at an IoU threshold of 0.1. Bootstrapping with 1000 iterations was performed to generate the distribution. This figure is adapted from [82].

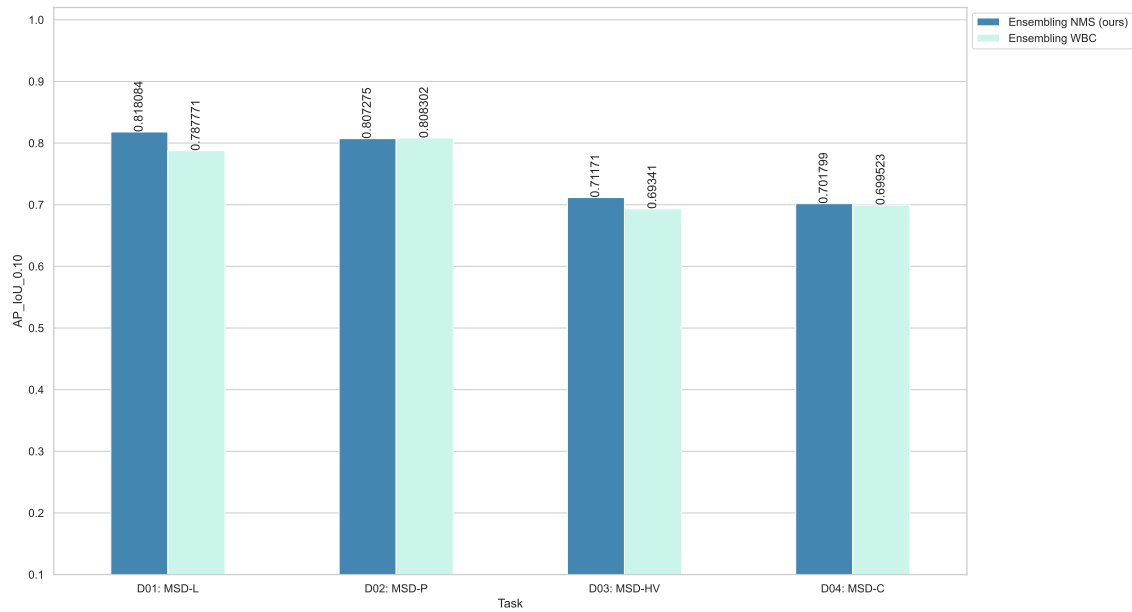


Figure B.4: Test set performance on development pool. Performance comparison between NMS and WBC ensembling of models on four test sets of the development pool. Detection performance is measured via the mAP metric at an IoU threshold of 0.1. This figure is adapted from [82].

B.3.2 Data Set Results

Table B.13: Cross-Validation detection performance on the MSD-L data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.589	0.491	0.330	0.426	0.324	0.233
nnUNet Plus	0.651	0.572	0.400	0.541	0.483	0.347
Anchor Single Stage						
1STAGE-MIX	0.658	0.575	0.324	0.495	0.443	0.270
1STAGE-BOX	0.625	0.545	0.329	0.470	0.420	0.271
Anchor Two Stage						
2STAGE-MIX	0.637	0.558	0.332	0.452	0.406	0.248
2STAGE-BOX	0.632	0.550	0.322	0.480	0.429	0.244
Set Prediction						
SETPREDICT	<u>0.681</u>	<u>0.587</u>	0.327	0.515	0.453	0.271
Ensemble						
nnDetection (ours)	0.704	0.604	<u>0.342</u>	<u>0.539</u>	<u>0.473</u>	<u>0.289</u>

Table B.14: Test set detection performance on the MSD-L data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Anchor Single Stage						
1STAGE-MIX	0.787	0.700	0.475	0.688	0.620	0.447
1STAGE-BOX	0.770	0.691	0.453	0.667	0.609	0.423
Anchor Two Stage						
2STAGE-MIX	0.776	0.701	0.532	0.675	0.616	0.481
2STAGE-BOX	0.779	0.708	0.484	0.682	0.632	0.454
Set Prediction						
SETPREDICT	<u>0.795</u>	<u>0.721</u>	<u>0.517</u>	<u>0.714</u>	<u>0.657</u>	<u>0.477</u>
Ensemble						
nnDetection (ours)	0.818	0.723	0.492	0.733	0.661	0.461

Table B.15: Cross-Validation detection performance on the MSD-P data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.690	0.523	0.286	0.721	0.593	0.361
nnUNet Plus	0.730	0.574	0.359	0.768	0.626	0.428
Anchor Single Stage						
1STAGE-MIX	0.784	<u>0.622</u>	0.341	0.816	0.665	0.426
1STAGE-BOX	0.764	0.614	<u>0.350</u>	0.808	0.654	<u>0.427</u>
Anchor Two Stage						
2STAGE-MIX	0.788	0.610	0.333	0.823	<u>0.677</u>	0.398
2STAGE-BOX	<u>0.820</u>	0.610	0.304	<u>0.845</u>	0.672	0.367
Set Prediction						
SETPREDICT	0.754	0.556	0.231	0.807	0.628	0.349
Ensemble						
nnDetection (ours)	0.820	0.643	0.331	0.859	0.712	0.396

Table B.16: Test set detection performance on the MSD-P data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Anchor Single Stage						
1STAGE-MIX	0.785	<u>0.610</u>	0.392	0.832	0.687	0.484
1STAGE-BOX	0.763	0.579	0.322	0.822	0.662	0.422
Anchor Two Stage						
2STAGE-MIX	0.788	0.602	0.353	0.834	0.671	0.445
2STAGE-BOX	0.794	0.580	0.278	0.839	0.655	0.373
Set Prediction						
SETPREDICT	<u>0.807</u>	0.588	0.309	<u>0.849</u>	0.662	0.415
Ensemble						
nnDetection (ours)	0.807	0.614	<u>0.356</u>	0.859	<u>0.686</u>	<u>0.455</u>

Table B.17: Cross-Validation detection performance on the MSD-HV data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.689	0.615	<u>0.466</u>	0.729	0.657	<u>0.483</u>
nnUNet Plus	0.750	0.705	0.592	<u>0.758</u>	0.716	0.609
Anchor Single Stage						
1STAGE-MIX	0.716	0.655	0.428	0.725	0.670	0.479
1STAGE-BOX	0.718	0.669	0.405	0.732	0.688	0.450
Anchor Two Stage						
2STAGE-MIX	0.725	0.643	0.360	0.742	0.676	0.405
2STAGE-BOX	0.733	0.633	0.357	0.749	0.661	0.400
Set Prediction						
SETPREDICT	0.732	<u>0.671</u>	0.424	0.739	0.682	0.472
Ensemble						
nnDetection (ours)	<u>0.743</u>	0.660	0.381	0.760	<u>0.693</u>	0.423

Table B.18: Test set detection performance on the MSD-HV data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Anchor Single Stage						
1STAGE-MIX	0.697	0.639	<u>0.404</u>	0.703	0.656	0.443
1STAGE-BOX	0.692	0.607	0.389	0.703	0.627	0.423
Anchor Two Stage						
2STAGE-MIX	0.686	0.600	0.346	<u>0.712</u>	0.640	0.408
2STAGE-BOX	<u>0.697</u>	0.573	0.317	0.709	0.618	0.392
Set Prediction						
SETPREDICT	0.683	0.604	0.405	0.700	0.636	0.451
Ensemble						
nnDetection (ours)	0.712	<u>0.610</u>	0.383	0.730	<u>0.653</u>	<u>0.445</u>

Table B.19: Cross-Validation detection performance on the MSD-C data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.486	0.475	0.371	0.532	0.516	0.416
nnUNet Plus	0.545	0.521	0.433	0.579	0.546	0.474
Anchor Single Stage						
1STAGE-MIX	<u>0.632</u>	<u>0.583</u>	0.488	<u>0.683</u>	0.625	0.535
1STAGE-BOX	0.612	0.559	0.423	0.659	0.598	0.462
Anchor Two Stage						
2STAGE-MIX	0.621	0.575	0.365	0.672	0.625	0.447
2STAGE-BOX	0.610	0.581	<u>0.467</u>	0.667	<u>0.631</u>	0.509
Set Prediction						
SETPREDICT	0.580	0.543	0.464	0.655	0.614	<u>0.518</u>
Ensemble						
nnDetection (ours)	0.665	0.628	0.412	0.722	0.684	0.504

Table B.20: Test set detection performance on the MSD-C data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Anchor Single Stage						
1STAGE-MIX	<u>0.719</u>	<u>0.577</u>	<u>0.445</u>	0.744	0.605	<u>0.492</u>
1STAGE-BOX	0.722	0.581	0.431	0.756	0.632	0.474
Anchor Two Stage						
2STAGE-MIX	0.669	0.535	0.349	0.733	0.605	0.436
2STAGE-BOX	0.712	0.561	0.415	<u>0.759</u>	<u>0.624</u>	0.455
Set Prediction						
SETPREDICT	0.695	0.536	0.465	0.741	0.617	0.530
Ensemble						
nnDetection (ours)	0.702	0.562	0.373	0.763	0.624	0.459

Table B.21: Cross-Validation detection performance on the CADA data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.932	0.922	0.887	0.945	0.937	0.906
nnUNet Plus	0.956	<u>0.956</u>	0.903	0.961	0.961	<u>0.921</u>
Anchor Single Stage						
1STAGE-MIX	<u>0.961</u>	0.960	0.893	<u>0.970</u>	<u>0.969</u>	0.901
1STAGE-BOX	0.942	0.925	0.892	0.948	0.935	0.900
Anchor Two Stage						
2STAGE-MIX	0.944	0.943	0.914	0.952	0.952	0.922
2STAGE-BOX	0.945	0.936	0.902	0.958	0.953	<u>0.921</u>
Set Prediction						
SETPREDICT	0.944	0.925	0.894	0.953	0.937	0.907
Ensemble						
nnDetection (ours)	0.963	0.956	<u>0.908</u>	0.975	0.970	0.919

Table B.22: Cross-Validation detection performance on the RibFrac data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.658	0.560	0.402	0.533	0.344	0.218
nnUNet Plus	0.688	0.597	0.453	0.591	0.504	0.358
Anchor Single Stage						
1STAGE-MIX	0.780	0.662	0.447	0.646	0.532	0.349
1STAGE-BOX	0.784	0.660	0.458	0.652	0.530	0.358
Anchor Two Stage						
2STAGE-MIX	0.789	0.677	0.490	0.668	0.563	0.392
2STAGE-BOX	0.784	<u>0.675</u>	0.483	0.655	<u>0.554</u>	<u>0.385</u>
Set Prediction						
SETPREDICT	<u>0.793</u>	0.664	0.483	0.666	0.544	0.382
Ensemble						
nnDetection (ours)	0.795	0.668	<u>0.484</u>	<u>0.666</u>	0.545	0.383

Table B.23: Cross-Validation detection performance on the KiTS21 data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.754	0.721	<u>0.644</u>	0.666	0.639	0.571
nnUNet Plus	0.816	0.790	0.699	0.792	0.768	0.681
Anchor Single Stage						
1STAGE-MIX	0.806	0.773	0.631	0.749	0.722	0.595
1STAGE-BOX	0.804	0.772	0.627	0.749	0.721	0.589
Anchor Two Stage						
2STAGE-MIX	<u>0.818</u>	0.782	0.631	0.758	0.731	0.602
2STAGE-BOX	0.803	0.763	0.626	0.758	0.724	0.594
Set Prediction						
SETPREDICT	0.813	0.779	0.628	0.758	0.733	0.594
Ensemble						
nnDetection (ours)	0.829	<u>0.784</u>	0.640	<u>0.781</u>	<u>0.745</u>	<u>0.615</u>

Table B.24: Cross-Validation detection performance on the PICAI data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.331	0.284	0.132	0.751	0.675	0.453
nnUNet Plus	0.378	0.338	0.185	0.769	0.706	0.487
Anchor Single Stage						
1STAGE-MIX	<u>0.463</u>	<u>0.397</u>	<u>0.188</u>	0.789	0.693	0.423
1STAGE-BOX	0.460	0.389	0.193	0.780	0.671	0.409
Anchor Two Stage						
2STAGE-MIX	0.398	0.342	0.170	0.692	0.618	0.382
2STAGE-BOX	0.438	0.345	0.156	0.760	0.617	0.377
Set Prediction						
SETPREDICT	0.443	0.388	0.180	<u>0.803</u>	0.724	<u>0.460</u>
Ensemble						
nnDetection (ours)	0.487	0.414	0.186	0.827	<u>0.721</u>	0.424

Table B.25: Cross-Validation detection performance on the ADAM data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.616	0.545	0.356	0.631	0.566	0.363
nnUNet Plus	0.673	0.626	0.407	0.691	0.645	0.481
Anchor Single Stage						
1STAGE-MIX	0.768	0.705	0.392	0.799	0.738	0.463
1STAGE-BOX	0.766	0.697	0.422	0.793	0.721	<u>0.488</u>
Anchor Two Stage						
2STAGE-MIX	<u>0.788</u>	<u>0.725</u>	0.396	<u>0.813</u>	0.745	0.474
2STAGE-BOX	0.780	0.724	<u>0.418</u>	0.808	<u>0.747</u>	0.509
Set Prediction						
SETPREDICT	0.734	0.662	0.387	0.762	0.696	0.465
Ensemble						
nnDetection (ours)	0.799	0.737	0.388	0.825	0.765	0.477

Table B.26: Cross-Validation detection performance on the LIDC data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.470	0.462	0.417	0.432	0.424	0.386
nnUNet Plus	0.536	0.524	0.454	0.468	0.457	0.406
Anchor Single Stage						
1STAGE-MIX	0.645	0.640	0.568	0.621	0.616	0.557
1STAGE-BOX	0.654	0.647	0.580	<u>0.632</u>	<u>0.627</u>	0.574
Anchor Two Stage						
2STAGE-MIX	0.612	0.597	0.525	0.594	0.584	0.524
2STAGE-BOX	0.614	0.601	0.525	0.601	0.592	0.530
Set Prediction						
SETPREDICT	0.639	0.631	0.564	0.630	0.623	0.566
Ensemble						
nnDetection (ours)	<u>0.650</u>	<u>0.644</u>	<u>0.573</u>	0.636	0.630	<u>0.573</u>

Table B.27: Cross-Validation detection performance on the KiPA data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.954	0.954	<u>0.954</u>	0.980	0.980	0.980
nnUNet Plus	0.971	0.971	0.971	0.980	0.980	0.980
Anchor Single Stage						
1STAGE-MIX	0.964	0.946	0.939	0.983	0.963	0.954
1STAGE-BOX	0.965	0.960	0.946	0.974	0.966	0.954
Anchor Two Stage						
2STAGE-MIX	0.979	0.979	0.944	0.989	0.989	0.949
2STAGE-BOX	<u>0.985</u>	<u>0.984</u>	0.952	0.994	0.994	0.960
Set Prediction						
SETPREDICT	0.961	0.961	0.906	0.980	0.980	0.920
Ensemble						
nnDetection (ours)	0.986	0.986	0.925	0.994	0.994	0.937

Table B.28: Test set detection performance on the KiPA data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.950	0.901	0.901	0.955	0.909	0.909
nnUNet Plus	0.950	0.950	0.950	0.955	0.955	0.955
Anchor Single Stage						
1STAGE-MIX	0.940	0.939	<u>0.906</u>	0.948	0.942	<u>0.916</u>
1STAGE-BOX	0.932	0.924	0.893	0.942	0.935	0.909
Anchor Two Stage						
2STAGE-MIX	0.942	0.942	0.896	0.948	0.948	0.903
2STAGE-BOX	0.948	<u>0.947</u>	0.900	0.955	0.955	0.909
Set Prediction						
SETPREDICT	0.933	0.924	0.870	0.948	0.942	0.883
Ensemble						
nnDetection (ours)	0.944	0.944	0.901	0.948	0.948	0.909

Table B.29: Cross-Validation detection performance on the MRA-A data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-BOX, 2STAGE-BOX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.657	0.507	0.311	0.728	0.590	0.432
nnUNet Plus	0.733	0.626	0.351	0.777	0.690	0.469
Anchor Single Stage						
1STAGE-MIX	0.793	0.660	0.374	0.840	0.719	0.462
1STAGE-BOX	0.778	0.655	0.336	0.834	0.712	0.430
Anchor Two Stage						
2STAGE-MIX	0.788	0.670	0.366	0.827	0.720	0.485
2STAGE-BOX	<u>0.795</u>	<u>0.679</u>	0.372	<u>0.844</u>	<u>0.725</u>	0.478
Set Prediction						
SETPREDICT	0.749	0.602	0.301	0.797	0.672	0.412
Ensemble						
nnDetection (ours)	0.803	0.690	<u>0.372</u>	0.852	0.739	<u>0.480</u>

Table B.30: Test set detection performance on the MRA-A data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-BOX, 2STAGE-BOX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.737	0.594	0.342	0.758	0.606	0.394
nnUNet Plus	0.814	0.572	0.481	0.883	0.658	0.563
Anchor Single Stage						
1STAGE-MIX	0.825	0.734	<u>0.413</u>	0.883	<u>0.797</u>	<u>0.515</u>
1STAGE-BOX	0.825	0.702	0.345	0.900	0.788	0.481
Anchor Two Stage						
2STAGE-MIX	0.784	0.673	0.294	0.887	0.810	0.476
2STAGE-BOX	0.834	<u>0.722</u>	0.379	<u>0.896</u>	0.792	0.481
Set Prediction						
SETPREDICT	0.755	0.676	0.258	0.848	0.775	0.420
Ensemble						
nnDetection (ours)	<u>0.828</u>	0.711	0.366	<u>0.896</u>	0.788	0.455

Table B.31: Cross-Validation detection performance on the CT-PC data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.417	0.342	0.225	0.445	0.339	0.255
nnUNet Plus	0.474	<u>0.432</u>	<u>0.321</u>	0.485	0.453	0.347
Anchor Single Stage						
1STAGE-MIX	0.422	0.399	0.284	0.429	0.412	0.314
1STAGE-BOX	0.401	0.369	0.263	0.407	0.384	0.299
Anchor Two Stage						
2STAGE-MIX	0.449	0.431	0.318	0.452	0.442	<u>0.348</u>
2STAGE-BOX	0.422	0.392	0.274	0.433	0.411	0.307
Set Prediction						
SETPREDICT	0.438	0.410	0.309	0.441	0.418	0.340
Ensemble						
nnDetection (ours)	<u>0.450</u>	0.433	0.331	<u>0.455</u>	<u>0.444</u>	0.358

Table B.32: Test set detection performance on the CT-PC data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.390	0.348	0.248	0.398	0.356	0.253
nnUNet Plus	0.464	<u>0.433</u>	0.347	0.476	0.448	<u>0.360</u>
Anchor Single Stage						
1STAGE-MIX	0.435	0.410	0.315	0.428	0.407	0.328
1STAGE-BOX	0.420	0.397	0.326	0.409	0.394	0.335
Anchor Two Stage						
2STAGE-MIX	0.450	0.426	0.336	<u>0.440</u>	0.422	0.340
2STAGE-BOX	0.440	0.424	0.327	0.429	0.418	0.334
Set Prediction						
SETPREDICT	0.442	0.431	0.362	0.427	0.423	0.368
Ensemble						
nnDetection (ours)	<u>0.452</u>	0.438	<u>0.350</u>	0.439	<u>0.432</u>	0.359

Table B.33: Cross-Validation detection performance on the DUKE data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-BOX, 2STAGE-BOX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.738	0.520	0.235	0.809	0.611	0.329
nnUNet Plus	0.718	0.558	0.299	0.778	0.630	0.381
Anchor Single Stage						
1STAGE-MIX	0.876	0.761	<u>0.459</u>	<u>0.919</u>	0.817	<u>0.556</u>
1STAGE-BOX	<u>0.881</u>	0.774	0.477	0.918	<u>0.817</u>	0.564
Anchor Two Stage						
2STAGE-MIX	0.870	0.752	0.421	0.909	0.817	0.493
2STAGE-BOX	0.873	0.751	0.396	0.908	0.812	0.481
Set Prediction						
SETPREDICT	0.864	0.759	0.456	0.901	0.814	0.554
Ensemble						
nnDetection (ours)	0.894	<u>0.764</u>	0.406	0.931	0.827	0.494

Table B.34: Test set detection performance on the DUKE data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-BOX, 2STAGE-BOX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.695	0.471	0.223	0.770	0.559	0.318
nnUNet Plus	0.688	0.494	0.291	0.757	0.583	0.377
Anchor Single Stage						
1STAGE-MIX	0.852	0.707	0.458	0.891	0.772	0.548
1STAGE-BOX	0.858	0.735	<u>0.475</u>	<u>0.890</u>	0.793	<u>0.565</u>
Anchor Two Stage						
2STAGE-MIX	0.832	0.698	0.420	0.875	0.778	0.542
2STAGE-BOX	0.842	<u>0.733</u>	0.455	0.876	<u>0.787</u>	0.564
Set Prediction						
SETPREDICT	0.842	0.715	0.487	0.889	0.771	0.572
Ensemble						
nnDetection (ours)	<u>0.853</u>	0.730	0.451	<u>0.890</u>	0.777	0.551

Table B.35: Cross-Validation detection performance on the BraTS-M data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.662	0.610	0.496	0.494	0.461	0.327
nnUNet Plus	0.764	0.717	0.573	0.680	0.629	0.481
Anchor Single Stage						
1STAGE-MIX	0.837	0.807	0.615	0.731	0.701	0.515
1STAGE-BOX	0.833	0.795	0.608	0.731	0.696	0.481
Anchor Two Stage						
2STAGE-MIX	<u>0.849</u>	0.817	0.640	<u>0.747</u>	<u>0.718</u>	0.454
2STAGE-BOX	0.849	<u>0.820</u>	<u>0.636</u>	0.739	0.711	0.449
Set Prediction						
SETPREDICT	0.845	0.812	0.619	0.732	0.699	<u>0.512</u>
Ensemble						
nnDetection (ours)	0.864	0.825	0.635	0.758	0.725	0.455

Table B.36: Test set detection performance on the BraTS-M data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.641	0.598	0.497	0.554	0.454	0.325
nnUNet Plus	0.743	0.699	0.579	0.667	0.603	0.484
Anchor Single Stage						
1STAGE-MIX	0.747	0.732	0.612	0.678	0.662	0.546
1STAGE-BOX	0.755	0.741	0.616	0.675	0.664	0.537
Anchor Two Stage						
2STAGE-MIX	<u>0.768</u>	<u>0.754</u>	<u>0.636</u>	<u>0.696</u>	<u>0.684</u>	<u>0.568</u>
2STAGE-BOX	0.761	0.748	0.642	0.688	0.676	0.572
Set Prediction						
SETPREDICT	0.761	0.746	0.616	0.682	0.668	0.541
Ensemble						
nnDetection (ours)	0.781	0.761	0.632	0.702	0.685	0.568

Table B.37: Cross-Validation detection performance on the CT-PaCS data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.294	0.232	0.151	0.595	0.517	0.405
nnUNet Plus	0.426	0.362	0.268	0.687	0.598	0.475
Anchor Single Stage						
1STAGE-MIX	0.633	0.570	0.402	0.815	0.710	0.504
1STAGE-BOX	0.619	0.552	0.407	0.801	0.684	0.511
Anchor Two Stage						
2STAGE-MIX	0.625	0.562	0.404	0.758	0.681	0.510
2STAGE-BOX	<u>0.644</u>	<u>0.572</u>	0.422	0.791	0.685	<u>0.518</u>
Set Prediction						
SETPREDICT	0.548	0.475	0.365	0.754	0.654	0.506
Ensemble						
nnDetection (ours)	0.648	0.577	<u>0.414</u>	<u>0.804</u>	<u>0.702</u>	0.529

Table B.38: Test set detection performance on the CT-PaCS data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.461	0.322	0.194	0.772	0.637	0.421
nnUNet Plus	0.586	0.418	0.269	0.776	0.650	0.501
Anchor Single Stage						
1STAGE-MIX	0.713	0.608	0.393	0.898	0.766	0.543
1STAGE-BOX	0.720	0.590	0.406	<u>0.895</u>	0.741	0.524
Anchor Two Stage						
2STAGE-MIX	<u>0.729</u>	0.636	0.466	0.865	0.753	0.579
2STAGE-BOX	0.729	<u>0.635</u>	0.451	0.870	<u>0.763</u>	0.569
Set Prediction						
SETPREDICT	0.629	0.513	0.390	0.816	0.706	<u>0.570</u>
Ensemble						
nnDetection (ours)	0.726	0.630	<u>0.459</u>	0.880	0.761	0.566

Table B.39: Cross-Validation detection performance on the MELA data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-BOX, 2STAGE-BOX. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.909	0.895	0.849	0.945	0.928	0.891
nnUNet Plus	0.912	0.901	0.865	0.944	0.930	0.897
Anchor Single Stage						
1STAGE-MIX	0.964	0.959	0.877	0.982	0.976	0.925
1STAGE-BOX	0.967	0.957	0.897	0.981	0.973	<u>0.930</u>
Anchor Two Stage						
2STAGE-MIX	0.967	<u>0.958</u>	0.886	0.983	0.975	0.924
2STAGE-BOX	<u>0.967</u>	0.956	0.891	<u>0.984</u>	0.974	0.926
Set Prediction						
SETPREDICT	0.962	0.955	0.883	0.975	0.967	0.906
Ensemble						
nnDetection (ours)	0.968	0.957	<u>0.893</u>	0.984	<u>0.975</u>	0.930

Table B.40: Test set detection performance on the MELA data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-BOX, 2STAGE-BOX. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.919	0.909	0.880	0.938	0.926	0.906
nnUNet Plus	0.911	0.903	0.882	0.938	0.930	0.909
Anchor Single Stage						
1STAGE-MIX	0.974	0.967	0.863	<u>0.987</u>	0.981	0.917
1STAGE-BOX	0.976	0.973	0.886	0.983	<u>0.979</u>	<u>0.918</u>
Anchor Two Stage						
2STAGE-MIX	0.977	<u>0.971</u>	<u>0.899</u>	0.981	0.975	0.914
2STAGE-BOX	0.980	0.955	0.890	0.984	0.964	0.910
Set Prediction						
SETPREDICT	0.984	0.966	0.899	0.990	0.973	0.927
Ensemble						
nnDetection (ours)	<u>0.981</u>	0.957	0.890	0.986	0.965	0.912

Table B.41: Cross-Validation detection performance on the VALDO-M data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.193	0.186	<u>0.151</u>	0.214	0.200	<u>0.160</u>
nnUNet Plus	0.305	0.292	0.248	0.331	0.318	0.261
Anchor Single Stage						
1STAGE-MIX	0.357	0.312	0.063	0.366	<u>0.337</u>	0.107
1STAGE-BOX	0.292	0.239	0.041	0.317	0.274	0.093
Anchor Two Stage						
2STAGE-MIX	<u>0.365</u>	<u>0.323</u>	0.075	<u>0.373</u>	<u>0.337</u>	0.121
2STAGE-BOX	0.359	0.317	0.050	0.370	0.334	0.096
Set Prediction						
SETPREDICT	0.326	0.305	0.046	0.335	0.314	0.096
Ensemble						
nnDetection (ours)	0.368	0.325	0.072	0.374	0.339	0.118

Table B.42: Test set detection performance on the VALDO-M data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, 2STAGE-MIX. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.246	0.238	0.166	0.247	0.240	0.172
nnUNet Plus	0.566	0.512	0.420	0.492	0.448	<u>0.368</u>
Anchor Single Stage						
1STAGE-MIX	<u>0.663</u>	0.603	0.397	0.521	0.490	0.341
1STAGE-BOX	0.625	0.547	0.254	0.518	0.440	0.251
Anchor Two Stage						
2STAGE-MIX	0.673	0.617	0.460	0.547	0.505	0.375
2STAGE-BOX	0.662	0.562	0.306	<u>0.538</u>	0.452	0.294
Set Prediction						
SETPREDICT	0.631	<u>0.611</u>	0.262	0.504	<u>0.500</u>	0.257
Ensemble						
nnDetection (ours)	0.653	0.595	<u>0.426</u>	0.526	0.489	0.361

Table B.43: Cross-Validation detection performance on the LNDb data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.340	0.316	0.243	0.360	0.314	0.255
nnUNet Plus	0.399	0.359	0.299	0.393	0.364	0.315
Anchor Single Stage						
1STAGE-MIX	0.516	0.447	0.316	0.473	0.420	0.333
1STAGE-BOX	0.508	0.446	0.321	0.468	0.418	0.337
Anchor Two Stage						
2STAGE-MIX	0.514	0.448	<u>0.341</u>	0.473	0.427	<u>0.351</u>
2STAGE-BOX	0.500	0.444	0.328	0.463	0.424	0.342
Set Prediction						
SETPREDICT	<u>0.522</u>	<u>0.470</u>	0.346	<u>0.483</u>	<u>0.448</u>	0.354
Ensemble						
nnDetection (ours)	0.541	0.479	0.336	0.494	0.453	0.347

Table B.44: Test set detection performance on the LNDb data set. The best model is highlighted in bold, the second best model is underlined. nnDetection ensemble: 1STAGE-MIX, SETPREDICT. Table reproduced from [82].

Model	AP			FROC		
	IoU 0.10	IoU 0.30	IoU 0.50	IoU 0.10	IoU 0.30	IoU 0.50
Segmentation						
nnUNet Basic	0.330	0.313	0.273	0.334	0.318	0.279
nnUNet Plus	0.368	0.355	0.287	0.365	0.355	0.300
Anchor Single Stage						
1STAGE-MIX	0.520	0.472	0.341	<u>0.467</u>	0.437	0.337
1STAGE-BOX	0.517	0.470	0.349	0.464	0.436	0.341
Anchor Two Stage						
2STAGE-MIX	0.516	0.475	0.376	0.462	<u>0.438</u>	0.364
2STAGE-BOX	0.516	0.478	<u>0.371</u>	0.451	0.430	0.353
Set Prediction						
SETPREDICT	<u>0.529</u>	0.492	0.370	0.463	<u>0.438</u>	<u>0.358</u>
Ensemble						
nnDetection (ours)	0.531	<u>0.485</u>	0.355	0.473	0.447	0.353

Table B.45: Overview of LUNA16 results. Shows the FROC score and individual sensitivities at the working points. 'SM' indicates the reference to the method, and 'SR' indicates the reference to LUNA16 results. Table reproduced from [82].

Method	SM	SR	Split	1/8	1/4	1/2	1	2	4	8	Score
Liao et al. (2019)	[78]	[70]		0.594	0.727	0.781	0.844	0.875	0.891	0.898	0.801
Harsono et al. (2022)	[79]	[70]		0.636	0.713	0.798	0.853	0.876	0.899	0.915	0.813
Dou et al. (2017)	[65]	[247]		0.659	0.745	0.819	0.865	0.906	0.933	0.946	0.839
Tang et al. (2019)	[77]	[70]		0.652	0.768	0.839	0.875	0.911	0.929	0.938	0.844
Li et al. (2020)	[72]	[70]		0.739	0.803	0.858	0.888	0.907	0.916	0.920	0.862
Lu et al. (2023)	[68]	[68]		0.712	0.792	0.858	0.900	0.928	0.949	0.961	0.871
Mei et al. (2022)	[70]	[70]		0.712	0.802	0.865	0.901	0.937	0.946	0.955	0.874
Wang et al. (2018)	[67]	[247]		0.676	0.776	0.879	0.949	0.958	0.958	0.958	0.878
Song et al. (2020)	[74]	[74]		0.723	0.838	0.887	0.911	0.928	0.934	0.948	0.881
Gong et al. (2020)	[174]	[174]		0.713	0.801	0.867	0.917	0.950	0.962	0.971	0.883
Ding et al. (2017)	[66]	[247]		0.748	0.853	0.887	0.922	0.938	0.944	0.946	0.891
Luo et al. (2022)	[69]	[69]		0.743	0.829	0.889	0.922	0.939	0.958	0.964	0.892
Khosravan et al. (2018)	[76]	[247]		0.709	0.836	0.921	0.953	0.953	0.953	0.953	0.897
Gong et al. (2020)	[174]	[174]		0.784	0.847	0.906	0.938	0.950	0.955	0.961	0.906
Cao et al. (2020)	[64]	[64]		0.848	0.899	0.925	0.936	0.949	0.957	0.960	0.925
SETPREDICT			811	0.810	0.874	0.908	0.930	0.950	0.969	0.979	0.917
nnDetection (ours)			811	0.790	0.879	0.916	0.948	0.958	0.975	0.980	0.921
2STAGE-BOX			811	0.811	0.877	0.917	0.949	0.963	0.970	0.977	0.923
1STAGE-BOX			811	0.821	0.873	0.923	0.947	0.962	0.971	0.980	0.925
Zhu et al. (2018)	[75]	[247]	901	0.692	0.769	0.824	0.865	0.893	0.917	0.933	0.842
Liu et al. (2019)	[247]	[247]	901	0.848	0.876	0.905	0.933	0.943	0.957	0.970	0.919
1STAGE-BOX			901	0.824	0.889	0.923	0.945	0.966	0.978	0.985	0.930
2STAGE-BOX			901	0.828	0.895	0.929	0.952	0.965	0.978	0.987	0.934
nnDetection (ours)			901	0.831	0.901	0.933	0.954	0.972	0.980	0.987	0.937
SETPREDICT			901	0.844	0.898	0.933	0.951	0.967	0.979	0.986	0.937
Liu et al. (2019)	[247]	[247]	901	0.904	0.914	0.933	0.957	0.971	0.971	0.971	0.952

Table B.46: Overview of PN9 results. Shows the FROC score and individual sensitivities at the working points. 'SM' indicates the reference to the method, and 'SR' indicates the reference to PN9 results. Table reproduced from [82].

Method	SM	SR	0.125	0.25	0.5	1.0	2.0	4.0	8.0	Score
RetinaNet (2D)	[35]	[70]	0.084	0.130	0.201	0.291	0.404	0.525	0.654	0.327
Faster R-CNN (2D)	[36]	[70]	0.108	0.158	0.232	0.329	0.466	0.619	0.755	0.381
SSD512 (2D)	[118]	[70]	0.123	0.188	0.280	0.403	0.569	0.732	0.865	0.451
Leaky Noisy-OR	[78]	[70]	0.281	0.364	0.470	0.567	0.661	0.738	0.817	0.557
3D Faster R-CNN	[36]	[70]	0.276	0.366	0.468	0.580	0.700	0.800	0.883	0.582
I3DR-Net	[79]	[70]	0.240	0.344	0.468	0.600	0.729	0.836	0.896	0.588
NoduleNet	[77]	[70]	0.273	0.383	0.494	0.611	0.731	0.833	0.898	0.603
DeepLung	[75]	[70]	0.286	0.391	0.502	0.623	0.726	0.820	0.886	0.605
DeepSEED	[72]	[70]	0.292	0.406	0.511	0.622	0.738	0.832	0.897	0.614
SANet	[70]	[70]	0.381	0.450	0.545	0.645	0.753	0.839	0.900	0.645
LSSANet	[71]	[71]	0.516	0.516	0.582	0.669	0.773	0.853	0.899	0.687
1STAGE-BOX			0.394	0.502	0.615	0.728	0.827	0.897	0.942	0.701
2STAGE-BOX			0.394	0.510	0.626	0.726	0.826	0.892	0.935	0.701
SETPREDICT (ours)			0.417	0.511	0.629	0.730	0.825	0.896	0.943	0.707
SETPREDICT			0.425	0.521	0.635	0.736	0.829	0.899	0.943	0.713
nnDetection (ours)			0.422	0.523	0.631	0.740	0.835	0.905	0.945	0.714

Table B.47: Overview of CTA-A internal test set results. Shows the FROC score and individual sensitivities at the working points. 'SM' indicates the reference to the method, and 'SR' indicates the reference to CTA-A results. Table reproduced from [82].

Method	SM	SR	0.125	0.25	0.5	1.0	2.0	4.0	8.0	Score
Xie et al. (2023)	[219]	[83]	0.706	0.754	0.817	0.849	0.913	0.929	0.929	0.842
Song et al. (2020)	[74]	[83]	0.723	0.812	0.901	0.911	0.941	0.941	0.941	0.881
Ceballos-Arroyo et al. (2024)	[83]	[83]	0.841	0.897	0.897	0.913	0.929	0.929	0.929	0.905
SETPREDICT			0.857	0.897	0.937	0.937	0.952	0.960	0.960	0.929
nnDetection (ours)			0.857	0.921	0.937	0.937	0.952	0.952	0.952	0.930
1STAGE-MIX			0.857	0.921	0.937	0.944	0.952	0.960	0.968	0.934
2STAGE-MIX			0.905	0.929	0.952	0.952	0.968	0.968	0.976	0.950

Table B.48: Overview of CTA-A external test set results. Shows the FROC score and individual sensitivities at the working points. 'SM' indicates the reference to the method, and 'SR' indicates the reference to CTA-A results. Table reproduced from [82].

Method	SM	SR	0.125	0.25	0.5	1.0	2.0	4.0	8.0	Score
Xie et al. (2023)	[219]	[83]	0.515	0.733	0.812	0.901	0.970	0.970	0.970	0.839
Song et al. (2020)	[74]	[83]	0.723	0.812	0.901	0.911	0.941	0.941	0.941	0.881
1STAGE-MIX			0.861	0.950	0.960	0.970	0.980	0.990	0.990	0.957
Ceballos-Arroyo et al. (2024)	[83]	[83]	0.861	0.960	0.970	0.970	0.980	0.990	0.990	0.960
SETPREDICT			0.941	0.960	0.960	0.960	0.960	0.980	0.980	0.963
2STAGE-MIX			0.921	0.931	0.970	0.980	0.980	0.980	0.980	0.963
nnDetection (ours)			0.921	0.970	0.980	0.980	0.980	0.980	0.980	0.970

B.3.3 Data Set Information

Label Preparation

D05 [17], D06 [11, 12], D12 [18], D19 [207] provided instance segmentations and did not undergo additional preparation steps.

D14 [25, 26], D17 [19], D20 [43], D21 [14] provided bounding box annotations and did not undergo additional filtering steps. The boxes of D14 were converted to bounding box like segmentations. The annotations of D17, D20 and D21 were converted to ellipsoid like segmentations.

The semantic segmentation labels of D01 [16], D11 [199, 200, 201, 202] were processed via connected component analysis to derive object level annotations. No further modifications were applied to the labels.

The semantic segmentation labels of D02 [16], D03 [16], D04 [16] were processed via connected component analysis to derive object level annotations. The smallest and largest instances were manually checked and obvious errors were corrected.

D07 [15] provided segmentations from multiple raters for each instance. The majority vote of the raters was used as ground truth object delineations. If voxels of neighboring objects overlapped, the voxels were assigned to the closest object center.

Semantic segmentation labels of D08 [197, 198] were converted to instance segmentations by connected component analysis via the official script.

D09 [44] underwent connected component analysis to convert semantic segmentations into instance segmentations. Only untreated intracranial aneurysms were considered as targets.

D10 [13] provides instance annotations from multiple readers. Only nodules annotated by at least two readers were considered as correct. Nodules with a malignancy score greater than three were considered malignant, the rest was considered benign. The detection network was tasked to differentiate benign and malignant nodules. Instances which were not automatically resolvable by pyLIDC [248] were manually resolved.

D13 [203] was processed via connected component analysis to derive object level annotations. Instances with a volume below 10mm^3 were discarded as outlined in the original publication [203].

D15 [204, 205] underwent connected component analysis to convert semantic segmentations into instance segmentations. The conversion protocol was adapted from the object-level evaluation of the original manuscript [204, 205].

Semantic segmentation labels from D16 [206] were processed via connected component analysis to derive object level annotations. Patients with PDAC and missing manual

annotations were removed. All images from D02 were removed from the data set since they are part of the development pool.

D18 [20, 21, 22, 23, 24] was processed via connected component analysis to convert semantic segmentations into instance segmentations.

D22 [170, 83] was processed via connected component analysis to convert semantic segmentations into instance segmentations. The additional file indicating the number of objects per image was used to ensure the correct number of objects per image.

Configurations

Table B.49: Configuration of parameters for data set: MSD-L. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data, while '3dlr1' refers to the low-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d	3dlr1
Preprocessing		
Target spacing (mm)	1.00 x 0.76 x 0.76	2.00 x 1.52 x 1.52
Training		
Patch Size	128 x 128 x 128	112 x 128 x 128
Blueprint Parameters		
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2	2
Detector Parameters		
Anchor sizes at first anchor level	Ax0: 2.0, 5.0, 8.0	Ax0: 3.0, 5.0, 10.0
	Ax1: 7.0, 9.0, 12.0	Ax1: 4.0, 6.0, 14.0
	Ax2: 6.0, 8.0, 11.0	Ax2: 4.0, 6.0, 12.0
Estimated number of objects in patch	16	27

Table B.50: Configuration of parameters for data set: MSD-P. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	2.50 x 0.80 x 0.80
Training	
Patch Size	40 x 256 x 224
Blueprint Parameters	
Downsampling strides	[1, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2], [1, 2, 1]
Convolutional kernels	[1, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 4.0, 6.0, 10.0 Ax1: 26.0, 15.0, 20.0 Ax2: 21.0, 16.0, 26.0
Estimated number of objects in patch	1

Table B.51: Configuration of parameters for data set: MSD-HV. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data, while '3dlr1' refers to the low-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d	3dlr1
Preprocessing		
Target spacing (mm)	1.50 x 0.79 x 0.79	3.00 x 1.58 x 1.58
Training		
Patch Size	64 x 224 x 192	64 x 192 x 192
Blueprint Parameters		
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2	2
Detector Parameters		
Anchor sizes at first anchor level	Ax0: 5.0, 8.0, 14.0	Ax0: 3.0, 5.0, 8.0
	Ax1: 9.0, 12.0, 16.0	Ax1: 6.0, 8.0, 11.0
	Ax2: 9.0, 12.0, 15.0	Ax2: 6.0, 8.0, 11.0
Estimated number of objects in patch	4	5

Table B.52: Configuration of parameters for data set: MSD-C. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	3.00 x 0.78 x 0.78
Training	
Patch Size	64 x 192 x 192
Blueprint Parameters	
Downsampling strides	[1, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]
Convolutional kernels	[1, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 2.0, 6.0, 4.0 Ax1: 13.0, 8.0, 18.0 Ax2: 12.0, 14.0, 19.0
Estimated number of objects in patch	1

Table B.53: Configuration of parameters for data set: CADA. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	0.54 x 0.54 x 0.54
Training	
Patch Size	128 x 160 x 128
Blueprint Parameters	
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 5.0, 7.0, 16.0
	Ax1: 5.0, 7.0, 15.0
	Ax2: 5.0, 7.0, 16.0
Estimated number of objects in patch	2

Table B.54: Configuration of parameters for data set: RibFrac. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	1.25 x 0.74 x 0.74
Training	
Patch Size	96 x 160 x 160
Blueprint Parameters	
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 5.0, 7.0, 9.0
	Ax1: 9.0, 13.0, 19.0
	Ax2: 8.0, 11.0, 16.0
Estimated number of objects in patch	8

Table B.55: Configuration of parameters for data set: KiTS21. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data, while '3dlr1' refers to the low-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d	3dlr1
Preprocessing		
Target spacing (mm)	0.78 x 0.78 x 0.78	1.56 x 1.56 x 1.56
Training		
Patch Size	160 x 128 x 128	160 x 128 x 128
Blueprint Parameters		
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2	2
Detector Parameters		
Anchor sizes at first anchor level	Ax0: 7.0, 10.0, 13.0	Ax0: 4.0, 6.0, 19.0
	Ax1: 6.0, 8.0, 11.0	Ax1: 4.0, 6.0, 18.0
	Ax2: 6.0, 8.0, 11.0	Ax2: 4.0, 6.0, 18.0
Estimated number of objects in patch	6	6

Table B.56: Configuration of parameters for data set: PICA1. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data, while '3dlr1' refers to the low-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d	3dlr1
Preprocessing		
Target spacing (mm)	3.00 x 0.50 x 0.50	6.00 x 1.00 x 1.00
Training		
Patch Size	16 x 320 x 320	12 x 192 x 192
Blueprint Parameters		
Downsampling strides	[1, 2, 2], [1, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2], [1, 2, 2]	[1, 2, 2], [1, 2, 2], [2, 2, 2] [1, 2, 2], [1, 2, 2]
Convolutional kernels	[1, 3, 3], [1, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3]	[1, 3, 3], [1, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2	2
Detector Parameters		
Anchor sizes at first anchor level	Ax0: 2.0, 3.0, 4.0 Ax1: 9.0, 12.0, 16.0 Ax2: 11.0, 15.0, 19.0	Ax0: 3.0, 2.0, 15.0 Ax1: 7.0, 10.0, 15.0 Ax2: 6.0, 10.0, 16.0
Estimated number of objects in patch	1	1

Table B.57: Configuration of parameters for data set: ADAM. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	0.50 x 0.36 x 0.36
Training	
Patch Size	56 x 224 x 224
Blueprint Parameters	
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [1, 2, 2], [1, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 6.0, 5.0, 8.0
	Ax1: 8.0, 6.0, 11.0
	Ax2: 6.0, 8.0, 12.0
Estimated number of objects in patch	2

Table B.58: Configuration of parameters for data set: LIDC. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	1.38 x 0.70 x 0.70
Training	
Patch Size	80 x 192 x 160
Blueprint Parameters	
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 4.0, 6.0, 3.0
	Ax1: 10.0, 8.0, 6.0
	Ax2: 8.0, 6.0, 11.0
Estimated number of objects in patch	3

Table B.59: Configuration of parameters for data set: KiPA. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data, while '3dlr1' refers to the low-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d	3dlr1
Preprocessing		
Target spacing (mm)	0.62 x 0.62 x 0.62	1.25 x 1.25 x 1.25
Training		
Patch Size	160 x 128 x 128	112 x 80 x 80
Blueprint Parameters		
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3]
First Anchor Level	2	1
Detector Parameters		
Anchor sizes at first anchor level	Ax0: 10.0, 12.0, 17.0	Ax0: 5.0, 6.0, 17.0
	Ax1: 12.0, 10.0, 17.0	Ax1: 6.0, 5.0, 15.0
	Ax2: 10.0, 14.0, 16.0	Ax2: 5.0, 6.0, 15.0
Estimated number of objects in patch	1	1

Table B.60: Configuration of parameters for data set: MRA-A. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	0.70 x 0.41 x 0.41
Training	
Patch Size	64 x 224 x 192
Blueprint Parameters	
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 3.0, 6.0, 10.0
	Ax1: 5.0, 6.0, 16.0
	Ax2: 5.0, 6.0, 16.0
Estimated number of objects in patch	2

Table B.61: Configuration of parameters for data set: CT-PC. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	1.00 x 0.75 x 0.75
Training	
Patch Size	128 x 128 x 160
Blueprint Parameters	
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 6.0, 5.0, 8.0
	Ax1: 5.0, 7.0, 9.0
	Ax2: 5.0, 7.0, 10.0
Estimated number of objects in patch	6

Table B.62: Configuration of parameters for data set: DUKE. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data, while '3dlr1' refers to the low-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d	3dlr1
Preprocessing		
Target spacing (mm)	1.00 x 0.70 x 0.70	2.00 x 1.41 x 1.41
Training		
Patch Size	80 x 192 x 192	80 x 192 x 192
Blueprint Parameters		
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2	2
Detector Parameters		
Anchor sizes at first anchor level	Ax0: 5.0, 8.0, 12.0 Ax1: 10.0, 13.0, 17.0 Ax2: 9.0, 12.0, 16.0	Ax0: 5.0, 7.0, 9.0 Ax1: 9.0, 12.0, 16.0 Ax2: 8.0, 14.0, 10.0
Estimated number of objects in patch	1	1

Table B.63: Configuration of parameters for data set: BraTS-M. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	1.50 x 1.00 x 1.00
Training	
Patch Size	80 x 160 x 128
Blueprint Parameters	
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 3.0, 2.0, 14.0
	Ax1: 3.0, 4.0, 5.0
	Ax2: 3.0, 4.0, 23.0
Estimated number of objects in patch	28

Table B.64: Configuration of parameters for data set: CT-PaCS. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data, while '3dlr1' refers to the low-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d	3dlr1
Preprocessing		
Target spacing (mm)	1.50 x 0.74 x 0.74	3.00 x 1.48 x 1.48
Training		
Patch Size	80 x 160 x 160	80 x 160 x 160
Blueprint Parameters		
Downsampling strides	[1, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]	[1, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]
Convolutional kernels	[1, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]	[1, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2	2
Detector Parameters		
Anchor sizes at first anchor level	Ax0: 4.0, 6.0, 8.0 Ax1: 9.0, 12.0, 15.0 Ax2: 10.0, 13.0, 17.0	Ax0: 4.0, 2.0, 6.0 Ax1: 7.0, 9.0, 12.0 Ax2: 8.0, 10.0, 13.0
Estimated number of objects in patch	1	1

Table B.65: Configuration of parameters for data set: MELA. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data, while '3dlr1' refers to the low-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d	3dlr1	3dlr2
Preprocessing			
Target spacing (mm)	0.70 x 0.72 x 0.72	1.40 x 1.43 x 1.43	2.80 x 2.87 x 2.87
Training			
Patch Size	160 x 128 x 128	160 x 128 x 128	128 x 128 x 128
Blueprint Parameters			
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2	2	2
Detector Parameters			
Anchor sizes at first anchor level	Ax0: 8.0, 10.0, 14.0 Ax1: 11.0, 19.0, 14.0 Ax2: 9.0, 12.0, 16.0	Ax0: 10.0, 8.0, 13.0 Ax1: 9.0, 15.0, 12.0 Ax2: 8.0, 10.0, 13.0	Ax0: 6.0, 8.0, 5.0 Ax1: 6.0, 8.0, 10.0 Ax2: 5.0, 8.0, 6.0
Estimated number of objects in patch	1	1	1

Table B.66: Configuration of parameters for data set: VALDO-M . The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	0.80 x 0.49 x 0.49
Training	
Patch Size	80 x 192 x 128
Blueprint Parameters	
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 7.0, 10.0, 15.0
	Ax1: 4.0, 6.0, 8.0
	Ax2: 5.0, 6.0, 8.0
Estimated number of objects in patch	5

Table B.67: Configuration of parameters for data set: LNDb. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	1.00 x 0.64 x 0.64
Training	
Patch Size	96 x 160 x 160
Blueprint Parameters	
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 2.0, 4.0, 3.0
	Ax1: 4.0, 5.0, 7.0
	Ax2: 5.0, 7.0, 4.0
Estimated number of objects in patch	4

Table B.68: Configuration of parameters for data set: LUNA16. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	1.25 x 0.70 x 0.70
Training	
Patch Size	80 x 192 x 160
Blueprint Parameters	
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 3.0, 8.0, 2.0
	Ax1: 4.0, 5.0, 13.0
	Ax2: 4.0, 5.0, 13.0
Estimated number of objects in patch	3

Table B.69: Configuration of parameters for data set: PN9. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	1.00 x 1.00 x 1.00
Training	
Patch Size	96 x 128 x 128
Blueprint Parameters	
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [1, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 3.0, 4.0, 12.0
	Ax1: 5.0, 6.0, 9.0
	Ax2: 5.0, 6.0, 9.0
Estimated number of objects in patch	7

Table B.70: Configuration of parameters for data set: CTA-A. The rows denote different parameters of the plan, while different columns denote different plans. The '3d' plan refers to the full-resolution data. Convolutional kernels and downsampling strides are sorted by their level. 'Ax[0-2]' refer to different axes of the preprocessed image. Table reproduced from [82].

Plan: D3V002	3d
Preprocessing	
Target spacing (mm)	0.40 x 0.46 x 0.46
Training	
Patch Size	160 x 128 x 128
Blueprint Parameters	
Downsampling strides	[2, 2, 2], [2, 2, 2], [2, 2, 2] [2, 2, 2], [2, 2, 2]
Convolutional kernels	[3, 3, 3], [3, 3, 3], [3, 3, 3] [3, 3, 3], [3, 3, 3], [3, 3, 3]
First Anchor Level	2
Detector Parameters	
Anchor sizes at first anchor level	Ax0: 7.0, 11.0, 9.0
	Ax1: 6.0, 8.0, 11.0
	Ax2: 7.0, 9.0, 11.0
Estimated number of objects in patch	2

List of Acronyms

3DRA	Rotational X-Ray Angiographies
ADAM	Aneurysm Detection And segMentation Challenge
AIS	Acute Ischemic Stroke
ANN	Artificial Neural Network
ATSS	Adaptive Training Sample Selection
AUROC	Area under the Receiver Operating Curve
BCE	Binary Cross Entropy
BVM	German Conference on Medical Image Computing
CAD	Computer Aided Diagnosis
CE	Cross Entropy
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CTA	Computed Tomography Angiography
CTP	Computed Tomographic Perfusion
CT	Computed Tomography
DETR	DEtection TRansformer
DL	Deep Learning
DSB	Data Science Bowl
DSC	Dice Similarity Coefficient

FCOS	Fully Convolutional One-Stage Object Detection
FC	Fully Connected
FFN	Feedforward Neural Network
FN	False Negative
FPN	Feature Pyramid Network
FPPI	False Positives per Image
FPR	False Positive Reduction
FP	False Positive
FROC	Free-response Receiver Operating Characteristic
GIoU	Generalised Intersection over Union
GPU	Graphics Processing Unit
HGS	High-grade Stenosis
HNM	Hard Negative Mining
HTC	Hybrid Task Cascade
HU	Hounsfield Unit
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IoU	Intersection over Union
LReLU	Leaky Rectified Linear Unit
LVO	Large Vessel Occlusion
mAP	mean Average Precision
MDT	Medical Detection Toolkit
MELA	Mediastinal Lesion Analysis
MeVO	Medium Vessel Occlusion
MHA	Multi-Head Attention
MHSA	Multi-Head Self-Attention
MICCAI	The Medical Image Computing and Computer Assisted Intervention
MITK	Medical Imaging Interaction Toolkit
MLP	Multilayer Perceptron

ML Machine Learning
MRI Magnetic Resonance Image
MSD Medical Segmentation Decathlon
NCCT Non-Contrast Computed Tomography
NMS Non-maximum suppression
NPV Negative Predictive Value
NSD Normalised Surface Distance
PACS Picture Archiving and Communication System
PPV Positive Predictive Value
R-CNN Regions with CNN features
RB Reference Boxes
ReLU Rectified Linear Unit
ROC Receiver Operating Curve
RoI Align Region of Interest Align
RoI Pooling Region of Interest Pooling
RoI Region of Interest
RPN Region Proposal Network
RP Reference Points
SAH Subarachnoid Haemorrhage
SAM Segment Anything Model
SGD Stochastic Gradient Descent
SSD Single Shot MultiBox Detector
SVD Singular Value Decomposition
SVM Support Vector Machine
SWA Stochastic Weight Averaging
TN True Negative
TOF-MRA Time-of-Flight MR-Angiography
TP True Positive

List of Acronyms

UIA Unruptured Intracranial Aneurysm

UKB University Clinic Bonn

UKHD University Clinic Heidelberg

ViT Vision Transformer

VRAM Video Random Access Memory

WBC Weighted Box Clustering

YOLO You Only Look Once

List of Figures

1.1	Overview of medical image modalities and object structures requiring diagnostic decision making.	3
1.2	Different levels of abstraction for Computer Aided Diagnosis support. . .	4
1.3	RQ1.1 - International Challenge: Design aspects of our submission to the MELA 2022 Challenge.	7
1.4	RQ1.2 - Clinical Application: Design aspects for developing a clinically relevant vessel occlusion detection model.	9
1.5	RQ1.3 - Overview of anchor-based and direct set prediction models. . . .	10
1.6	RQ 2: Overview of current task-specific design versus proposed self-configuring method design for detection models.	13
2.1	Structural overview of Regions with CNN features (R-CNN) detection method.	18
2.2	Structural overview of the Fast R-CNN detection method.	20
2.3	Structural overview of Faster R-CNN detection method.	22
2.4	Structural overview of You Only Look Once (YOLO) detection method. . .	25
2.5	Structural overview of Single Shot MultiBox Detector (SSD) detection method.	27
2.6	Structural overview of Retina Net detection method.	30
2.7	Matching between predictions and ground truth of DEtection TRansformer (DETR) detection method.	33
2.8	Architectural components of DEtection TRansformer (DETR) detection method.	34
2.9	Transformer architecture of DEtection TRansformer (DETR) detection method.	35

2.10	Schematic of Deformable Attention.	37
2.11	Object-level evaluation procedure.	40
4.1	Visualisation of the bounding box and the converted spherical segmentation annotations for the MELA data set.	59
4.2	Visualisation of the Retina U-Net architecture for the MELA data set. . .	61
4.3	Inference Procedure for Vessel Occlusion Detection.	65
4.4	Retina Net architecture for vessel occlusion detection.	66
4.5	Removed object clusters from KiTS19 data set.	69
4.6	Architectural patterns of DETR, Conditional DETR and DINO DETR. . .	70
4.7	Development, Generalisation and Benchmarking Pool of nnDetection. . .	75
4.8	One-Stage Anchor-based Architectural Blueprint.	85
4.9	One-Stage Anchor-based Model Configuration.	87
4.10	Two-Stage RoI Head Architecture.	88
4.11	Two-Stage Anchor-based Model Configuration.	89
4.12	Direct Set Prediction Architectural Blueprint.	91
4.13	Direct Set Prediction Model Configuration.	92
4.14	Overview of the Inference Pipeline.	95
5.1	Cross-validation results for MELA baseline and final ensemble.	101
5.2	Patient and Object level performance of vessel occlusion detection model across Heidelberg, FAST and UKB cohorts.	104
5.3	Object Level Performance Across Vessel Occlusion Subgroups on the UKHD Test Set.	106
5.4	Patient-Level Performance on External Cohorts with and without Inclusion of HGS.	107
5.5	Inference Time on Heidelberg Test Set.	110
5.6	mean Average Precision of Detection Transformers on four data sets. . .	112
5.7	FROC curves of Detection Transformers at IoU threshold of 0.1.	113
5.8	Aggregated results from the test sets of the generalisation pool.	116
5.9	Ranking histogram for each test set of the generalisation pool.	118
5.10	Performance comparison with different metrics on the test sets of the generalisation pool.	119
5.11	Performance comparison with different metrics on the test sets of the generalisation pool.	120
5.12	Benchmarking results against task-specific models on LUNA16.	122
5.13	Benchmarking results against task-specific models on PN9.	123
5.14	Benchmarking results against task-specific models on CTA-A.	124
6.1	Qualitative results of cross-validation results on MELA data set.	127
B.1	Cross-validation performance on development pool.	152
B.2	Cross-validation performance on generalisation pool.	152

B.3	Test set performance on generalisation pool.	153
B.4	Test set performance on development pool.	153

List of Tables

1.1	Overview of research questions in this thesis.	15
3.1	Overview of existing literature on vessel occlusion detection.	49
3.2	Overview of existing literature on lung nodule detection on LIDC, LUNA16 and PN9.	51
4.1	Shows two different augment schemes for the MELA data set.	62
4.2	Hyperparameters of DETR models for selected data sets.	72
4.3	Overview of empirical inference parameters.	96
4.4	Overview of empirical parameters for nnU-Net Plus.	98
5.1	Shows mAP and FROC results of MELA models.	101
5.2	Performance Comparison of HD-CTA, CS1 and CS2 on UKB Cohort. . . .	108
5.3	Performance Comparison of HD-CTA and CS2 on FAST Cohort.	109
B.1	Scanner, Convolution Kernel and Slice Thickness Details Across Vessel Occlusion Cohorts.	148
B.2	Acquisition Phase Across Vessel Occlusion Test Sets.	148
B.3	Object-level results in the training set (cross-validation) of the Heidelberg cohort.	149
B.4	Object-level results in the test set of the Heidelberg cohort.	149
B.5	Object-level results in the test set of the FAST cohort.	149
B.6	Object-level results in the test set of the UKB cohort.	150
B.7	Patient-level performance of HD-CTA on Heidelberg cohort.	150
B.8	Patient-level performance of HD-CTA on FAST cohort.	150
B.9	Patient-level performance of HD-CTA on UKB cohort.	150

B.10	Number of false positives on object level in FAST and UKB cohort shown for different acquisition phases.	150
B.11	mean Average Precision at an IoU threshold of 0.1 for DETR models and Retina U-Net across four data sets.	151
B.12	mean Average Precision at an IoU threshold of 0.5 for DETR models and Retina U-Net across four data sets.	151
B.13	Cross-Validation detection performance on the MSD-L data set.	154
B.14	Test set detection performance on the MSD-L data set.	154
B.15	Cross-Validation detection performance on the MSD-P data set.	155
B.16	Test set detection performance on the MSD-P data set.	155
B.17	Cross-Validation detection performance on the MSD-HV data set.	156
B.18	Test set detection performance on the MSD-HV data set.	156
B.19	Cross-Validation detection performance on the MSD-C data set.	157
B.20	Test set detection performance on the MSD-C data set.	157
B.21	Cross-Validation detection performance on the CADA data set.	158
B.22	Cross-Validation detection performance on the RibFrac data set.	158
B.23	Cross-Validation detection performance on the KiTS21 data set.	159
B.24	Cross-Validation detection performance on the PICAI data set.	159
B.25	Cross-Validation detection performance on the ADAM data set.	160
B.26	Cross-Validation detection performance on the LIDC data set.	160
B.27	Cross-Validation detection performance on the KiPA data set.	161
B.28	Test set detection performance on the KiPA data set.	161
B.29	Cross-Validation detection performance on the MRA-A data set.	162
B.30	Test set detection performance on the MRA-A data set.	162
B.31	Cross-Validation detection performance on the CT-PC data set.	163
B.32	Test set detection performance on the CT-PC data set.	163
B.33	Cross-Validation detection performance on the DUKE data set.	164
B.34	Test set detection performance on the DUKE data set.	164
B.35	Cross-Validation detection performance on the BraTS-M data set.	165
B.36	Test set detection performance on the BraTS-M data set.	165
B.37	Cross-Validation detection performance on the CT-PaCS data set.	166
B.38	Test set detection performance on the CT-PaCS data set.	166
B.39	Cross-Validation detection performance on the MELA data set.	167
B.40	Test set detection performance on the MELA data set.	167
B.41	Cross-Validation detection performance on the VALDO-M data set.	168
B.42	Test set detection performance on the VALDO-M data set.	168
B.43	Cross-Validation detection performance on the LNDb data set.	169
B.44	Test set detection performance on the LNDb data set.	169
B.45	Overview of LUNA16 results.	170
B.46	Overview of PN9 results.	171
B.47	Overview of CTA-A internal test set results.	171
B.48	Overview of CTA-A external test set results.	171

B.49	Configuration of parameters for data set: MSD-L.	173
B.50	Configuration of parameters for data set: MSD-P.	174
B.51	Configuration of parameters for data set: MSD-HV.	175
B.52	Configuration of parameters for data set: MSD-C.	176
B.53	Configuration of parameters for data set: CADA.	177
B.54	Configuration of parameters for data set: RibFrac.	178
B.55	Configuration of parameters for data set: KiTS21.	179
B.56	Configuration of parameters for data set: PICAI.	180
B.57	Configuration of parameters for data set: ADAM.	181
B.58	Configuration of parameters for data set: LIDC.	182
B.59	Configuration of parameters for data set: KiPA.	183
B.60	Configuration of parameters for data set: MRA-A.	184
B.61	Configuration of parameters for data set: CT-PC.	185
B.62	Configuration of parameters for data set: DUKE.	186
B.63	Configuration of parameters for data set: BraTS-M.	187
B.64	Configuration of parameters for data set: CT-PaCS.	188
B.65	Configuration of parameters for data set: MELA.	189
B.66	Configuration of parameters for data set: VALDO-M.	190
B.67	Configuration of parameters for data set: LNDb.	191
B.68	Configuration of parameters for data set: LUNA16.	192
B.69	Configuration of parameters for data set: PN9.	193
B.70	Configuration of parameters for data set: CTA-A.	194

List of Algorithms

Bibliography

- [1] H. Gernsheim. *A Concise History of Photography*. Dover photography collections. Dover Publications, 1986. ISBN: 9780486251288. URL: <https://books.google.de/books?id=GDSRJQ3BZ5EC> (cit. on p. 1).
- [2] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022 (cit. on p. 1).
- [3] Mayu Nishimura, Suzy Scherf, and Marlene Behrmann. “Development of object recognition in humans”. In: *F1000 biology reports* 1 (2009) (cit. on p. 1).
- [4] Hans Moravec. “Mind Children: The Future of Robot and Human Intelligence”. In: *Harvard UP* (1988) (cit. on p. 1).
- [5] Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009 (cit. on p. 1).
- [6] Giuseppe Lo Presti, Marina Carbone, Damiano Ciriaci, Daniele Aramini, Mauro Ferrari, and Vincenzo Ferrari. “Assessment of DICOM viewers capable of loading patient-specific 3D models obtained by different segmentation platforms in the operating room”. In: *Journal of digital imaging* 28 (2015), pp. 518–527 (cit. on p. 2).
- [7] Mayo Clinic. *Minimally invasive heart surgery*. <https://www.mayoclinic.org/tests-procedures/minimally-invasive-heart-surgery/about/pac-20384895>. Accessed: 2024-11-27. 2023 (cit. on p. 2).
- [8] Cleveland Clinic. *Thrombectomy*. Accessed: 2024-11-27. 2024. URL: <https://my.clevelandclinic.org/health/treatments/22897-thrombectomy> (cit. on p. 2).
- [9] National Health Service. *What happens during radiotherapy*. Accessed: 2025-02-02. URL: <https://www.nhs.uk/conditions/radiotherapy/what-happens/> (cit. on p. 2).

- [10] Cancer Research UK. *Planning your external radiotherapy*. Accessed: 2025-02-02. URL: <https://www.cancerresearchuk.org/about-cancer/treatment/radiotherapy/external/planning/your-planning/> (cit. on p. 2).
- [11] Liang Jin, Jiancheng Yang, Kaiming Kuang, Bingbing Ni, Yiyi Gao, Yingli Sun, Pan Gao, Weiling Ma, Mingyu Tan, and Hui Kang. “Deep-learning-assisted detection and segmentation of rib fractures from CT scans: Development and validation of FracNet”. In: *EBioMedicine* 62 (2020). ISSN: 2352-3964. URL: [https://www.thelancet.com/journals/ebiom/article/PIIS2352-3964\(20\)30482-5/fulltext](https://www.thelancet.com/journals/ebiom/article/PIIS2352-3964(20)30482-5/fulltext) (cit. on pp. 2, 3, 10, 18, 25, 68, 74, 78, 97, 102, 112, 113, 172).
- [12] Jiancheng Yang, Rui Shi, Liang Jin, Xiaoyang Huang, Kaiming Kuang, Donglai Wei, Shixuan Gu, Jianying Liu, Pengfei Liu, and Zhizhong Chai. “Deep Rib Fracture Instance Segmentation and Classification from CT on the RibFrac Challenge”. In: *arXiv preprint arXiv:2402.09372* (2024). URL: <https://arxiv.org/abs/2402.09372> (cit. on pp. 2, 3, 10, 18, 25, 68, 74, 102, 112, 113, 172).
- [13] Arnaud Arindra Adiyoso Setio, Alberto Traverso, Thomas De Bel, Moira S N Berens, Cas Van Den Bogaard, Piergiorgio Cerello, Hao Chen, Qi Dou, Maria Evelina Fantacci, and Bram Geurts. “Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge”. In: *Medical image analysis* 42 (2017), pp. 1–13. ISSN: 1361-8415. URL: <https://pubmed.ncbi.nlm.nih.gov/28732268/> (cit. on pp. 2, 3, 20, 22, 27, 30, 33, 34, 50, 51, 53, 68, 74, 76, 78, 112, 113, 121, 172).
- [14] Jie Mei, Ming-Ming Cheng, Gang Xu, Lan-Ruo Wan, and Huan Zhang. “SANet: A slice-aware network for pulmonary nodule detection”. In: *IEEE transactions on pattern analysis and machine intelligence* 44 (8 2021), pp. 4374–4387. ISSN: 0162-8828. URL: <https://ieeexplore.ieee.org/document/9373930> (cit. on pp. 2, 6, 14, 50, 51, 54, 76, 78, 100, 102, 114, 115, 122, 131, 132, 137, 172).
- [15] Nicholas Heller, Fabian Isensee, Dasha Trofimova, Resha Tejpaul, Zhongchen Zhao, Huai Chen, Lisheng Wang, Alex Golts, Daniel Khapun, and Daniel Shats. “The KiTS21 Challenge: Automatic segmentation of kidneys, renal tumors, and renal cysts in corticomedullary-phase CT”. In: *arXiv preprint arXiv:2307.01984* (2023). URL: <https://arxiv.org/abs/2307.01984> (cit. on pp. 2, 3, 74, 112, 113, 172).
- [16] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, Bram van Ginneken, Michel Bilello, Patrick Bilic, Patrick F Christ, Richard K G Do, Marc J Gollub, Stephan H Heckers, Henkjan Huisman, William R Jarnagin, Maureen K McHugo, Sandy Napel, Jennifer S Golia Pernicka, Kawal Rhode, Catalina Tobon-Gomez, Eugene Vorontsov, James A Meakin, Sebastien Ourselin, Manuel Wiesenfarth, Pablo Arbeláez, Byeonguk Bae, Sihong Chen, Laura Daza, Jianjiang Feng, Baochun He, Fabian Isensee,

-
- Yuanfeng Ji, Fucang Jia, Ildoo Kim, Klaus Maier-Hein, Dorit Merhof, Akshay Pai, Beomhee Park, Mathias Perslev, Ramin Rezaifar, Oliver Rippel, Ignacio Sarasua, Wei Shen, Jaemin Son, Christian Wachinger, Liansheng Wang, et al. “The Medical Segmentation Decathlon”. In: *Nature Communications* 13 (1 2022), p. 4128. ISSN: 2041-1723. DOI: 10.1038/s41467-022-30695-9. URL: <https://doi.org/10.1038/s41467-022-30695-9> (cit. on pp. 2, 56, 74, 172).
- [17] Matthias Ivantsits, Leonid Goubergrits, Jan-Martin Kuhnigk, Markus Huellebrand, Jan Bruening, Tabea Kossen, Boris Pfahringer, Jens Schaller, Andreas Spuler, and Titus Kuehne. “Detection and analysis of cerebral aneurysms based on X-ray rotational angiography-the CADA 2020 challenge”. In: *Medical image analysis* 77 (2022), p. 102333. ISSN: 1361-8415. URL: <https://www.sciencedirect.com/science/article/pii/S1361841521003789> (cit. on pp. 2, 3, 68, 74, 112, 113, 172).
- [18] Tommaso Di Noto, Guillaume Marie, Sebastien Tourbier, Yasser Alemán-Gómez, Oscar Esteban, Guillaume Saliou, Meritxell Bach Cuadra, Patric Hagmann, and Jonas Richiardi. “Towards automated brain aneurysm detection in TOF-MRA: open data, weak labels, and anatomical knowledge”. In: *Neuroinformatics* 21 (1 2023), pp. 21–34. ISSN: 1539-2791. URL: <https://link.springer.com/article/10.1007/s12021-022-09597-0> (cit. on pp. 2, 6, 49, 74, 78, 97, 132, 172).
- [19] Shuang Song, Rui Xu, Yong Luo, Bo Du, Zhijian Yang, Jiancheng Yang, Kaiming Kuang, Bingbing Ni, Chang Chen, Deping Zhao, Dong Xie, Xiwen Sun, Jingyun Shi, Yunlang She, Mengmeng Zhao, Jiajun Deng, Junqi Wu, and Tingting Wang. *Mediastinal Lesion Analysis*. Mar. 2022. DOI: 10.5281/zenodo.6361949. URL: <https://doi.org/10.5281/zenodo.6361949> (cit. on pp. 2, 3, 7, 74, 102, 131, 172).
- [20] José Luis Molinuevo, Nina Gramunt, Juan Domingo Gispert, Karine Fauria, Manel Esteller, Carolina Minguillon, Gonzalo Sánchez-Benavides, Gema Huesa, Sebastián Morán, and Rafael Dal-Ré. “The ALFA project: a research platform to identify early pathophysiological features of Alzheimer’s disease”. In: *Alzheimer’s & Dementia: Translational Research & Clinical Interventions* 2 (2 2016), pp. 82–92. ISSN: 2352-8737. URL: <https://alz-journals.onlinelibrary.wiley.com/doi/10.1016/j.trci.2016.02.003> (cit. on pp. 2, 74, 115, 173).
- [21] Siana Jones, Therese Tillin, Chloe Park, Suzanne Williams, Alicja Rapala, Lamia Al Saikhan, Sophie V Eastwood, Marcus Richards, Alun D Hughes, and Nishi Chaturvedi. “Cohort Profile Update: Southall and Brent Revisited (SABRE) study: a UK population-based comparison of cardiovascular disease and diabetes in people of European, South Asian and African Caribbean heritage”. In: *International journal of epidemiology* 49 (5 2020), 1441–1442e. ISSN: 0300-5771. URL: <https://>

- academic.oup.com/ije/article/49/5/1441/5922749?login=true (cit. on pp. 2, 74, 115, 173).
- [22] Therese Tillin, Nita G Forouhi, Paul M McKeigue, and Nish Chaturvedi. “Southall And Brent REvisited: Cohort profile of SABRE, a UK population-based comparison of cardiovascular disease and diabetes in people of European, Indian Asian and African Caribbean origins”. In: *International journal of epidemiology* 41 (1 2012), pp. 33–42. ISSN: 1464-3685. URL: <https://pubmed.ncbi.nlm.nih.gov/21044979/> (cit. on pp. 2, 74, 115, 173).
 - [23] M Arfan Ikram, Aad van der Lugt, Wiro J Niessen, Peter J Koudstaal, Gabriel P Krestin, Albert Hofman, Daniel Bos, and Meike W Vernooij. “The Rotterdam Scan Study: design update 2016 and main findings”. In: *European journal of epidemiology* 30 (2015), pp. 1299–1315. ISSN: 0393-2990. URL: <https://link.springer.com/article/10.1007/s10654-015-0105-7> (cit. on pp. 2, 74, 115, 173).
 - [24] Carole H Sudre, Kimberlin Van Wijnen, Florian Dubost, Hieab Adams, David Atkinson, Frederik Barkhof, Mahlet A Birhanu, Esther E Bron, Robin Camarasa, and Nish Chaturvedi. “Where is VALDO? VAScular lesions detection and segmentation challenge at MICCAI 2021”. In: *Medical Image Analysis* 91 (2024), p. 103029. ISSN: 1361-8415. URL: <https://www.sciencedirect.com/science/article/pii/S136184152300289X> (cit. on pp. 2, 74, 115, 173).
 - [25] Ashirbani Saha, Michael R Harowicz, Lars J Grimm, Connie E Kim, Sujata V Ghate, Ruth Walsh, and Maciej A Mazurowski. “A machine learning approach to radiogenomics of breast cancer: a study of 922 subjects and 529 DCE-MRI features”. In: *British journal of cancer* 119 (4 2018), pp. 508–516. ISSN: 0007-0920. URL: <https://www.nature.com/articles/s41416-018-0185-8> (cit. on pp. 2, 3, 74, 78, 172).
 - [26] A. Saha, M. R. Harowicz, L. J. Grimm, J. Weng, E. H. Cain, C. E. Kim, S. V. Ghate, R. Walsh, and M. A. Mazurowski. *Dynamic contrast-enhanced magnetic resonance images of breast cancer patients with tumor locations*. The Cancer Imaging Archive, 2021. DOI: 10.7937/TCIA.e3sv-re93. URL: <https://doi.org/10.7937/TCIA.e3sv-re93> (cit. on pp. 2, 3, 74, 172).
 - [27] Rebecca Smith-Bindman, Diana L Miglioretti, and Eric B Larson. “Rising use of diagnostic medical imaging in a large integrated health system”. In: *Health affairs* 27.6 (2008), pp. 1491–1502 (cit. on p. 2).
 - [28] NHS England. *Diagnostic Imaging Dataset Annual Statistical Release 2021/22*. <https://www.england.nhs.uk/statistics/wp-content/uploads/sites/2/2022/12/Annual-Statistical-Release-2021-22-PDF-1.3-MB.pdf>. Accessed: 2024-10-03. 2024 (cit. on p. 2).

-
- [29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition. arXiv e-prints”. In: *arXiv preprint arXiv:1512.03385* 10 (2015) (cit. on pp. 5, 26, 29).
- [30] Irwan Bello, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. “Revisiting resnets: Improved training and scaling strategies”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 22614–22627. URL: <https://arxiv.org/abs/2103.07579> (cit. on p. 5).
- [31] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708 (cit. on pp. 5, 47, 52).
- [32] Geoff Pleiss, Danlu Chen, Gao Huang, Tongcheng Li, Laurens Van Der Maaten, and Kilian Q Weinberger. “Memory-efficient implementation of densenets”. In: *arXiv preprint arXiv:1707.06990* (2017). URL: <https://arxiv.org/abs/1707.06990> (cit. on p. 5).
- [33] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. “A convnet for the 2020s”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11976–11986. URL: <https://arxiv.org/abs/2201.03545> (cit. on p. 5).
- [34] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvain Gelly. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020). URL: <https://openreview.net/forum?id=YicbFdNTTy> (cit. on pp. 5, 24, 31).
- [35] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. “Focal loss for dense object detection”. In: 2017, pp. 2980–2988. URL: <https://arxiv.org/abs/1708.02002> (cit. on pp. 5, 9, 28–30, 32, 38, 53, 54, 64, 65, 71, 77, 83, 84, 90, 122, 171).
- [36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems* 28 (2015). URL: <https://arxiv.org/abs/1506.01497> (cit. on pp. 5, 9, 21–23, 26, 32, 51, 52, 77, 83, 86, 122, 171).
- [37] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask r-cnn”. In: 2017, pp. 2961–2969. URL: <https://arxiv.org/abs/1703.06870> (cit. on pp. 5, 9, 23, 32, 53, 77, 83, 86, 138).

- [38] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. “End-to-end object detection with transformers”. In: *European conference on computer vision*. Springer, 2020, pp. 213–229. URL: <https://arxiv.org/abs/2005.12872> (cit. on pp. 5, 9, 33–35, 69–71, 77).
- [39] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. “Dino: Detr with improved denoising anchor boxes for end-to-end object detection”. In: *arXiv preprint arXiv:2203.03605* (2022) (cit. on pp. 5, 9, 38, 54, 69–72, 130).
- [40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18. Springer, 2015, pp. 234–241. ISBN: 3319245732 (cit. on pp. 5, 47, 49, 51–53).
- [41] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE transactions on pattern analysis and machine intelligence* 40 (4 2017), pp. 834–848 (cit. on p. 5).
- [42] Lena Maier-Hein, Matthias Eisenmann, Annika Reinke, Sinan Onogur, Marko Stankovic, Patrick Scholz, Tal Arbel, Hrvoje Bogunovic, Andrew P Bradley, and Aaron Carass. “Why rankings of biomedical image analysis competitions should be interpreted with care”. In: *Nature communications* 9 (1 2018), p. 5217. ISSN: 2041-1723 (cit. on pp. 6, 74).
- [43] Arnaud Arindra Adiyoso Setio, Alberto Traverso, Thomas De Bel, Moira S N Berens, Cas Van Den Bogaard, Piergiorgio Cerello, Hao Chen, Qi Dou, Maria Evelina Fantacci, and Bram Geurts. “Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge”. In: *Medical image analysis* 42 (2017), pp. 1–13. ISSN: 1361-8415. URL: <https://arxiv.org/abs/1612.08012> (cit. on pp. 6, 50, 51, 54, 76, 78, 102, 114, 115, 121, 132, 137, 172).
- [44] Kimberley M Timmins, Irene C van der Schaaf, Edwin Bennink, Ynte M Ruigrok, Xingle An, Michael Baumgartner, Pascal Bourdon, Riccardo De Feo, Tommaso Di Noto, Florian Dubost, Augusto Fava-Sanches, Xue Feng, Corentin Giroud, Inteneural Group, Minghui Hu, Paul F Jaeger, Juhana Kaiponen, Michał Klimont, Yuexiang Li, Hongwei Li, Yi Lin, Timo Loehr, Jun Ma, Klaus H Maier-Hein, Guillaume Marie, Bjoern Menze, Jonas Richiardi, Saifeddine Rjiba, Dhaval Shah, Suprosanna Shit, Jussi Tohka, Thierry Urruty, Urszula Walińska, Xiaoping Yang, Yunqiao Yang, Yin Yin, Birgitta K Velthuis, and Hugo J Kuijf. “Comparing methods of detecting and segmenting unruptured intracranial aneurysms on TOF-MRAS: The ADAM challenge”. In: *NeuroImage* 238 (2021), p. 118216. ISSN: 1053-8119. DOI: <https://doi.org/10.1016/j.neuroimage.2021.118216>. URL: <https://www.>

sciencedirect.com/science/article/pii/S1053811921004936 (cit. on pp. 6, 74, 131, 138, 172).

- [45] Hwejin Jung, Bumsoo Kim, Inyeop Lee, Minhwan Yoo, Junhyun Lee, Sooyoun Ham, Okhee Woo, and Jaewoo Kang. “Detection of masses in mammograms using a one-stage object detector based on a deep convolutional neural network”. In: *PloS one* 13 (9 2018), e0203355. ISSN: 1932-6203 (cit. on p. 6).
- [46] Paul F Jaeger, Simon A A Kohl, Sebastian Bickelhaupt, Fabian Isensee, Tristan Anselm Kuder, Heinz-Peter Schlemmer, and Klaus H Maier-Hein. “Retina U-Net: Embarrassingly simple exploitation of segmentation supervision for medical object detection”. In: PMLR, 2020, pp. 171–183. ISBN: 2640-3498 (cit. on pp. 6, 7, 51, 53, 54, 60, 72, 77, 81, 83, 84, 95, 97, 102, 130).
- [47] Fabian Isensee, Paul F Jaeger, Simon A A Kohl, Jens Petersen, and Klaus H Maier-Hein. “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation”. In: *Nature methods* 18 (2 2021), pp. 203–211. ISSN: 1548-7105. URL: <https://www.nature.com/articles/s41592-020-01008-z> (cit. on pp. 6, 11, 56, 57, 66, 72, 73, 77–81, 84, 86, 93, 97, 114).
- [48] Lena Maier-Hein, Annika Reinke, Patrick Godau, Minu D Tizabi, Florian Buettner, Evangelia Christodoulou, Ben Glocker, Fabian Isensee, Jens Kleesiek, Michal Kozubek, et al. “Metrics reloaded: recommendations for image analysis validation”. In: *Nature methods* 21 (2 2024), pp. 195–212. ISSN: 1548-7091 (cit. on pp. 7, 11, 39, 41).
- [49] UXWing. *UXWing*. <https://uxwing.com/graphic-card-icon/>. Accessed: 2024-09-26. 2024 (cit. on pp. 7, 9).
- [50] Valery L Feigin, Michael Brainin, Bo Norrving, Sheila Martins, Ralph L Sacco, Werner Hacke, Marc Fisher, Jeyaraj Pandian, and Patrice Lindsay. “World Stroke Organization (WSO): global stroke fact sheet 2022”. In: *International Journal of Stroke* 17 (1 2022), pp. 18–29. ISSN: 1747-4930 (cit. on p. 8).
- [51] World Stroke Organization. *Impact of Stroke*. Accessed: 2024-11-07. 2024. URL: <https://www.world-stroke.org/world-stroke-day-campaign/about-stroke/impact-of-stroke> (cit. on p. 8).
- [52] Tasneem F Hasan, Hunaid Hasan, and Roger E Kelley. “Overview of acute ischemic stroke evaluation and management”. In: *Biomedicines* 9 (10 2021), p. 1486. ISSN: 2227-9059 (cit. on p. 8).
- [53] Nuno Vasconcelos and Zhaowei Cai. “Cascade R-CNN Delving into High Quality Object Detection”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2017 (cit. on pp. 9, 24).

- [54] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, and Wanli Ouyang. “Hybrid task cascade for instance segmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4974–4983 (cit. on pp. 9, 24).
- [55] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. “FCOS: fully convolutional one-stage object detection”. In: *Proceedings of the 2019 IEEE/Cvf International Conference on Computer Vision, Seoul, Korea*. 2019, pp. 27–28 (cit. on pp. 9, 31, 32, 39).
- [56] Hei Law and Jia Deng. “Cornersnet: Detecting objects as paired keypoints”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 734–750 (cit. on pp. 9, 39).
- [57] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. “Bottom-up object detection by grouping extreme and center points”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 850–859 (cit. on pp. 9, 39).
- [58] Yang Yang, Min Li, Bo Meng, Zihao Huang, Junxing Ren, and Degang Sun. “Objects as Extreme Points”. In: *PRICAI 2021: Trends in Artificial Intelligence: 18th Pacific Rim International Conference on Artificial Intelligence, PRICAI 2021, Hanoi, Vietnam, November 8–12, 2021, Proceedings, Part III 18*. Springer, 2021, pp. 195–208. ISBN: 3030893693 (cit. on p. 9).
- [59] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. “Dn-detr: Accelerate detr training by introducing query denoising”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 13619–13627 (cit. on pp. 9, 38, 54, 71, 72).
- [60] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. “DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR”. In: *International Conference on Learning Representations*. 2022. URL: <https://openreview.net/forum?id=oMI9PjOb9Jl> (cit. on pp. 9, 38, 54, 71, 72).
- [61] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. “Conditional detr for fast training convergence”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 3651–3660 (cit. on pp. 9, 38, 69–71).
- [62] Qiang Chen, Xiaokang Chen, Jian Wang, Shan Zhang, Kun Yao, Haocheng Feng, Junyu Han, Errui Ding, Gang Zeng, and Jingdong Wang. “Group detr: Fast detr training with group-wise one-to-many assignment”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 6633–6642 (cit. on pp. 9, 71).
- [63] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. “Anchor DETR: Query design for transformer-based object detection”. In: *arXiv preprint arXiv:2109.07107* 3 (6 2021) (cit. on pp. 9, 71).

-
- [64] Haichao Cao, Hong Liu, Enmin Song, Guangzhi Ma, Xiangyang Xu, Renchao Jin, Tengying Liu, and Chih-Cheng Hung. “A two-stage convolutional neural networks for lung nodule detection”. In: *IEEE journal of biomedical and health informatics* 24 (7 2020), pp. 2006–2015. ISSN: 2168-2194 (cit. on pp. 10, 51, 52, 97, 121, 132, 170).
 - [65] Qi Dou, Hao Chen, Yueming Jin, Huangjing Lin, Jing Qin, and Pheng-Ann Heng. “Automated pulmonary nodule detection via 3d convnets with online sample filtering and hybrid-loss residual learning”. In: *Medical Image Computing and Computer Assisted Intervention MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11–13, 2017, Proceedings, Part III* 20. Springer, 2017, pp. 630–638. ISBN: 3319661787 (cit. on pp. 10, 50, 51, 121, 132, 170).
 - [66] J Ding, A Li, Z Hu, and L Wang. *Accurate Pulmonary Nodule Detection in Computed Tomography Images Using Deep Convolutional Neural Networks*. *arXiv e-prints*. 2017 (cit. on pp. 10, 51, 121, 132, 170).
 - [67] Bin Wang, Guojun Qi, Sheng Tang, Liheng Zhang, Lixi Deng, and Yongdong Zhang. “Automated pulmonary nodule detection: High sensitivity with few candidates”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 759–767 (cit. on pp. 10, 51, 52, 121, 132, 170).
 - [68] Xiaoxi Lu, Na Zeng, Xingyue Wang, Jingqi Huang, Yan Hu, Jiansheng Fang, and Jiang Liu. “Ffnet: an end-to-end framework based on feature pyramid network and filter network for pulmonary nodule detection”. In: *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2023, pp. 1–6. ISBN: 1665473584 (cit. on pp. 10, 51, 54, 121, 132, 170).
 - [69] Xiangde Luo, Tao Song, Guotai Wang, Jieneng Chen, Yinan Chen, Kang Li, Dimitris N Metaxas, and Shaoting Zhang. “SCPM-Net: An anchor-free 3D lung nodule detection network using sphere representation and center points matching”. In: *Medical image analysis* 75 (2022), p. 102287. ISSN: 1361-8415 (cit. on pp. 10, 51, 53, 55, 121, 123, 132, 170).
 - [70] Jie Mei, Ming-Ming Cheng, Gang Xu, Lan-Ruo Wan, and Huan Zhang. “SANet: A slice-aware network for pulmonary nodule detection”. In: *IEEE transactions on pattern analysis and machine intelligence* 44 (8 2021), pp. 4374–4387. ISSN: 0162-8828 (cit. on pp. 10, 51, 53–55, 121–123, 131, 132, 170, 171).
 - [71] Rui Xu, Yong Luo, Bo Du, Kaiming Kuang, and Jiancheng Yang. “LSSANet: a long short slice-aware network for pulmonary nodule detection”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2022, pp. 664–674 (cit. on pp. 10, 51, 54, 55, 122, 131, 132, 171).

- [72] Yuemeng Li and Yang Fan. “DeepSEED: 3D squeeze-and-excitation encoder-decoder convolutional neural networks for pulmonary nodule detection”. In: *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 1866–1869. ISBN: 1538693305 (cit. on pp. 10, 51, 53, 121, 122, 131, 132, 170, 171).
- [73] Jingya Liu, Liangliang Cao, Oguz Akin, and Yingli Tian. “3DFPN-HS: 3D Feature Pyramid Network Based High Sensitivity and Specificity Pulmonary Nodule Detection”. In: *MICCAI*. Springer, 2019, pp. 513–521 (cit. on pp. 10, 51, 52, 121, 132).
- [74] Tao Song, Jieneng Chen, Xiangde Luo, Yechong Huang, Xinglong Liu, Ning Huang, Yinan Chen, Zhaoxiang Ye, Huaqiang Sheng, and Shaoting Zhang. “CPM-Net: A 3D center-points matching network for pulmonary nodule detection in CT scans”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 550–559. ISBN: 3030597253 (cit. on pp. 10, 51, 53, 121, 131, 132, 170, 171).
- [75] Wentao Zhu, Chaochun Liu, Wei Fan, and Xiaohui Xie. “Deeplung: Deep 3d dual path nets for automated pulmonary nodule detection and classification”. In: *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 673–681. ISBN: 1538648865. URL: <https://arxiv.org/abs/1801.09555> (cit. on pp. 10, 51, 121, 122, 131, 132, 170, 171).
- [76] Naji Khosravan and Ulas Bagci. “S4ND: Single-shot single-scale lung nodule detection”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*. Springer, 2018, pp. 794–802. ISBN: 3030009335 (cit. on pp. 10, 51, 52, 121, 132, 170).
- [77] Hao Tang, Chupeng Zhang, and Xiaohui Xie. “Nodulenet: Decoupled false positive reduction for pulmonary nodule detection and segmentation”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part VI 22*. Springer, 2019, pp. 266–274. ISBN: 3030322254 (cit. on pp. 10, 51, 52, 121, 122, 132, 170, 171).
- [78] Fangzhou Liao, Ming Liang, Zhe Li, Xiaolin Hu, and Sen Song. “Evaluate the malignancy of pulmonary nodules using the 3-d deep leaky noisy-or network”. In: *IEEE transactions on neural networks and learning systems* 30 (11 2019), pp. 3484–3495. ISSN: 2162-237X. URL: <https://pubmed.ncbi.nlm.nih.gov/30794190/> (cit. on pp. 10, 51, 52, 121, 122, 132, 170, 171).
- [79] Ivan William Harsono, Suryadiputra Liawatimena, and Tjeng Wawan Cenggoro. “Lung nodule detection and classification from Thorax CT-scan using RetinaNet with transfer learning”. In: *Journal of King Saud University-Computer and Information Sciences* 34 (3 2022), pp. 567–577. ISSN: 1319-1578 (cit. on pp. 10, 51, 54, 121, 122, 170, 171).

-
- [80] Martin Zlocha, Qi Dou, and Ben Glocker. “Improving RetinaNet for CT lesion detection with dense masks from weak RECIST labels”. In: *MICCAI*. Springer, 2019, pp. 402–410 (cit. on pp. 10, 58).
- [81] Fabian Isensee, Tassilo Wald, Constantin Ulrich, Michael Baumgartner, Saikat Roy, Klaus Maier-Hein, and Paul F. Jäger. “nnU-Net Revisited: A Call for Rigorous Validation in 3D Medical Image Segmentation”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*. Ed. by Marius George Linguraru, Qi Dou, Aasa Feragen, Stamatia Giannarou, Ben Glocker, Karim Lekadir, and Julia A. Schnabel. Cham: Springer Nature Switzerland, 2024, pp. 488–498. ISBN: 978-3-031-72114-4 (cit. on pp. 11, 131).
- [82] Michael Baumgartner, Marc K. Ickler, Paul F. Jäger, Fabian Isensee, Constantin Ulrich, Tassilo Wald, Julius Holzschuh, Balint Kovacs, Partha Ghosh, for the ALFA study, and Klaus H. Maier-Hein. “nnDetection: A Self-configuring Method for Volumetric 3D Object Detection”. In: *In preparation* (2025) (cit. on pp. 13, 57, 75, 85, 87–89, 91, 92, 96, 98, 100, 116, 118–120, 122–125, 152–171, 173–194).
- [83] Alberto M Ceballos-Arroyo, Hieu T Nguyen, Fangrui Zhu, Shrikanth M Yadav, Jisoo Kim, Lei Qin, Geoffrey Young, and Huaizu Jiang. “Vessel-Aware Aneurysm Detection Using Multi-scale Deformable 3D Attention”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2024, pp. 754–765 (cit. on pp. 14, 50, 76, 115, 123, 124, 132, 171, 173).
- [84] Mark Everingham, Luc Van Gool, Christopher K I Williams, John Winn, and Andrew Zisserman. “The pascal visual object classes (voc) challenge”. In: *International journal of computer vision* 88 (2010), pp. 303–338. ISSN: 0920-5691 (cit. on pp. 17, 23, 26).
- [85] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. “Microsoft coco: Common objects in context”. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755. ISBN: 3319106015 (cit. on pp. 17, 23, 26, 28, 36, 42, 54, 71).
- [86] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Mallocci, and Alexander Kolesnikov. “The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale”. In: *International journal of computer vision* 128 (7 2020), pp. 1956–1981. ISSN: 0920-5691 (cit. on p. 17).
- [87] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. “Objects365: A large-scale, high-quality dataset for object detection”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 8430–8439 (cit. on pp. 17, 30).

- [88] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. “The cityscapes dataset for semantic urban scene understanding”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223 (cit. on p. 17).
- [89] Agrim Gupta, Piotr Dollar, and Ross Girshick. “Lvis: A dataset for large vocabulary instance segmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 5356–5364 (cit. on p. 17).
- [90] Paul Viola and Michael Jones. “Rapid object detection using a boosted cascade of simple features”. In: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. Vol. 1. Ieee, 2001, pp. I–I. ISBN: 0769512720 (cit. on p. 17).
- [91] Navneet Dalal and Bill Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*. Vol. 1. Ieee, 2005, pp. 886–893. ISBN: 0769523722 (cit. on p. 17).
- [92] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012) (cit. on pp. 17–19).
- [93] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014) (cit. on pp. 17, 27).
- [94] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255. ISBN: 1424439922 (cit. on pp. 17–19, 25, 54).
- [95] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: 2014, pp. 580–587. URL: <https://arxiv.org/abs/1311.2524> (cit. on pp. 18, 27, 86).
- [96] Jasper R R Uijlings, Koen E A Van De Sande, Theo Gevers, and Arnold W M Smeulders. “Selective search for object recognition”. In: *International journal of computer vision* 104 (2013), pp. 154–171. ISSN: 0920-5691 (cit. on pp. 18, 21).
- [97] Corinna Cortes. “Support-Vector Networks”. In: *Machine Learning* (1995) (cit. on p. 18).
- [98] Pedro F Felzenszwalb and Daniel P Huttenlocher. “Efficient graph-based image segmentation”. In: *International journal of computer vision* 59 (2004), pp. 167–181. ISSN: 0920-5691 (cit. on p. 18).

-
- [99] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. “Object detection with discriminatively trained part-based models”. In: *IEEE transactions on pattern analysis and machine intelligence* 32 (9 2009), pp. 1627–1645. ISSN: 0162-8828 (cit. on p. 19).
- [100] K-K Sung and Tomaso Poggio. “Example-based learning for view-based human face detection”. In: *IEEE Transactions on pattern analysis and machine intelligence* 20 (1 1998), pp. 39–51. ISSN: 0162-8828 (cit. on p. 19).
- [101] Ross Girshick. “Fast r-cnn”. In: 2015, pp. 1440–1448. URL: <https://arxiv.org/abs/1504.08083> (cit. on p. 20, 21, 23, 26, 28, 65, 86).
- [102] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Spatial pyramid pooling in deep convolutional networks for visual recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 37 (9 2015), pp. 1904–1916. ISSN: 0162-8828 (cit. on p. 20).
- [103] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. “Deconvolutional networks”. In: *2010 IEEE Computer Society Conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2528–2535. ISBN: 1424469856 (cit. on pp. 23, 60).
- [104] Yue Wu, Yinpeng Chen, Lu Yuan, Zicheng Liu, Lijuan Wang, Hongzhi Li, and Yun Fu. “Rethinking classification and localization for object detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10186–10195 (cit. on p. 24).
- [105] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. “Libra r-cnn: Towards balanced learning for object detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 821–830 (cit. on p. 24).
- [106] Hongkai Zhang, Hong Chang, Bingpeng Ma, Naiyan Wang, and Xilin Chen. “Dynamic R-CNN: Towards high quality object detection via dynamic training”. In: *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV* 16. Springer, 2020, pp. 260–275. ISBN: 3030585549 (cit. on p. 24).
- [107] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. “Probabilistic two-stage detection”. In: *arXiv preprint arXiv:2103.07461* (2021). URL: <https://arxiv.org/abs/2103.07461> (cit. on p. 24).
- [108] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, and Changhu Wang. “Sparse r-cnn: End-to-end object detection with learnable proposals”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 14454–14463 (cit. on p. 24).

- [109] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. “Convnext v2: Co-designing and scaling convnets with masked autoencoders”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 16133–16142 (cit. on pp. 24, 30, 140).
- [110] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. “Exploring plain vision transformer backbones for object detection”. In: *European conference on computer vision*. Springer, 2022, pp. 280–296 (cit. on pp. 24, 31).
- [111] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. “Eva: Exploring the limits of masked visual representation learning at scale”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 19358–19369 (cit. on pp. 24, 30).
- [112] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, and Subhojit Som. “Image as a foreign language: Beit pretraining for all vision and vision-language tasks”. In: *arXiv preprint arXiv:2208.10442* (2022) (cit. on pp. 24, 30).
- [113] J Redmon. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016 (cit. on pp. 25, 26, 31).
- [114] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9 (cit. on p. 25).
- [115] Joseph Redmon and Ali Farhadi. “YOLO9000: better, faster, stronger”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7263–7271 (cit. on p. 26).
- [116] Sergey Ioffe. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *arXiv preprint arXiv:1502.03167* (2015) (cit. on p. 26).
- [117] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. “Feature pyramid networks for object detection”. In: 2017, pp. 2117–2125. URL: <https://arxiv.org/abs/1612.03144> (cit. on pp. 26, 60, 65).
- [118] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. “Ssd: Single shot multibox detector”. In: *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37. ISBN: 3319464477 (cit. on pp. 26–28, 122, 171).

-
- [119] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. “Training region-based object detectors with online hard example mining”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 761–769 (cit. on p. 28).
- [120] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. “Yolov4: Optimal speed and accuracy of object detection”. In: *arXiv preprint arXiv:2004.10934* (2020) (cit. on p. 30).
- [121] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, and Weiqiang Nie. “YOLOv6: A single-stage object detection framework for industrial applications”. In: *arXiv preprint arXiv:2209.02976* (2022) (cit. on p. 30).
- [122] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 7464–7475 (cit. on p. 30).
- [123] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. “Yolov9: Learning what you want to learn using programmable gradient information”. In: *arXiv preprint arXiv:2402.13616* (2024) (cit. on p. 30).
- [124] Z Ge. “Yolox: Exceeding yolo series in 2021”. In: *arXiv preprint arXiv:2107.08430* (2021) (cit. on p. 30).
- [125] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. “ImageNet-21K Pretraining for the Masses”. In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*. 2021. URL: https://openreview.net/forum?id=Zkj_VcZ6o1 (cit. on p. 30).
- [126] Mingxing Tan, Ruoming Pang, and Quoc V Le. “Efficientdet: Scalable and efficient object detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10781–10790 (cit. on pp. 30, 31).
- [127] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, and Hongsheng Li. “Internimage: Exploring large-scale vision foundation models with deformable convolutions”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 14408–14419 (cit. on p. 30).
- [128] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. “Masked autoencoders are scalable vision learners”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 16000–16009. URL: <https://arxiv.org/abs/2111.06377> (cit. on pp. 30, 140).

- [129] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, and Chunyuan Li. “Florence: A new foundation model for computer vision”. In: *arXiv preprint arXiv:2111.11432* (2021) (cit. on pp. 30, 31).
- [130] Weijie Su, Xizhou Zhu, Chenxin Tao, Lewei Lu, Bin Li, Gao Huang, Yu Qiao, Xiaogang Wang, Jie Zhou, and Jifeng Dai. “Towards all-in-one pre-training via maximizing multi-modal mutual information”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 15888–15899 (cit. on p. 30).
- [131] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. “Path aggregation network for instance segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8759–8768 (cit. on p. 31).
- [132] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. “Nas-fpn: Learning scalable feature pyramid architecture for object detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 7036–7045 (cit. on p. 31).
- [133] Kai Chen, Yuhang Cao, Chen Change Loy, Dahua Lin, and Christoph Feichtenhofer. “Feature pyramid grids”. In: *arXiv preprint arXiv:2004.03580* (2020) (cit. on p. 31).
- [134] Xianzhi Du, Barret Zoph, Wei-Chih Hung, and Tsung-Yi Lin. “Simple training strategies and model scaling for object detection”. In: *arXiv preprint arXiv:2107.00057* (2021) (cit. on p. 31).
- [135] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. “Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21002–21012 (cit. on p. 31).
- [136] Xiang Li, Wenhai Wang, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. “Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 11632–11641 (cit. on p. 31).
- [137] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. “Iou loss for 2d/3d object detection”. In: *2019 international conference on 3D vision (3DV)*. IEEE, 2019, pp. 85–94. ISBN: 1728131316 (cit. on p. 31).
- [138] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. “Generalized intersection over union: A metric and a loss for bounding box regression”. In: 2019, pp. 658–666. URL: <https://arxiv.org/abs/1902.09630> (cit. on pp. 31, 90).

-
- [139] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. “Distance-IoU loss: Faster and better learning for bounding box regression”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 2020, pp. 12993–13000. ISBN: 2374-3468 (cit. on p. 31).
- [140] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. “Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection”. In: 2020, pp. 9759–9768. URL: <https://arxiv.org/abs/1912.02424> (cit. on pp. 31, 39).
- [141] Kang Kim and Hee Seok Lee. “Probabilistic anchor assignment with iou prediction for object detection”. In: *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV* 16. Springer, 2020, pp. 355–371. ISBN: 3030585948 (cit. on p. 31).
- [142] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. “Ota: Optimal transport assignment for object detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 303–312 (cit. on p. 31).
- [143] Jianfeng Wang, Lin Song, Zeming Li, Hongbin Sun, Jian Sun, and Nanning Zheng. “End-to-end object detection with fully convolutional network”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 15849–15858 (cit. on p. 31).
- [144] Shuai Li, Minghan Li, Ruihuang Li, Chenhang He, and Lei Zhang. “One-to-few label assignment for end-to-end dense detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 7350–7359 (cit. on p. 31).
- [145] Roy Jonker and Ton Volgenant. “A shortest augmenting path algorithm for dense and sparse linear assignment problems”. In: *DGOR/NSOR: Papers of the 16th Annual Meeting of DGOR in Cooperation with NSOR/Vorträge der 16. Jahrestagung der DGOR zusammen mit der NSOR*. Springer, 1988, p. 622. ISBN: 3540193650 (cit. on p. 33).
- [146] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017). URL: <https://arxiv.org/abs/1706.03762> (cit. on p. 34).
- [147] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. “Deformable {DETR}: Deformable Transformers for End-to-End Object Detection”. In: 2021. URL: <https://openreview.net/forum?id=gZ9hCDWe6ke%20https://arxiv.org/abs/2010.04159> (cit. on pp. 36, 37, 54, 71, 72, 77, 90, 130).

- [148] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. “Pix2seq: A language modeling framework for object detection”. In: *arXiv preprint arXiv:2109.10852* (2021) (cit. on p. 39).
- [149] Annika Reinke, Minu D Tizabi, Michael Baumgartner, Matthias Eisenmann, Doreen Heckmann-Nötzel, A Emre Kavur, Tim Rädtsch, Carole H Sudre, Laura Acion, Michela Antonelli, et al. “Understanding metric-related pitfalls in image analysis validation”. In: *Nature methods* 21.2 (2024), pp. 182–194 (cit. on pp. 40, 74, 114).
- [150] BACM Fasen, R J J Heijboer, F-JH Hulsmans, and R M Kwee. “CT angiography in evaluating large-vessel occlusion in acute anterior circulation ischemic stroke: factors associated with diagnostic error in clinical practice”. In: *American Journal of Neuroradiology* 41 (4 2020), pp. 607–611. ISSN: 0195-6108 (cit. on p. 46).
- [151] Shalini A Amukotuwa, Matus Straka, Seena Dehkharghani, and Roland Bammer. “Fast automatic detection of large vessel occlusions on CT angiography”. In: *Stroke* 50 (12 2019), pp. 3431–3438. ISSN: 0039-2499 (cit. on pp. 46, 49, 128).
- [152] Shalini A Amukotuwa, Matus Straka, Heather Smith, Ronil V Chandra, Seena Dehkharghani, Nancy J Fischbein, and Roland Bammer. “Automated detection of intracranial large vessel occlusions on computed tomography angiography: a single center experience”. In: *Stroke* 50 (10 2019), pp. 2790–2798. ISSN: 0039-2499 (cit. on pp. 46, 49).
- [153] Arko Barman, Mehmet E Inam, Songmi Lee, Sean Savitz, Sunil Sheth, and Luca Giancardo. “Determining ischemic stroke from CT-angiography imaging using symmetry-sensitive convolutional networks”. In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019, pp. 1873–1877. ISBN: 1538636417 (cit. on pp. 46, 49).
- [154] Matthew T Stib, Justin Vasquez, Mary P Dong, Yun Ho Kim, Sumera S Subzwari, Harold J Triedman, Amy Wang, Hsin-Lei Charlene Wang, Anthony D Yao, and Mahesh Jayaraman. “Detecting large vessel occlusion at multiphase CT angiography by using a deep convolutional neural network”. In: *Radiology* 297 (3 2020), pp. 640–649. ISSN: 0033-8419 (cit. on pp. 46, 49, 128).
- [155] Florian Thamm, Markus Jürgens, Hendrik Ditt, and Andreas Maier. “VirtualDSA++-Automated Segmentation, Vessel Labeling, Occlusion Detection, and Graph Search on CT Angiography Data.” In: *VCBM*. 2020, pp. 151–155 (cit. on pp. 47, 49).
- [156] Marta Olive-Gadea, Carlos Crespo, Cristina Granes, Maria Hernandez-Perez, Natalia Perez de la Ossa, Carlos Laredo, Xabier Urra, Juan Carlos Soler, Alexander Soler, and Paloma Puyalto. “Deep learning based software to identify large vessel occlusion on noncontrast computed tomography”. In: *Stroke* 51 (10 2020), pp. 3133–3137. ISSN: 0039-2499 (cit. on pp. 47, 49).

-
- [157] Seena Dehkharghani, Maarten Lansberg, Chitra Venkatsubramanian, Carlo Cereda, Fabricio Lima, Henrique Coelho, Felipe Rocha, Abid Qureshi, Hafez Haerian, and Francisco Mont’Alverne. “High-performance automated anterior circulation CT angiographic clot detection in acute stroke: a multireader comparison”. In: *Radiology* 298 (3 2021), pp. 665–670. ISSN: 0033-8419 (cit. on pp. 47, 49, 128).
- [158] A Yahav-Dovrat, M Saban, G Merhav, I Lankri, E Abergel, A Eran, D Tanne, R G Nogueira, and R Sivan-Hoffmann. “Evaluation of artificial intelligence–powered identification of large-vessel occlusions in a comprehensive stroke center”. In: *American Journal of Neuroradiology* 42 (2 2021), pp. 247–254. ISSN: 0195-6108 (cit. on pp. 47, 49, 128).
- [159] Ryan A Rava, Blake A Peterson, Samantha E Seymour, Kenneth V Snyder, Maxim Mokin, Muhammad Waqas, Yiemeng Hoi, Jason M Davies, Elad I Levy, and Adnan H Siddiqui. “Validation of an artificial intelligence-driven large vessel occlusion detection algorithm for acute ischemic stroke patients”. In: *The Neuroradiology Journal* 34 (5 2021), pp. 408–417. ISSN: 1971-4009 (cit. on pp. 47, 49, 128).
- [160] Dan Paz, Daniel Yagoda, and Theodore Wein. “Single site performance of AI software for stroke detection and triage”. In: *medRxiv* (2021), pp. 2021–2024 (cit. on pp. 47, 49).
- [161] Sven P R Luijten, Lennard Wolff, Martijne H C Duvekot, Pieter-Jan van Doormaal, Walid Moudrous, Henk Kerkhoff, Geert J Lycklama a Nijeholt, Reinoud P H Bokkers, S F Lonneke, and Jeannette Hofmeijer. “Diagnostic performance of an algorithm for automated large vessel occlusion detection on CT angiography”. In: *Journal of neurointerventional surgery* 14 (8 2022), pp. 794–798. ISSN: 1759-8478 (cit. on pp. 47, 49, 128).
- [162] Fatih Seker, Johannes Alex Rolf Pfaff, Yahia Mokli, Anne Berberich, Rafael Namias, Steven Gerry, Simon Nagel, Martin Bendszus, and Christian Herweh. “Diagnostic accuracy of automated occlusion detection in CT angiography using e-CTA”. In: *International Journal of Stroke* 17 (1 2022), pp. 77–82. ISSN: 1747-4930 (cit. on pp. 47, 49, 128).
- [163] Florian Thamm, Oliver Taubmann, Markus Jürgens, Hendrik Ditt, and Andreas Maier. “Detection of large vessel occlusions using deep learning by deforming vessel tree segmentations”. In: *Bildverarbeitung für die Medizin 2022: Proceedings, German Workshop on Medical Image Computing, Heidelberg, June 26-28, 2022*. Springer, 2022, pp. 44–49 (cit. on pp. 47–49).
- [164] Florian Thamm, Oliver Taubmann, Markus Jürgens, Aleksandra Thamm, Felix Denzinger, Leonhard Rist, Hendrik Ditt, and Andreas Maier. “Building brains: Sub-volume recombination for data augmentation in large vessel occlusion detection”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2022, pp. 634–643 (cit. on pp. 48, 49).

- [165] Jad Kassam, Florian Thamm, Leonhard Rist, Oliver Taubmann, and Andreas Maier. “Detecting Large Vessel Occlusions using Graph Deep Learning”. In: *Geometric Deep Learning in Medical Image Analysis*. PMLR, 2022, pp. 149–159. ISBN: 2640-3498 (cit. on pp. 48, 49).
- [166] Sunil A Sheth, Victor Lopez-Rivera, Arko Barman, James C Grotta, Albert J Yoo, Songmi Lee, Mehmet E Inam, Sean I Savitz, and Luca Giancardo. “Machine learning-enabled automated determination of acute ischemic core from computed tomography angiography”. In: *Stroke* 50 (11 2019), pp. 3093–3100. ISSN: 0039-2499 (cit. on p. 49).
- [167] Alexander Keedy. “An overview of intracranial aneurysms”. In: *McGill Journal of Medicine: MjM* 9 (2 2006), p. 141 (cit. on p. 48).
- [168] Dennis J Nieuwkamp, Larissa E Setz, Ale Algra, Francisca H H Linn, Nicolien K de Rooij, and Gabriël J E Rinkel. “Changes in case fatality of aneurysmal subarachnoid haemorrhage over time, according to age, sex, and region: a meta-analysis”. In: *The Lancet Neurology* 8 (7 2009), pp. 635–642. ISSN: 1474-4422 (cit. on p. 48).
- [169] Youssef Assis, Liang Liao, Fabien Pierre, René Anxionnat, and Erwan Kerrien. “Intracranial aneurysm detection: an object detection perspective”. In: *International Journal of Computer Assisted Radiology and Surgery* (2024), pp. 1–9. ISSN: 1861-6429 (cit. on pp. 49, 132).
- [170] Zi-Hao Bo, Hui Qiao, Chong Tian, Yuchen Guo, Wuchao Li, Tiantian Liang, Dongxue Li, Dan Liao, Xianchun Zeng, Leilei Mei, et al. “Toward human intervention-free clinical diagnosis of intracranial aneurysm via deep neural network”. In: *Patterns* 2.2 (2021), p. 100197 (cit. on pp. 50, 76, 114, 115, 173).
- [171] World Health Organization. *Lung Cancer Fact Sheet*. Accessed: 2024-11-06. 2020. URL: <https://www.who.int/news-room/fact-sheets/detail/lung-cancer> (cit. on p. 50).
- [172] The National Lung Screening Trial Research Team. “Reduced Lung-Cancer Mortality with Low-Dose Computed Tomographic Screening”. In: *New England Journal of Medicine* 365 (5 2011), pp. 395–409. DOI: 10.1056/NEJMoa1102873. URL: <https://www.nejm.org/doi/full/10.1056/NEJMoa1102873> (cit. on p. 50).
- [173] Bram Van Ginneken, Samuel G Armato III, Bartjan de Hoop, Saskia van Amelsvoort-van de Vorst, Thomas Duindam, Meindert Niemeijer, Keelin Murphy, Arnold Schilham, Alessandra Retico, and Maria Evelina Fantacci. “Comparing and combining algorithms for computer-aided detection of pulmonary nodules in computed tomography scans: the ANODE09 study”. In: *Medical image analysis* 14 (6 2010), pp. 707–722. ISSN: 1361-8415 (cit. on pp. 50, 102).

-
- [174] Zehui Gong, Dong Li, Jiatai Lin, Yun Zhang, and Kin-Man Lam. “Towards accurate pulmonary nodule detection by representing nodules as points with high-resolution network”. In: *IEEE access* 8 (2020), pp. 157391–157402. ISSN: 2169-3536 (cit. on pp. 51, 53, 121, 132, 170).
 - [175] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. “Dual path networks”. In: *Advances in neural information processing systems* 30 (2017) (cit. on pp. 51, 131).
 - [176] AJ Buckeye, Jacob Kriss, Josette BoozAllen, Josh Sullivan, Meghan O’Connell, Nilofer, and Will Cukierski. *Data Science Bowl 2017*. <https://kaggle.com/competitions/data-science-bowl-2017>. Kaggle. 2017 (cit. on p. 52).
 - [177] Jie Hu, Li Shen, and Gang Sun. “Squeeze-and-excitation networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141 (cit. on pp. 52, 53, 131).
 - [178] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. “Inception-v4, inception-resnet and the impact of residual connections on learning”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 2017. ISBN: 2374-3468 (cit. on p. 53).
 - [179] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. “Objects as points”. In: *arXiv preprint arXiv:1904.07850* (2019) (cit. on p. 53).
 - [180] Bastian Wittmann, Fernando Navarro, Suprosanna Shit, and Bjoern Menze. “Focused decoding enables 3D anatomical detection by transformers”. In: *arXiv preprint arXiv:2207.10774* (2022) (cit. on p. 55).
 - [181] Nicholas Heller, Niranjana Sathianathan, Arveen Kalapara, Edward Walczak, Keenan Moore, Heather Kaluzniak, Joel Rosenberg, Paul Blake, Zachary Rengel, Makinna Oestreich, et al. “The kits19 challenge data: 300 kidney tumor cases with clinical context, ct semantic segmentations, and surgical outcomes”. In: *arXiv preprint arXiv:1904.00445* (2019) (cit. on pp. 55, 68).
 - [182] Michael Baumgartner, Peter M Full, and Klaus H Maier-Hein. “Accurate Detection of Mediastinal Lesions with nnDetection”. In: Springer, 2022, pp. 79–85 (cit. on pp. 57, 61, 62, 100, 101, 125, 127, 131, 138).
 - [183] Gianluca Brugnara, Michael Baumgartner, Edwin David Scholze, Katerina Deike-Hofmann, Klaus Kades, Jonas Scherer, Stefan Denner, Hagen Meredig, Aditya Rastogi, Mustafa Ahmed Mahmutoglu, Christian Ulfert, Ulf Neuberger, Silvia Schönenberger, Kai Schlamp, Zeynep Bendella, Thomas Pinetz, Carsten Schmeel, Wolfgang Wick, Peter A Ringleb, Ralf Floca, Markus Möhlenbruch, Alexander Radbruch, Martin Bendszus, Klaus Maier-Hein, and Philipp Vollmuth. “Deep-learning based detection of vessel occlusions on CT-angiography in patients with suspected acute ischemic stroke”. In: *Nature Communications* 14 (1 2023), p. 4938. ISSN: 2041-1723. DOI: 10.1038/s41467-023-40564-8. URL: <https://doi.org/10.1038/s41467-023-40564-8>.

- //doi.org/10.1038/s41467-023-40564-8 (cit. on pp. 57, 64–66, 100, 104, 106–110, 125, 148–150).
- [184] Marc K Ickler, Michael Baumgartner, Saikat Roy, Tassilo Wald, and Klaus H Maier-Hein. “Taming Detection Transformers for Medical Object Detection”. In: *BVM Workshop*. Springer, 2023, pp. 183–188 (cit. on pp. 57, 70, 72, 100, 125).
 - [185] Michael Baumgartner, Paul F Jäger, Fabian Isensee, and Klaus H Maier-Hein. “nnDetection: a self-configuring method for medical object detection”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part V* 24. Springer, 2021, pp. 530–539. ISBN: 3030872394 (cit. on pp. 57, 69, 100, 102, 111, 125, 129).
 - [186] You-Bao Tang, Ke Yan, Yu-Xing Tang, Jiamin Liu, Jin Xiao, and Ronald M Summers. “ULDor: a universal lesion detector for CT scans with pseudo masks and hard negative example mining”. In: *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*. IEEE, 2019, pp. 833–836. ISBN: 1538636417 (cit. on p. 58).
 - [187] Fabian Isensee, Paul Jäger, Jakob Wasserthal, David Zimmerer, Jens Petersen, Simon Kohl, Justus Schock, Andre Klein, Tobias Roß, Sebastian Wirkert, Peter Neher, Stefan Dinkelacker, Gregor Köhler, and Klaus Maier-Hein. *batchgenerators - a python framework for data augmentation*. Jan. 2020. DOI: 10.5281/zenodo.3632567. URL: <https://doi.org/10.5281/zenodo.3632567> (cit. on pp. 58, 62).
 - [188] Ivo Wolf, Marcus Vetter, Ingmar Wegner, Thomas Böttger, Marco Nolden, Max Schöbinger, Mark Hastenteufel, Tobias Kunert, and Hans-Peter Meinzer. “The medical imaging interaction toolkit”. In: *Medical image analysis* 9.6 (2005), pp. 594–604 (cit. on p. 59).
 - [189] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. “Instance normalization: The missing ingredient for fast stylization”. In: *arXiv preprint arXiv:1607.08022* (2016) (cit. on pp. 60, 64, 69, 84, 90).
 - [190] Yuxin Wu and Kaiming He. “Group normalization”. In: 2018, pp. 3–19. URL: <https://arxiv.org/abs/1803.08494> (cit. on pp. 60, 84, 86).
 - [191] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. “V-net: Fully convolutional neural networks for volumetric medical image segmentation”. In: *2016 fourth international conference on 3D vision (3DV)*. Ieee, 2016, pp. 565–571. ISBN: 1509054073 (cit. on p. 60).

-
- [192] Muhammad Waqas, Ansaar T Rai, Kunal Vakharia, Felix Chin, and Adnan H Siddiqui. “Effect of definition and methods on estimates of prevalence of large vessel occlusion in acute ischemic stroke: a systematic review and meta-analysis”. In: *Journal of neurointerventional surgery* 12 (3 2020), pp. 260–265. ISSN: 1759-8478 (cit. on p. 64).
 - [193] Johanna Maria Ospel and Mayank Goyal. “A review of endovascular treatment for medium vessel occlusion stroke”. In: *Journal of NeuroInterventional Surgery* 13 (7 2021), pp. 623–630. ISSN: 1759-8478 (cit. on p. 64).
 - [194] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. “Image transformer”. In: *International conference on machine learning*. PMLR, 2018, pp. 4055–4064. ISBN: 2640-3498 (cit. on p. 69).
 - [195] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. “Rethinking transformer-based set prediction for object detection”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 3611–3620 (cit. on p. 71).
 - [196] I Loshchilov. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017) (cit. on pp. 72, 90, 98).
 - [197] Anindo Saha, Joeran Bosma, Jasper Twilt, Bram van Ginneken, Derya Yakar, Mattijs Elschot, Jeroen Veltman, Jurgen Fütterer, and Maarten de Rooij. “Artificial intelligence and radiologists at prostate cancer detection in mri—the pi-cai challenge”. In: 2023. URL: <https://openreview.net/forum?id=XfXcA9-0XxR> (cit. on pp. 74, 172).
 - [198] Anindo Saha, Joeran S Bosma, Jasper J Twilt, Bram van Ginneken, Anders Bjartell, Anwar R Padhani, David Bonekamp, Geert Villeirs, Georg Salomon, and Gianluca Giannarini. “Artificial intelligence and radiologists in prostate cancer detection on MRI (PI-CAI): an international, paired, non-inferiority, confirmatory study”. In: *The Lancet Oncology* (2024). ISSN: 1470-2045 (cit. on pp. 74, 172).
 - [199] Yuting He, Guanyu Yang, Jian Yang, Rongjun Ge, Youyong Kong, Xiaomei Zhu, Shaobo Zhang, Pengfei Shao, Huazhong Shu, and Jean-Louis Dillenseger. “Meta grayscale adaptive network for 3D integrated renal structures segmentation”. In: *Medical image analysis* 71 (2021), p. 102055. ISSN: 1361-8415. URL: <https://www.sciencedirect.com/science/article/pii/S1361841521001018> (cit. on pp. 74, 172).
 - [200] Yuting He, Guanyu Yang, Jian Yang, Yang Chen, Youyong Kong, Jiasong Wu, Lijun Tang, Xiaomei Zhu, Jean-Louis Dillenseger, and Pengfei Shao. “Dense biased networks with deep priori anatomy and hard region adaptation: Semi-supervised learning for fine renal artery segmentation”. In: *Medical image analysis* 63 (2020), p. 101722. ISSN: 1361-8415. URL: <https://www.sciencedirect.com/science/article/pii/S1361841520300864> (cit. on pp. 74, 172).

- [201] Pengfei Shao, Chao Qin, Changjun Yin, Xiaoxin Meng, Xiaobing Ju, Jie Li, Qiang Lv, Wei Zhang, and Zhengquan Xu. “Laparoscopic partial nephrectomy with segmental renal artery clamping: technique and clinical outcomes”. In: *European urology* 59 (5 2011), pp. 849–855. ISSN: 0302-2838. URL: <https://www.sciencedirect.com/science/article/pii/S0302283810011395> (cit. on pp. 74, 172).
- [202] Pengfei Shao, Lijun Tang, Pu Li, Yi Xu, Chao Qin, Qiang Cao, Xiaobing Ju, Xiaoxin Meng, Qiang Lv, and Jie Li. “Precise segmental renal artery clamping under the guidance of dual-source computed tomography angiography during laparoscopic partial nephrectomy”. In: *European urology* 62 (6 2012), pp. 1001–1008. ISSN: 0302-2838. URL: <https://www.sciencedirect.com/science/article/pii/S0302283812006409> (cit. on pp. 74, 172).
- [203] Lorraine Abel, Jakob Wasserthal, Thomas Weikert, Alexander W Sauter, Ivan Nesic, Marko Obradovic, Shan Yang, Sebastian Manneck, Carl Glessgen, and Johanna M Ospel. “Automated detection of pancreatic cystic lesions on CT using deep learning”. In: *Diagnostics* 11 (5 2021), p. 901. ISSN: 2075-4418. URL: <https://www.mdpi.com/2075-4418/11/5/901> (cit. on pp. 74, 172).
- [204] Jeffrey D Rudie, Rachit Saluja, David A Weiss, Pierre Nedelec, Evan Calabrese, John B Colby, Benjamin Laguna, John Mongan, Steve Braunstein, and Christopher P Hess. “The University of California San Francisco Brain Metastases Stereotactic Radiosurgery (UCSF-BMSR) MRI Dataset”. In: *Radiology: Artificial Intelligence* 6 (2 2024), e230126. ISSN: 2638-6100 (cit. on pp. 74, 172).
- [205] Ahmed W Moawad, Anastasia Janas, Ujjwal Baid, Divya Ramakrishnan, Leon Jekel, Kiril Krantchev, Harrison Moy, Rachit Saluja, Klara Osenberg, and Klara Wilms. “The brain tumor segmentation (BraTS-METS) challenge 2023: Brain metastasis segmentation on pre-treatment MRI”. In: *ArXiv* (2023) (cit. on pp. 74, 172).
- [206] Natália Alves, Megan Schuurmans, Derya Yakar, Pierpaolo Vendittelli, Geert Litjens, John Hermans, and Henkjan Huisman. *The PANORAMA Challenge: Public Training and Development Dataset*. Apr. 2024. DOI: 10.5281/zenodo.11034178. URL: <https://doi.org/10.5281/zenodo.11034178> (cit. on pp. 74, 172).
- [207] João Pedrosa, Guilherme Aresta, Carlos Ferreira, Gurraj Atwal, Hady Ahmady Phoulady, Xiaoyu Chen, Rongzhen Chen, Jiaoliang Li, Liansheng Wang, and Adrian Galdran. “LNDb challenge on automatic lung cancer patient management”. In: *Medical image analysis* 70 (2021), p. 102027. ISSN: 1361-8415. URL: <https://www.sciencedirect.com/science/article/pii/S1361841521000736#tb10005> (cit. on pp. 74, 78, 172).
- [208] J. Rapin and O. Teytaud. *Nevergrad - A gradient-free optimization platform*. <https://GitHub.com/FacebookResearch/Nevergrad>. 2018 (cit. on p. 82).

-
- [209] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. “On the Convergence of Adam and Beyond”. In: *International Conference on Learning Representations*. 2018. URL: <https://openreview.net/forum?id=ryQu7f-RZ> (cit. on p. 90).
- [210] Fabian Isensee, Paul Jäger, Jakob Wasserthal, David Zimmerer, Jens Petersen, Simon Kohl, Justus Schock, Andre Klein, Tobias Roß, Sebastian Wirkert, Peter Neher, Stefan Dinkelacker, Gregor Köhler, and Klaus Maier-Hein. *batchgenerators - a python framework for data augmentation*. Jan. 2020. DOI: 10.5281/zenodo.3632567. URL: <https://doi.org/10.5281/zenodo.3632567> (cit. on p. 93).
- [211] Shuhao Wang, Dijia Wu, Lifang Ye, Zirong Chen, Yiqiang Zhan, and Yuehua Li. “Assessment of automatic rib fracture detection on chest CT using a deep learning algorithm”. In: *European Radiology* 33 (3 2023), pp. 1824–1834. ISSN: 1432-1084 (cit. on p. 97).
- [212] Na Wang, Cheng Bian, Yi Wang, Min Xu, Chenchen Qin, Xin Yang, Tianfu Wang, Anhua Li, Dinggang Shen, and Dong Ni. “Densely deep supervised networks with threshold loss for cancer detection in automated breast ultrasound”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part IV 11*. Springer, 2018, pp. 641–648. ISBN: 303000936X (cit. on p. 97).
- [213] Endre Grøvik, Darvin Yi, Michael Iv, Elizabeth Tong, Daniel Rubin, and Greg Zaharchuk. “Deep learning enables automatic detection and segmentation of brain metastases on multisequence MRI”. In: *Journal of Magnetic Resonance Imaging* 51 (1 2020), pp. 175–182. ISSN: 1053-1807 (cit. on p. 97).
- [214] Grzegorz Chlebus, Andrea Schenk, Jan Hendrik Moltz, Bram van Ginneken, Horst Karl Hahn, and Hans Meine. “Automatic liver tumor segmentation in CT with fully convolutional neural networks and object-based postprocessing”. In: *Scientific reports* 8 (1 2018), p. 15497. ISSN: 2045-2322 (cit. on p. 97).
- [215] Anindo Saha, Matin Hosseinzadeh, and Henkjan Huisman. “End-to-end prostate cancer detection in bpMRI via 3D CNNs: effects of attention mechanisms, clinical priori and decoupled false positive reduction”. In: *Medical image analysis* 73 (2021), p. 102155. ISSN: 1361-8415 (cit. on p. 97).
- [216] Anindo Saha, Joeran Bosma, Jasper Linmans, Matin Hosseinzadeh, and Henkjan Huisman. “Anatomical and Diagnostic Bayesian Segmentation in Prostate MRI – Should Different Clinical Objectives Mandate Different Loss Functions?” In: *arXiv preprint arXiv:2110.12889* (2021) (cit. on p. 97).
- [217] Nils Netzer, Carolin Eith, Oliver Bethge, Thomas Hielscher, Constantin Schwab, Albrecht Stenzinger, Regula Gnirs, Heinz-Peter Schlemmer, Klaus H Maier-Hein, and Lars Schimmöller. “Application of a validated prostate MRI deep learning system to independent same-vendor multi-institutional data: demonstration of

- transferability”. In: *European Radiology* 33 (11 2023), pp. 7463–7476. ISSN: 1432-1084 (cit. on p. 97).
- [218] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, and Wan-Yen Lo. “Segment anything”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 4015–4026 (cit. on pp. 117, 139).
 - [219] Yiming Xie, Huaizu Jiang, Georgia Gkioxari, and Julian Straub. “Pixel-Aligned Recurrent Queries for Multi-View 3D Object Detection”. In: *ICCV*. 2023 (cit. on pp. 123, 171).
 - [220] Nick M Murray, Mathias Unberath, Gregory D Hager, and Ferdinand K Hui. “Artificial intelligence to diagnose ischemic stroke and identify large vessel occlusions: a systematic review”. In: *Journal of neurointerventional surgery* 12 (2 2020), pp. 156–164. ISSN: 1759-8478 (cit. on p. 128).
 - [221] Kuanquan Wang, Mingwang Xu, Qiucheng Wang, Wen Cheng, Wei Wang, and Xinjie Liang. *Tumor Detection, Segmentation, and Classification Challenge on Automated 3D Breast Ultrasound*. Mar. 2022. DOI: 10.5281/zenodo.6362504. URL: <https://doi.org/10.5281/zenodo.6362504> (cit. on p. 131).
 - [222] Huijun Chen, Xihai Zhao, Rui Li, Haokun Li, Haozhong Sun, Ziming Xu, Haining Wei, Yan Li, Jiaqi Dou, and Xueyan Li. *Intracranial Aneurysm and Intracranial Artery Stenosis Detection and Segmentation Challenge*. Apr. 2024. DOI: 10.5281/zenodo.10990482. URL: <https://doi.org/10.5281/zenodo.10990482> (cit. on pp. 131, 138).
 - [223] Michael Baumgartner, P Jaeger, Fabian Isensee, and Klaus H Maier-Hein. “Retina U-Net for aneurysm detection in MR images”. In: *Automatic Detection and Segmentation Challenge (ADAM)* (2020) (cit. on p. 138).
 - [224] *MICCAI 2022 MELA Challenge: Mediastinal Lesion Analysis*. <https://mela.grand-challenge.org/challenge-program/>. 2022 (cit. on p. 138).
 - [225] *TDSC-ABUS 2023: Tumor Detection, Segmentation and Classification Challenge on Automated 3D Breast Ultrasound*. <https://tdscabus.github.io/>. Accessed: 2024-11-26. 2023 (cit. on p. 138).
 - [226] *INSTED: Intracranial Aneurysm and Intracranial Artery Stenosis Detection and Segmentation Challenge*. Accessed: November 26, 2024. 2024. URL: <https://www.codabench.org/competitions/2139/> (cit. on p. 138).
 - [227] Alejandro Newell and Jia Deng. “Associative Embedding: End-to-End Learning for Joint Detection and Grouping. CoRR abs/1611.05424 (2016)”. In: *arXiv preprint arXiv:1611.05424* (2016). URL: <https://arxiv.org/abs/1611.05424> (cit. on p. 138).

-
- [228] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. “Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 12475–12485 (cit. on p. 138).
- [229] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. “Cellpose: a generalist algorithm for cellular segmentation”. In: *Nature methods* 18 (1 2021), pp. 100–106. ISSN: 1548-7091 (cit. on p. 138).
- [230] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. “Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images”. In: *Medical image analysis* 58 (2019), p. 101563. ISSN: 1361-8415 (cit. on p. 138).
- [231] Bowen Cheng, Alex Schwing, and Alexander Kirillov. “Per-pixel classification is not all you need for semantic segmentation”. In: *Advances in neural information processing systems* 34 (2021), pp. 17864–17875 (cit. on p. 138).
- [232] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. “Masked-attention mask transformer for universal image segmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 1290–1299 (cit. on p. 138).
- [233] Ke Yan, Xiaoli Yin, Yingda Xia, Fakai Wang, Shu Wang, Yuan Gao, Jiawen Yao, Chunli Li, Xiaoyu Bai, and Jingren Zhou. “Liver tumor screening and diagnosis in CT with pixel-lesion-patient network”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2023, pp. 72–82 (cit. on p. 138).
- [234] Jakob Wasserthal, Hanns-Christian Breit, Manfred T Meyer, Maurice Pradella, Daniel Hinck, Alexander W Sauter, Tobias Heye, Daniel T Boll, Joshy Cyriac, and Shan Yang. “TotalSegmentator: robust segmentation of 104 anatomic structures in CT images”. In: *Radiology: Artificial Intelligence* 5 (5 2023) (cit. on p. 139).
- [235] Ziyang Huang, Haoyu Wang, Zhongying Deng, Jin Ye, Yanzhou Su, Hui Sun, Junjun He, Yun Gu, Lixu Gu, and Shaoting Zhang. “STU-Net: Scalable and Transferable Medical Image Segmentation Models Empowered by Large-Scale Supervised Pre-training”. In: *arXiv preprint arXiv:2304.06716* (2023). URL: <https://arxiv.org/abs/2304.06716> (cit. on p. 139).
- [236] Wenxuan Li, Chongyu Qu, Xiaoxi Chen, Pedro R A S Bassi, Yijia Shi, Yuxiang Lai, Qian Yu, Huimin Xue, Yixiong Chen, and Xiaorui Lin. “AbdomenAtlas: A large-scale, detailed-annotated, multi-center dataset for efficient transfer learning and open algorithmic benchmarking”. In: *Medical Image Analysis* 97 (2024), p. 103285. ISSN: 1361-8415 (cit. on p. 139).

- [237] Constantin Ulrich, Fabian Isensee, Tassilo Wald, Maximilian Zenk, Michael Baumgartner, and Klaus H Maier-Hein. “MultiTalent: A Multi-dataset Approach to Medical Image Segmentation”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*. Ed. by Hayit Greenspan, Anant Madabhushi, Parvin Mousavi, Septimiu Salcudean, James Duncan, Tanveer Syeda-Mahmood, and Russell Taylor. Springer Nature Switzerland, 2023, pp. 648–658. ISBN: 978-3-031-43898-1. URL: <https://arxiv.org/abs/2303.14444> (cit. on p. 139).
- [238] Jie Liu, Yixiao Zhang, Jie-Neng Chen, Junfei Xiao, Yongyi Lu, Bennett A Landman, Yixuan Yuan, Alan Yuille, Yucheng Tang, and Zongwei Zhou. “Clip-driven universal model for organ segmentation and tumor detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 21152–21164. URL: <https://arxiv.org/abs/2301.00785> (cit. on p. 139).
- [239] Jianpeng Zhang, Yutong Xie, Yong Xia, and Chunhua Shen. “Dodnet: Learning to segment multi-organ and tumors from multiple partially labeled datasets”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 1195–1204 (cit. on p. 139).
- [240] Gonglei Shi, Li Xiao, Yang Chen, and S Kevin Zhou. “Marginal loss and exclusion loss for partially supervised multi-organ segmentation”. In: *Medical Image Analysis* 70 (2021), p. 101979. ISSN: 1361-8415 (cit. on p. 139).
- [241] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. “Momentum contrast for unsupervised visual representation learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738 (cit. on p. 139).
- [242] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning*. PMLR, 2020, pp. 1597–1607. ISBN: 2640-3498 (cit. on p. 139).
- [243] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. “Emerging properties in self-supervised vision transformers”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9650–9660 (cit. on p. 139).
- [244] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, and Alaaeldin El-Nouby. “Dinov2: Learning robust visual features without supervision”. In: *arXiv preprint arXiv:2304.07193* (2023) (cit. on p. 139).
- [245] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. “Self-supervised learning from images with a joint-embedding predictive architecture”. In: *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, pp. 15619–15629 (cit. on p. 140).

- [246] Keyu Tian, Yi Jiang, Qishuai Diao, Chen Lin, Liwei Wang, and Zehuan Yuan. “Designing bert for convolutional networks: Sparse and hierarchical masked modeling”. In: *arXiv preprint arXiv:2301.03580* (2023) (cit. on p. 140).
- [247] Jingya Liu, Liangliang Cao, Oguz Akin, and Yingli Tian. “Accurate and robust pulmonary nodule detection by 3D feature pyramid network with self-supervised feature learning”. In: *arXiv preprint arXiv:1907.11704* (2019) (cit. on p. 170).
- [248] Matthew C Hancock and Jerry F Magnan. “Lung nodule malignancy classification using only radiologist-quantified image features as inputs to statistical learning algorithms: probing the Lung Image Database Consortium dataset with two statistical learning methods”. In: *Journal of Medical Imaging* 3 (4 2016), p. 44504. ISSN: 2329-4302 (cit. on p. 172).