DISSERTATION

submitted
to the
Combined Faculty for the Natural Sciences and for Mathematics
of the
Ruperto-Carola University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Put forward by
Angela Eigenstetter
Born in Bogen
Oral examination:

Learning Mid-Level Representations for Visual Recognition

Advisor: Prof. Dr. Björn Ommer

Abstract

The objective of this thesis is to enhance visual recognition for objects and scenes through the development of novel mid-level representations and appendent learning algorithms. In particular, this work is focusing on category level recognition which is still a very challenging and mainly unsolved task. One crucial component in visual recognition systems is the representation of objects and scenes. However, depending on the representation, suitable learning strategies need to be developed that make it possible to learn new categories automatically from training data. Therefore, the aim of this thesis is to extend low-level representations by mid-level representations and to develop suitable learning mechanisms.

A popular kind of mid-level representations are higher order statistics such as self-similarity and co-occurrence statistics. While these descriptors are satisfying the demand for higher-level object representations, they are also exhibiting very large and ever increasing dimensionality. In this thesis a new object representation, based on curvature self-similarity, is suggested that goes beyond the currently popular approximation of objects using straight lines. However, like all descriptors using second order statistics, it also exhibits a high dimensionality. Although improving discriminability, the high dimensionality becomes a critical issue due to lack of generalization ability and curse of dimensionality. Given only a limited amount of training data, even sophisticated learning algorithms such as the popular kernel methods are not able to suppress noisy or superfluous dimensions of such high-dimensional data. Consequently, there is a natural need for feature selection when using present-day informative features and, particularly, curvature self-similarity. We therefore suggest an embedded feature selection method for support vector machines that reduces complexity and improves generalization capability of object models. The proposed curvature self-similarity representation is successfully integrated together with the embedded feature selection in a widely used state-of-the-art object detection framework.

The influence of higher order statistics for category level object recognition, is further investigated by learning co-occurrences between foreground and background, to reduce the number of false detections. While the suggested curvature self-similarity descriptor is improving the model for more detailed description of the foreground, higher order statistics are now shown to be also suitable for explicitly modeling the background. This is of particular use for the popular chamfer matching technique, since it is prone to accidental matches in dense clutter. As clutter only interferes with the foreground model contour, we learn where to place the background contours with respect to the foreground object boundary. The co-occurrence of background contours is integrated into a max-margin framework. Thus the suggested approach combines the advantages of accurately detecting object parts via chamfer matching and the robustness of max-margin learning.

While chamfer matching is very efficient technique for object detection, parts are only detected based on a simple distance measure. Contrary to that, mid-level parts and patches are explicitly trained to distinguish true positives in the foreground from false positives in the background. Due to the independence of mid-level patches and parts it is possible to train a large number of instance specific part classifiers. This is contrary

to the current most powerful discriminative approaches that are typically only feasible for a small number of parts, as they are modeling the spatial dependencies between them. Due to their number, we cannot directly train a powerful classifier to combine all parts. Instead, parts are randomly grouped into fewer, overlapping compositions that are trained using a maximum-margin approach. In contrast to the common rationale of compositional approaches, we do not aim for semantically meaningful ensembles. Rather we seek randomized compositions that are discriminative and generalize over all instances of a category. Compositions are all combined by a non-linear decision function which is completing the powerful hierarchy of discriminative classifiers.

In summary, this thesis is improving visual recognition of objects and scenes, by developing novel mid-level representations on top of different kinds of low-level representations. Furthermore, it investigates in the development of suitable learning algorithms, to deal with the new challenges that are arising form the novel object representations presented in this work.

Zusammenfassung

Ziel dieser Arbeit ist es, die visuelle Erkennung von Objekten und Szenen, durch die Entwicklung neuer Mid-Level Repräsentationen und dazugehöriger Lernverfahren zu verbessern. Insbesondere beschäftigt sich diese Arbeit mit Kategorielevelobjekterkennung, die immer noch eine herausfordernde und großteils ungelöste Aufgabe darstellt. Ein wichtiger Bestandteil visueller Erkennungssysteme ist die Repräsentation von Objekten und Szenen. Jedoch müssen in Abhängigkeit der jeweiligen Repräsentation geeignete Lernstrategien entwickelt werden, die es ermöglichen neue Kategorien automatisch anhand der Trainingsdaten zu lernen. Daher, ist das Ziel dieser Arbeit, Low-Level Repräsentationen durch Mid-Level Repräsentationen zu erweitern und geeignete Lernverfahren zu entwickeln.

Eine häufig verwendete Mid-Level Repräsentation sind Statistiken höherer Ordnung, wie Self-Similarity und Co-occurrence Statistiken. Diese Deskriptoren erfüllen die Forderung nach Objektrepräsentationen auf einem höheren Level, weisen jedoch eine sehr hohe und immer größer werdende Dimensionalität auf. In dieser Arbeit wird eine neue Objektrepräsentation basierend auf Curvature Self-Similarity entwickelt, die über die momentan gängige Approximation von Objekten durch gerade Linien hinausgeht. Allerdings hat dieser Deskriptor, wie alle Deskriptoren die Statistiken zweiter Ordnung verwenden, eine sehr hohe Dimensionalität. Obwohl die hohe Dimensionalität die Diskriminabilität verbessert, wird sie zu einem kritischen Problem durch mangelnde Generalisierung und den sogenannten Fluch der Dimensionalität. Unter Verwendung einer begrenzten Menge von Trainingsdaten können selbst hochentwickelte Lernalgorithmen, wie die beliebten Kernelmethoden, die verrauschten und überflüssigen Dimensionen solcher hochdimensionaler Repräsentationen nicht unterdrücken. Infolgedessen besteht ein natürliches Bedürfnis der Featureselektion durch die Verwendung heutiger hoch-informativen Deskriptoren und insbesondere von Curvature Self-Similarity. In dieser Arbeit wird daher ein eingebetteter Featureselektions-Algorithmus für Support Vektor Maschinen entwickelt um die Komplexität zu reduzieren und die Generalisierung von Objekterkennungsmethoden zu verbessern. Die vorgeschlagene Curvature Self-Similarity wird zusammen mit der eingebetteten Featureselektion in ein weitverbreitetes State-of-the-art Objekterkennungs Framework integriert.

Der Einfluss von Statistiken höherer Ordnung auf die Kategorielevelobjekterkennung wird weiter anhand von gelernten Co-Occurrences zwischen Vordergrund und Hintergrund untersucht, die die Anzahl der falschen Detektionen reduzieren. Während der vorgeschlagene Curvature Self-Similarity Deskriptor das Model für die detaillierte Beschreibung des Vordergrunds liefert, wird nun gezeigt, dass Statistiken höherer Ordnung sich auch dafür eignen den Hintergrund explizit zu modellieren. Dies ist von besonderem Nutzen für die Chamfer Matching Methode, da diese anfällig für zufällige übereinstimmungen in dichtem Hintergrundrauschen ist. Da Hintergrundrauschen sich ausschließlich störend auf die Modelkontur des Vordergrunds auswirkt wird gelernt, wo die Hintergrundkonturen bezüglich der Vordergrundkontur platziert werden müssen. Die Co-Occurrence von Hintergrundkonturen wird in ein Max-Margin Framework integriert. Daher kombiniert der vorgeschlagene Ansatz die Vorteile einer genauen

Detektion von Objektteilen durch Chamfer Matching und die Robustheit von Max-Margin Lernverfahren.

Obwohl Chamfer Matching eine sehr effiziente Methode für die Erkennung von Objektteilen ist, basiert die Erkennung nur auf einem einfachen Distanzmaß. Im Gegensatz dazu, werden Mid-Level Parts und Patches explizit dafür trainiert, zwischen korrekten Detektionen im Vordergrund von falsch positiven im Hintergrund zu unterscheiden. Aufgrund der Unabhängigkeit von Mid-Level Patches und Parts ist es möglich eine große Menge von instanzspezifischen Partklassifikatoren zu trainieren. Diese Vorgehensweise steht im Gegensatz zu den augenblicklich mächtigsten diskriminativen Ansätzen, die typischerweise nur für eine kleine Anzahl von Parts anwendbar sind, da diese die räumlichen Abhängigkeiten zwischen den Parts modellieren. Aufgrund ihrer Anzahl ist es nicht möglich direkt einen Klassifikator für die Kombination der Parts zu lernen. Aus diesem Grund werden die Parts in eine geringere Anzahl von überlappenden Kompositionen zufällig zusammengruppiert, die dann mit einem Max-Margin Ansatz trainiert werden. Im Gegensatz zu dem üblichen Vorgehen von kompositionellen Ansätzen, werden keine semantisch sinnvollen Ensembles gesucht. Vielmehr werden randomisierte Kompositionen gesucht, die diskriminativ sind und über alle Instanzen einer Kategorie generalisieren. Alle Kompositionen werden durch eine nichtlineare Entscheidungsfunktion kombiniert, die die leistungsfähige Hierarchie aus diskriminativen Klassifikatoren vervollständigt.

Zusammenfassend beschäftigt sich diese Arbeit mit der Verbesserung visueller Erkennung von Objekten und Szenen, durch die Entwicklung neuer Mid-Level Repräsentationen, basierend auf unterschiedlichen Low-Level Repräsentationen. Darüber hinaus, untersucht diese Arbeit die Entwicklung von geeigneten Lernverfahren, die die neuen Herausforderungen, die sich aus den in dieser Arbeit vorgestellten neuen Objektrepräsentationen ergeben, bewältigen.

Acknowledgments

My sincere thanks go to all the people that have contributed to the successful completion of this thesis with their constant help and support over the last years.

First of all I would like to thank my advisor Prof. Dr. Björn Ommer for supervising this thesis. I am especially thankful for the numerous discussions where he gave me advice and remarks on my research. Also I would like to thank Prof. Dr. Christoph Schnörr for agreeing to review this thesis.

Thanks to all my colleagues at the HCI for entertaining discussions during lunch and after work. Especially I would like to thank all members of the computer vision group: Antonio, Pradeep, Boris, Peter, Hongwei, Jose, Masato, Niko, Tobias, Christoph and Timo. I enjoyed our fruitful discussions about computer vision and every day life. In particular I am grateful for the contribution of all my colleagues to our good work doing research and writing articles.

I am grateful for Tanjas and Karins support in all administrative matters and entertaining discussions. Furthermore, I would like to thank the technical support provided by all HCI and IWR administrators: Ole Hansen, Dominic Spangenberger, Markus Nullmeier, Jürgen Moldenhauer, Markus Ridinger, Martin Neisen and Hermann Lauer.

Thanks also go to all my friends for the nice time we spent together the past years. Special thanks go to Markus for his love and constant believe in me and my skills. Finally I want to thank my father for his support and encouragement in every life situation.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Human Visual Perception of the World

The ability to see is the most important of our senses. In our everyday life we are heavily relying on our visual sense for nearly every interaction with the surrounding world. However, what we mean when we say *seeing* is actually much more than that. When we talk about *seeing*, we mean the ability to perceive our environment. We are interpreting what we see, e.g. we are categorizing objects, recognize persons and take adequate actions according to the overall situation, such as greet familiar persons. All this is so natural for us that we are not even thinking about this constantly ongoing process of perception. To get a better understanding of how we are perceiving the world let us shortly review the human perceptual process.

It begins with the distal stimulus in the environment. Objects are reflecting light, which is resulting in a light stimulus, that can be received by our eyes. The light enters the eye through the pupil and produces a projection of the environment on the retina (proximal stimulus). Receptors in the eye respond to the light and transform light energy into electrical energy. The electrical signal is then transmitted to the brain and is then processed by the brain resulting in a perception. The perception step is followed by a recognition and an action step. Researches have shown that perception and recognition are separate processes. Nevertheless, these steps do not always follow each other, but can also happen at the same time or reverse order. Moreover the action can also change the perception and the recognition [71].

The projection of points from the 3D environment onto the retina is a well-defined problem as each point in the 3D environment maps exactly onto one point in the 2D retinal image. However, in order to perceive the world as it is, the real-world 3D information needs to be recovered using the 2D retinal image. This task, known as the inverse problem, is an ill posed problem as each point in the 2D retinal image could have resulted from an infinite number of points in the 3D environment. Despite the fact that the inverse problem

doesn't have a unique correct solution, our visual systems manages to recover the correct 3D information surprisingly well. But how is this possible? The common consensus among most vision theorists is that additional information is used for perception. This additional knowledge, from experiences we made (probably throughout our whole life), is affecting our perception and recognition, even though we are usually not aware of it. This concept of *unconscious inference* was first suggested by Hermann von Helmholtz [172]. In 1867 he was already aware of the gap between the retinal image and correct perceptual interpretation of the 3D world. He claimed that vision requires a process of *inference* to transform the retinal 2D information into the correct perceptual interpretation of the 3D world [72, 129].

From the ideas of Helmholtz until today many different vision theories have been proposed. A review of all this theories is going beyond the scope of this thesis and therefore only a short review will be given on one of the most important developments that changed the entire understanding of vision: the invention of the computer. Since we are living today in a world in which we are using computers for almost everything, it is hard to imagine what life was like without them. However, imagining a life without computers can help to understand what a fundamental change the invention of the computer might have been. And as it did with many other fields it also dramatically changed the field of perceptual psychology. On the one hand the idea to simulate perceptual processes on a computer lead to more explicit theories because the programming of a computer requires detailed information. Furthermore the application of information processing theories by Marr [118] in the field of psychology lead to a novel framework for understanding the concepts of vision. However, the invention of the computer did not only influenced the field of perceptual psychology but also inspired scientists to simulate all kind of other processes, e.g. the simulation of intelligence which gave raise to the field of artificial intelligence.

1.2 The Dream of Artificial Intelligence

For most people the term *artificial intelligence* is best known from its use in science fiction books and films or series. Typically, in this context, the term refers to intelligent robots whose shape is resembled that of the human body and are able to act like humans. The idea to build machines that are like humans can be considered the dream of artificial intelligence. From a philosophical point of view one can distinguish between strong and weak artificial intelligence. Weak artificial intelligence refers to the simulation of intelligent behavior, i.e. machines that act if they were intelligent. Compared to strong artificial intelligence which means machines are actually intelligent and not only acting as they were. From a more technical point of view this difference was never really an issue, since the goal is to build machines that act in an intelligent way as defined in 1950 by Alan Turing, who suggested a behavioral intelligence test [163] known as the *Turing Test*. The idea is that, if a machine passes the test, it can be considered as *intelligent*. The test is supposed to be a conversation between a person and a computer. The person writes down questions which the computer is supposed to answer. If the person cannot tell if he or she was talking to a computer or a real person the computer passes the test. The so

called *total Turing Test* provides an additional video signal, and the person can also show or pass things to the computer. To pass the test a computer needs to handle the following tasks [145]:

- *natural language processing* to communicate using a common language e.g. English
- *knowledge representation* to store knowledge
- *automated reasoning* to use the stored information to answer questions and draw new conclusions
- *machine learning* to adapt to new circumstances and detect and extrapolate patterns
- *computer vision* to perceive objects
- *robotics* to manipulate objects and move about

The research field of artificial intelligence emerged a few years after the suggestion of the *Turing Test*. Originally it was defined by a summer workshop organized by John McCarthy in 1956. The proposal of McCarthy [120] was making the suggestion that weak AI is possible. The workshop envolved 10 researchers that were working two month on the topic of artificial intelligence. The goal was to describe learning or other intelligent tasks in such detail that it could be simulated by a computer.

After some early successes solving simple AI tasks, such as solving smaller mathematical theorems, researches were very enthusiastic that more complex problems could be solved just using better hardware. A famous example of this early enthusiasm is the claim of Simon in 1957 that a computer will be chess world cup champion in ten years from then. However, things didn't progress so fast and many AI systems failed to perform well on more complex tasks. The reason for that might have been that the difficulty of many tasks was underestimated. Especially the importance of additional knowledge was not considered. Another problem of early AI was that most of the systems were based on exhaustive search algorithms that tried out a series of steps until a solution was found. However, contrary to the former common believe, it is not possible to solve larger problems by simple search algorithms due to the huge computational complexity. Nevertheless the forecast of Simon became true in 1997. Not ten years, but forty years later, IBM's Deep Blue won against the chess world champion. Until today AI research made great achievements, which are now also available in our everyday life, such as voice recognition on smart-phones and hotlines, automatic translation, driver assistance systems in cars and many more. Probably the most recent and amazing AI achievement was that IBM's super computer Watson clearly won the game show *Jeopardy!* against two former winners in 2011. This game is particularly challenging, as the tasks are given as, typically ambiguous, answers to which the candidates need to find the correct question.

1.3 Computer Vision

As mentioned in the previous section AI theorists originally tried to simulate difficult intellectual tasks as playing chess and proving mathematical theorems. Only later it was realized that programming computers to perceive the environment visually is a

challenging and useful goal. This was the start of the field which is today known as *Computer Vision*. The great goal of computer vision is to develop algorithms which extract and interpret all the information about the environment. Originally this was considered one of the easier AI tasks that needed to be solved in order to build robots with intelligent behavior. Since perceiving the environment is done effortlessly by humans it was thought to be a rather easy task. People were more focused on teaching the computer to perform tasks that were considered to be intellectual challenging. A famous anecdote tells that in 1966 Marvin Minsk at MIT asked one of his students to "spend the summer linking a camera to a computer and getting the computer to describe what it saw" [17]. It turned out that the problem was slightly more difficult as we know today.

But why is vision so difficult? As discussed in Section 1.1 vision is an inverse problem which doesn't provide enough information to reliably reconstruct 3D information from the 2D retinal images. While the projection from the 3D world onto 2D retinal image can find a single correct solution this is not possible the other way round. Perceptual researchers are still trying to understand how humans can solve this problem, since the visual process is rather complex. Taking this knowledge into account, it is not surprising, that teaching a computer to *see* is also a demanding task.

Until today the field of computer vision was expanding and diversifying in many directions. The tasks that are attending most of the attention in the high-level vision research community are:

- Segmentation
- Recognition
- 3D Reconstruction
- Motion and Tracking

Note, that this is a very coarse structuring and many of these fields can be divided in several very diverging subfields. Additionally this structuring is only considering the tasks that are tackled and not the diverse approaches that are explored to solve these problems which often developed to be own research fields.

1.4 Visual Recognition

The focus of this thesis is the visual recognition of objects and scenes in still images. Visual recognition can be divided in *classification* and *detection*. The classification task is to categorize an object or scene shown in an image or image region. In the case of objects, however, it is not only interesting which objects are present in the image but also where exactly they occur, since images typically contain a whole scene with more than one object. This task is called detection. This also shows the close connection between the task of object detection and scene classification, as it is important to categorize each object in the image, the scene label provides additional information about the context of the detected objects and therefore provides a higher-level abstraction. Due to the close connection of the two problems and the focus on object detection of this thesis discussions

will be on the level of objects if the topic is not specifically about scenes. However, typically statements about objects are also accounting for scenes.

The classification task can be divided into two broad categories, *instance level* recognition and *category level* recognition. Instance level recognition is dealing with the task of recognizing a particular object instance (e.g. my car), while category level recognition is dealing with the much more challenging task to recognize any instance of an object category (e.g. car in general not just my car). While the problem of instance level recognition can be solved very well by current state-of-the-art algorithms, the problem of category level recognition is still a very challenging task to be solved. Compared to humans, the current state-of-the-art systems have still not reached the categorization performance of a two year old child [157].

In order solve category level object recognition tasks one has to solve two related problems:

Representation Representation is crucial for the success of a recognition system. Images on a computer are stored as three dimensional matrices where each pixel is represented by a three dimensional vector representing e.g. the strength of the colors red, green and blue. However, this is not an appropriate representation for recognition, since a single image has ten thousands of pixels and is therefore providing a huge amount of data that cannot be handled efficiently. Instead, the image needs to be represented by a more compact description, e.g. using low-level features that are simple statistics on the image or the edges of an image. Starting with brightness values of image pixels and simple edge histograms, descriptors evolved and more sophisticated features were suggested. The probably most widely used and best performing image descriptors today are modeling edge orientation histograms. In the last few years also more complicated image statistics such as co-occurrence and self-similarity have been utilized. Such higher-level statistics are leading to more robust image descriptions than first order statistics such as simple histograms.

In order to recognize objects one needs not only to represent the image, but also the object category that we want to recognize. Such a representation needs to be compact enough to be easily stored and applied to new images but also needs to be detailed enough to make it possible to distinguish between similar object categories, e.g. cow and horse. A recognizable trend regarding the representation of objects is that image statistics first have been computed over the whole image/object, while later approaches are splitting up the image/object into smaller regions either utilizing a rigid grid, special labellings or by utilizing other automatic extraction methods such as e.g. interest points detectors that give indications for specifically useful regions of the image. Such image regions, or parts, can be treated as an unordered collection of features resulting in a bag-of-words representation, or relations between the parts can be explicitly modeled either by defining their configuration by hand, or by learning their spatial relationship. Besides the classical part-based approaches, which are combining parts first by a spatial model and then train all parts together, learning individual part classifiers first and take care of the spatial arrangement afterwards has been applied quite successfully. Furthermore parts can be combined by grouping, to form larger more meaningful compositions. Typically this is done in a hierarchy were the compositions of the last layer are combined to form new

compositions until the whole object is represented. Instead of parts that are mainly defined by their visual appearance, another popular representation concept are attributes which are defined to be a property of an object which can be named by a person. While parts define the visual appearances of an object, attributes are describing its properties, e.g. furry or red.

It is noteworthy that all these approaches are seeking to bridge the gap between simple image descriptors and the object by dividing it in its constituent parts or characteristics. Recently, the term mid-level representation was coined, for such object representations that are providing an intermediate representation of objects and scenes.

Learning In order to find such a general object representation one can either define suitable object characteristics by hand or learn them automatically. However, the definition of suitable object characteristics by a human expert, is a very time consuming and costly task. Therefore, instead our goal is to automatically learn a model capturing significant characteristics for and arbitrary object or scene category. In order to solve this task one can make use of machine learning techniques, which are able to learn from a set of training data, which characteristics, captured by the representation, are important to distinguish between categories.

In general, one cannot tell, what is the best learning strategy to recognize objects. There exists a large amount of learning strategies which have been originally designed for general machine learning problems and have been adapted and successfully used to solve visual recognition tasks. However, the decision for a certain learning strategy is not only depending on the task that needs to be solved. The kind of representation plays an important role for the selection of a proper learning method. In the context of mid-level representations the development of object descriptors using co-occurrences and self-similarity leads to very high-dimensional descriptors. Therefore learning strategies need to be applied, that are able to deal with such preprocessed and high-dimensional data. On the other hand, when individual part classifiers are trained, the amount and detail of training annotation is influencing the choice of a learning strategy essentially. If the location of object parts is given by human annotation, straight forward training of individual part classifiers can be performed, as they have been developed for the learning of object classifiers. If such information is not provided, proper parts need to be defined automatically. But whats a good strategy to find such parts? And when we decided for a suitable part, how do we find a set of positive training data? Typically these problems are solved using heuristics and unsupervised learning methods. However, depending on the part representation, the dataset and the task, that needs to be solved, different strategies need to be applied. Going one step further brings us to the problem of grouping parts into compositions. Here it is preferable to have a learning strategy which is able to efficiently deal with construction of compositional hierarchies.

1.5 Objectives of the Thesis

The aim of this thesis is to develop learning algorithms for visual recognition. While the main focus of this thesis is on category level object detection, a part of the thesis also

deals with scene classification and shows that well designed object detection algorithms are also well applicable to scenes. Since, the problem of object and scene recognition are closely related it is consequential to use the same representation and learning algorithm for both tasks. In the remainder of this section, the objectives are explained for the task of object detection, as this is the main task tackled in this thesis. However, due to the relatedness of object detection and scene classification the goals typically also account for scenes.

Specifically this thesis engages in the automatic learning of object models for the representation of arbitrary object categories from labeled training data. Labeled, in this case, means localized object class labels using bounding boxes. This is due to the fact that labeling beyond the bounding box label, is for one thing time consuming and costly, and on the other hand, because this is not a well-defined problem for a lot of categories, leads to ambiguous or unsuitable training data. Finding an object in an image e.g. an aeroplane and drawing a bounding box that encloses the object is a task for which the result of many different persons will be more or less the same. However, on the level of parts the problem becomes also much more ambiguous for the annotators. What are the parts of an aeroplane? The labeling result of different persons will look much more different than for the task on the object level. Furthermore, the question is, if parts are labeled because of semantical similarity or due to visual similarity. However, aeroplane is still an object category for which one can imagine to get reasonable part labels, even if the variance of labeling between different persons is high. Thinking about a category defined as potted plant, the task of finding parts can be extremely challenging, if not impossible. As a result label information beyond the bounding box level might not even be helpful. Therefore this work is utilizing bounding box labels as they are most reliable.

In this work both subproblems of recognition are tackled: the representation and the learning problem (see Section 1.4). The goal on the representation side is to design novel mid-level representations for objects to improve existing low-level object representations. Low-level object representations are typically utilizing simple statistics to describe the object. Such representations are easy to compute and simple to deal with during the learning process. However, such simple features are often not providing enough characteristic information to learn a reliable object model. Since object models need to describe the large intra-class variabilities of an object category, there is the need for a higher-level concept that goes beyond commonly used simple representation. It is of particular interest that mid-level representations are developed, that are bridging the gap between low-level representations, that are just compressing the pixel content of an image, and the complex appearance of an object. This thesis investigates the development of mid-level representations build upon different visual recognition systems that are utilizing different low-level object descriptors and are combined with different modeling schemes. Specifically this thesis is dealing with, appearance and shape based representations, and holistic and part based object models. Therefore the second central point of this thesis is the design of appropriate learning algorithms for the suggested mid-level representations. It is of particular importance, to devise learning algorithms that are able to deal with the different requirements that arise from the appendent basic framework.

For the improvement of holistic object models a novel curvature self-similarity descriptor is suggested, which is exploiting curvature as a cue to perform the visual search task, and

in addition exploits the improved descriptive power of self-similarity descriptors. While self-similarity representations are providing more detailed and accurate characterization of the object, they are typically exhibiting larger dimensionality and contain a larger amount of noise than low-level features. This can cause systems to suffer from curse of dimensionality and overfitting. Therefore noisy and superfluous dimensions need to be discarded before or during the learning process. In this work the learning process is described as an optimization problem and sparsity is enforced by an L1 regularization on the model parameter.

Furthermore, chamfer matching, a contour based object representations, is improved by reducing the amount of accidental false positive object detections in dense clutter. Accidentalness of a match is measured by introducing a set of generic background contours which are placed relative to the foreground contour. Then the joint co-occurrences between foreground and background is determined. Such co-occurrences are very unlikely to appear by accident and are therefore providing a robust mid-level representation that helps to identify false positive matches in dense clutter. To capture all pairs, triples, quadruples etc. a non-liner radial basis function kernel that is comprising an infinite amount of feature combinations is utilized.

Moreover a robust part-based recognition system utilizing a large amount of mid-level parts is developed. This goes beyond the common scheme of utilizing a small number of general parts. To deal with the high dimensionality caused by a large amount of specific parts they are grouped into stronger compositions. A final non-linear discriminative classifier is trained to blend compositions. In this manner a powerful visual recognition framework is developed that is combining the concept of mid-level parts and compositions into a powerful hierarchy of classifiers.

1.6 Challenges

As the aim of this thesis is to devise proper mid-level representations for visual recognition of objects and scenes it is important to understand what exactly makes this task challenging. We as humans are able to recognize a large variety of objects almost effortlessly. Therefore, we are often lacking the sensitivity for the problems that occur during the development of object detection systems. In this Section a number of conditions will be discussed which have a large impact on the difficulty of the classification and detection task.

Categorization vs. Identification The identification of a specific instance of a class can already be solved well by current computer vision systems. On the other hand the task of object detection on the category level is still considered an extremely difficult task. For us, there is almost no difference between the tasks, and if we think about it, identification of a specific object is even more difficult for us than *just* categorization. Thus the question arises “Why is categorization so difficult for recognition systems ?”.
From perceptual research we know that children need to learn to categorize objects. Similar to computer vision systems, young children confuse similar object categories,

(a) (b)

Figure 1.1: Due to similar texture and shape different object categories, such as the dog in (a) and the horse in (b) can look quite similar.

(a) (b) (c)

Figure 1.2: Aeroplanes exhibit a high intra-class variability reaching from (a) vintage propeller driven airplanes over (b) commercial airliners to (c) modern combat aircrafts.

such as *horse* and *dog*. Of course, there is a certain similarity between this two objects: both have fur and four legs and can have a quite similar appearance as shown in Figure 1.1. Over time children learn the characteristics of objects and are able to distinguish similar object categories. Additionally they also learn more and more object classes and even a hierarchy of categories, e.g. most people would agree that the animal in Figure 1.1a is a dog, nevertheless many of them are aware that it is also a dalmatian. Similar to children object detection systems also need to learn what is characteristic for a category. However, contrary to children, who learn the categorization of objects over years and probably during their whole life, having an endless amount of training data, computer vision systems need to solve the same task with a comparable very limited amount of training data and classes. This thesis is in particular dealing with a rather small number of classes (fewer than 100) and a limited amount of training data.

High intra-class variance Object categories like *aeroplane* or *dog* are exhibiting a very high intra-class variance, i.e. objects within one category are exhibiting very different visual appearances. Depending on the object category intra-class variance can have very diverse occurrences. As shown in Figure 1.2 the category *aeroplane* is reaching from vintage propeller driven airplanes over commercial airliners to modern combat aircrafts which exhibit a very different appearance. Moreover, even the same instance of an object

(a) (b) (c)

Figure 1.3: Non-rigid object classes such as horses exhibit large appearance variations due to articulation.

Figure 1.4: Objects can have different scales and viewpoints which makes detection of these objects more challenging.

looks totally different from another viewpoint, e.g. looking at an aeroplane from the front and from the side (see Figure 1.4). Additionally, non-rigid object categories such as *person*, *dog*, *horse* etc. are showing additional variation in their appearance due to articulation. See Figure 1.3 for some examples. Besides appearance based variance there is also variance in scale. As shown in Figure 1.4 the same image can contain very large objects close to the camera and also very small objects in the background of the scene. In this thesis all of this variations need to be considered.

Low inter-class variance Another factor for the complexity of the object detection task is low inter-class variance. In the case that two classes are very similar to each other the task of learning an appropriate object model to distinguish the two classes becomes more difficult, as the important characteristics for a correct classification might be very subtle. The model needs to be invariant to the intra-class variance and sensitive to the low inter-class variance. Due to the difficulty of exactly this problem a new subtask of object recognition called *fine grained object recognition* has developed. Fine grained object recognition particularly deals with the task of low inter-class variance. The task addressed, are distinction into species of animals, car models, architectural styles etc., where the differences between categories can be very subtle. It is likely that some of the standard category level techniques need to be reassessed to develop algorithms for fine grained recognition.
While fine grained object recognition is beyond the scope of this thesis, high similarity

(a) (b)

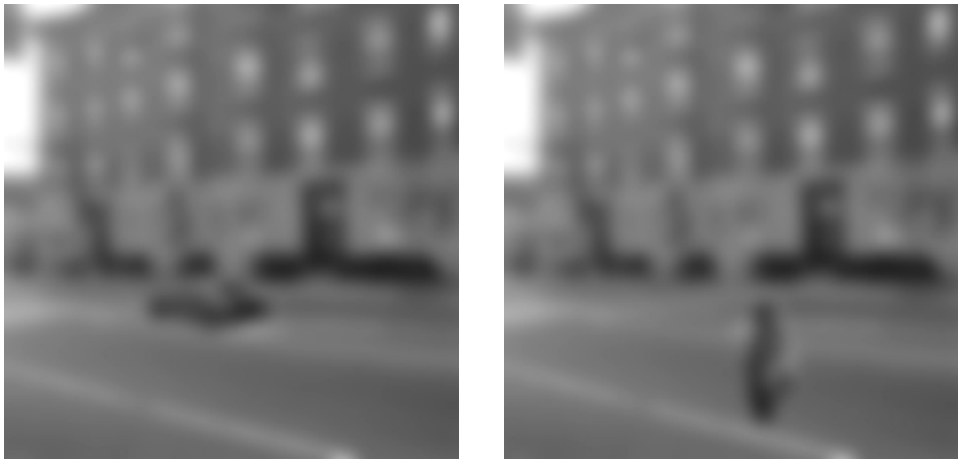
Figure 1.5: (a) Occlusion and (b) truncation of objects make parts of the object invisible and therefore complicate the detection process.

between categories, such as between *motorbike* and *bicycle* need still to be handled.

Occlusion and truncation In the case that more than one object instance is present in the image, or that the object of interest is not in the focus of the scene, the object might be occluded by other objects or other parts of the scene considered as background. In the case of occlusions parts of the object are hidden by other non-related objects or background. A special case is self-occlusion, where the object is occluding itself. This typically happens with objects that are highly articulated. A similar but slightly different situation occurs, when an object is truncated, i.e. the image is only showing a part of the whole object. See Figure 1.5 for an example of occlusion and truncation. In both cases, some parts of the object are not visible in the image, which means that there is no visual information available. Occlusions and truncations make recognition of objects more difficult as the model needs to be flexible to missing parts without becoming to unspecific to distinguish the object for arbitrary background.

Image properties Real world images collected from the Internet are often not showing the object of interest in the focus of the image and are exhibiting certain difficulties for object detection compared to images taken by a photographer in a controlled environment.

- **Varying illumination** When looking at an object that is partly in the shadow we are aware, that the color of the object is not different in the light and the shadow. However, in the image the two colors have different values. Therefore, object representation needs to compensate for such changes of colors due to shadows etc.
- **Background clutter** Real world images of objects are taken in front of arbitrary background. This background might contain other objects, or just structures of plants etc., which make it more difficult for the detection algorithm to distinguish the object of interest from the background.
- **Low image quality** Especially images collected from the Internet are often of relatively low quality. The image resolution is typically rather small since storage



(a) (b)

Figure 1.6: This scene is typically described as (a) a street scene with a car and (b) a street scene with a pedestrian. However, the image patch recognized as the pedestrian is the same as the one recognized as car. The only difference is that the car patch was rotated by 90 degrees to be perceived as a pedestrian [160]. From this example it becomes clear that in case of low image quality the context is becoming more important for recognition than the appearance.

size of the images grows with the resolution. Moreover the standard jpg format can exhibit artifacts that cause distortions during detection. This issues become more critical for small objects. To reliable recognize such objects the human observer typically relies on the context of the object rather than its actual shape etc. See Figure 1.6 for an example how context affects our visual perception. The two blurry scenes are both containing an object that is not recognizable when looking at it in isolation. However, in the context of whole scene, here a street scene, the objects can be easily recognized by test persons as car and pedestrian. In fact the two blurry objects recognized as car and pedestrian are the same image patch with different orientation. In computer vision, context-based methods are typically utilizing the outputs of object detection systems and then update the recognition results based on the recognized objects in the scene. In that way reliable detections of objects can give evidence to recognize other objects or to discard false hypotheses. The aim of this thesis is to create a reliable object detection systems while context approaches are beyond the scope of this work.

1.7 Contributions

The contributions of this thesis are summarized as follows:

- Exploration and development of more informative mid-level representations, build on top of different basic features, for object representation under consideration of different object modeling schemes.
- Design of appropriate learning algorithms, suitable for novel mid-level

representations and different object modeling scheme.

- Integration of curvature self-similarity representation into a widely used state-of-the-art object detection framework. The novel mid-level descriptor utilizes co-occurrences between discriminatingly curved boundaries that provide a more detailed and accurate object description.
- Development of a novel embedded feature selection algorithm to reduce the extremely high dimensionality that is exhibited by object representations that are using more informative second order statistics, such as curvature self-similarity. Dimensionality reduction prevents the object detection system to suffer from curse of dimensionality and overfitting.
- Structured evaluation of the suggested curvature self-similarity representation in combination with the suggested feature selection algorithm shows the individual merit of both contributions and significant performance improvement over standard object descriptors on a standard benchmark dataset.
- Reducing the number of accidental matches in dense background clutter of state-of-the-art chamfer matching methods by learning the co-occurrences of generic background contours.
- Integration of the learned co-placement of background contours with foreground regularized templates in a single max-margin framework, build on top of state-of-the-art directional chamfer matching.
- State-of-the-art chamfer matching approaches are significantly outperformed by the suggested max-margin framework on standard benchmark datasets.
- Development of a part-based discriminative compositional hierarchy. Detailed analysis of different grouping strategies is provided and results show that randomized grouping outperforms the common rationale of compositional approaches to seek semantically meaningful compositions.
- Filling the gap between part-based representations and the whole object by utilizing compositions in a discriminative framework.
- Paradigm shift from small number of generic parts to a large number of instance specific parts and from strong to weak localization constraints. Separate evaluation of the suggested part classifiers support the potential of this approach.
- Additionally to the localization of objects in cluttered scenes, the suggested randomized part-based compositions provide a qualitative 2D reconstruction of the detected object.
- Application of randomized max-margin compositions on several object detection benchmark datasets and one scene classification benchmark dataset show state-of-the-art performance.

1.8 Organization of the Thesis

The remainder of this thesis is organized as follows:

Chapter 2 categorizes this work according to different object recognition paradigms. Furthermore, common low-level feature representations as well as several object modeling schemes are discussed. Finally the concept of mid-level object representation is introduced.

Chapter 3 focuses on holistic category-level object detection and the design of curvature self-similarity, a novel mid-level object representation. Due to the risk of overfitting and curse of dimensionality, the system, trained on a limited set of training data, has a natural need for feature selection. The development of a new embedded feature selection method is discussed in this chapter and shows how the reduction of noise uncovers the full potential of the high-dimensional curvature self-similarity representations.

Chapter 4 describes how mid-level representations can be used to improve state-of-the-art chamfer matching methods. In particular it describes how accidentalness in dense clutter is reduced, by placing generic background contours on the model contour and learning to distinguish the typical co-occurrence of these contours on cluttered background compared to actual objects. Additionally, this chapter describes how the suggested background regularization is integrated with foreground regularization, i.e. the relative importance of all model points of a template instead of treating them as independent.

Chapter 6 investigates the merit of randomized compositions for discriminative part-based object detection framework. Instance specific weakly localized parts are utilized in a compositional max-margin framework. This chapter provides results of careful evaluation of different composition techniques and the novel specific part classifiers. Beyond the localization of objects the suggested category level object detection framework provides a 2D parsing of the detected object.

Chapter 7 discusses conclusions of the presented thesis.

CHAPTER 2

OBJECT MODELS AND REPRESENTATIONS

The purpose of this chapter is to classify this work within the area of visual recognition, review common object representations and models, and introduce the concept of mid-level representations. This work examines mid-level object representations build on different low-level object representations using several object models. However, three main object recognition paradigms remain constant throughout all object recognition approaches presented in this thesis. In particular this work deals with 2D template matching approaches utilizing sliding windows (Section 2.1.1) and discriminative learning methods (Section 2.1.2). Additionally the level of supervision is kept fixed for all presented methods (Section 2.1.3). Furthermore, common low-level object representations (Section 2.2) and popular object models (Section 2.3) from this domain are reviewed. Finally, the concept of mid-level representations is introduced (Section 2.4).

2.1 Object Recognition Paradigms

2.1.1 3D Geometric Models and 2D Template Matching

An important approach of early object recognition was 3D geometric modeling of objects. Such 3D models are either based on holistic 3D object models or a collection of volumetric parts such as polyhedra [142], generalized cylinders [1, 125] and super-quadrics [134]. Such 3D models are providing a rich and view-point invariant description of objects and have been widely used until the 1990's. However, finding such shape abstraction for a whole object category remains a challenge to this day. Nevertheless, due to more powerful computers and machine learning techniques, 3D modeling is becoming more popular these days and some of the early ideas of 3D modeling are revisited [75, 81, 83, 174]. Gupta et al. [75] is following the idea of a "blocks world" presented by Roberts [142] who

suggested an approach for 3D scene reconstruction. In [75] real world outdoor scenes are reconstructed using objects that have volume and mass. Hoim et al. [83] use context information to model relationships between the objects in the 3D environment instead of the 2D image plane. This way perspective based distortions influencing the relationship between objects are eliminated. Hedau et al. [81] are describing indoor scenes, assuming that objects are aligned with the dominant direction of the scene. The dominant direction of the scene can be computed using their earlier work [80] where the spatial 3D layout of a scene is estimated using a box layout. After finding the box layout of the scene geometrical constraints are introduced that are considering size, visibility, and location of the object with respect to the room. Wang et al. [174] are following the idea presented in [80], but reduce the amount of supervision that is needed to identify clutter such as furniture, decoration etc. Instead of using labeling information for clutter, during the training phase of the classifier, they use latent variables which are learned automatically.

In the 1990's the object recognition philosophy shifted from such model-based approaches to view-based approaches, which are not utilizing 3D models but a set of 2D views. One popular technique in this area is template matching [136, 56], where a template is provided or learned from training data and then compared to image regions of a test image. The higher the matching score between the template and an image region the more likely it is that the object is present in that region. In order to localize objects in the image one can perform an exhaustive search, called sliding window [146, 171], where the image is partitioned into a set of overlapping windows. Each region is matched against the template and it is decided if the window contains the target object or not based on the matching score. Typically, sliding windows search not only over location, but also over different scales, to be able to recognize objects of different sizes.

To speed up such exhaustive search using sliding window one can reduce the number of hypotheses in different ways. One approach is to use a cascade of classifiers [171]. Faster, but typically weaker classifiers, are run first in a sliding window fashion and discard all hypotheses that are not containing the object. On the remaining hypotheses a stronger classifier is applied. This way one can utilize stronger classifiers in a reasonable amount of time. Recently general object proposal methods [3, 164] have gained increasing interest in the field of object recognition. These methods provide object hypotheses independent of the object class and greatly reduce the amount of hypotheses that have to be classified compared to and exhaustive search.

2.1.2 Discriminative and Generative Object Recognition Models

The overall goal of object recognition systems is to assign a class label to a given representation of an image or an image region. Instead of defining class specific characteristics by hand, e.g. a hand drawn template, it is much more common to learn such characteristics using a classifier. Depending on the type of classifier used one can divide recognition systems, or rather the classifiers used in the systems, into two broad categories: *generative* and *discriminative* classifiers.

Lets assume, that the object hypothesis that needs to be classified is already given in a

(a) Image class label (b) Bounding box annotation (c) Pixel wise annotation

Figure 2.1: Different amount of supervision are shown in (a)-(c). While in (a) only the class label *horse* is given (b) and (c) provide additional localization information on bounding box and pixel level respectively.

suitable representation $\boldsymbol{x} \in \mathbb{R}^n$ and that the optimal class label $y \in \{1, \ldots, k\}$ needs to be assigned by computing the posterior probability $p(y|\boldsymbol{x})$. A generative classifier learns from given training data the likelihood probability $p(\boldsymbol{x}|y)$ and the prior probability $p(y)$ for each class y. Equivalently, one can model the joint probability directly since $p(\boldsymbol{x}, y) = p(\boldsymbol{x}|y)p(y)$. In order to compute the posterior probability Bayes' theorem

$$p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y)p(y)}{p(\boldsymbol{x})} \tag{2.1}$$

$$\propto p(\boldsymbol{x}|y)p(y) \tag{2.2}$$

is applied and the most likely class is selected, i.e the class label with the highest posterior probability. Contrary to that, discriminative classifiers are directly learning the posterior probability $p(y|\boldsymbol{x})$ from the training data. Note, that there is also discriminative classifiers that are learning a mapping function $f(\boldsymbol{x})$ which maps the input $\boldsymbol{x}$ directly onto a class label y.

The learning of a generative classifier is very demanding, since it involves to find the joint probability $p(\boldsymbol{x}, y)$. However, both models have their advantages [16, 124]: Compared with discriminative approaches, generative models typically (i) are able to handle missing features and unlabeled training data, (ii) do not have the necessity to be retrained when a new class is added, since parameters of each class conditional density are estimated independently and (iii) can readily handle compositionality. On the other hand, discriminative approaches (i) are expected to have better predictive performance, since they are directly trained to predict the class label, while generative approaches model the joint distribution instead, (ii) are very fast in predicting class labels for new test data and (iii) allow for arbitrary preprocessing of the data $\boldsymbol{x}$ with $\phi(\boldsymbol{x})$, while it is often hard to define a generative model for such pre-processed data.

2.1.3 Level of Supervision

The level of supervision used to train a recognition system is an important design decision. Commonly, the level of supervision in training is the same as the level of detail one can expect that the system gives as an output on unseen test images. If the amount of supervision in training is increased, while the output of the system is not getting more

Figure 2.2: Keypoint annotations for the category aeroplane.

detailed the learning complexity is reduced. On the other hand, if supervision information in training is lower than the level of detail provided on unseen images the complexity is increasing.

The most challenging task is to learn the hidden structure of unlabeled data, i.e. without any supervision. This problem is refereed to as unsupervised learning, in contrast to supervised learning, where different amounts of supervision are provided. In case, that the output granularity of a recognition system matches the amount of supervision provided in training, the amount of supervision is indirectly defining the task that is tackled by the recognition system. In the following, the three main forms of supervision information: (i) image label, (ii) bounding box annotation and (iii) pixel wise annotation are discussed. Figure 2.1 shows an example of all three on the level of objects. The smallest amount of supervision is to give a class label for an image. This is typically used to solve classification tasks, where the recognition system is providing a class label for each image. In case of classification of objects, it is usually also of large interest where the object is located in the image. In that case bounding box labels are provided for training and the recognition system is detecting objects, i.e. localizing and classifying objects in the image. The most detailed supervision information is provided by a pixel-labeling for each object which makes it possible to learn not only how to detect objects, but also provide a segmentation.

Part-based models provide a powerful framework for visual recognition and therefore, additionally to object annotations, part annotations have been introduced. Note, that there are also part-based approaches that are not utilizing any kind of part-supervision, but only object annotations (see Section 2.3.1). Approaches that use part supervision, e.g [6, 23, 32], are aiming to detect the object and predefined parts. Evaluation of these approaches is often performed on the level of objects, since comparison to other approaches is difficult because parts are often defined according to the requirements of the approach. Part supervision is typically beneficial for the object detection results, since disambiguities that might occur without supervision information are resolved. Part annotations are typically given in form of part bounding boxes [6, 32], however, other annotations are also possible, e.g. [23] suggested keypoint annotations (see Figure 2.2). These keypoints are used for the estimation of the 3D configuration, which allows to learn parts that have similar appearance and 3D object configuration. Furthermore, Chen et al. [32] recently provided even more detailed annotation by labeling individual part segmentations.

2.2 Basic Features for Object Representation

In order to represent images in a compact, generic representation, the raw pixel information needs to be transformed into a feature-based representation. The robust representation of complex objects turned out to be one of the key challenges of computer vision, and so, over the years, increasingly rich features have been proposed. Starting with brightness values of the image pixels and simple edge histograms [65] descriptors evolved and more sophisticated features were suggested. Until this day, the most powerful features capture the shape of an object, either by representing the contours (Section 2.2.1) or by edge orientation histograms (Section 2.2.2). Edges of objects are carrying important semantic information, since they are describing the boundaries of objects and therefore capturing their shape. Note, that there is also other kinds of descriptors that are not modeling shape but e.g. texture [176] or color [45].

Edge Extraction To find edges in an image it is preferable to detect edges using purely local information. Therefore, edges can be defined as locations of sudden intensity changes in an image. The gradient of an image points in the direction of the most rapid increase in intensity. Mathematically the gradient of an image is defined as

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right), \tag{2.3}$$

were the local gradient vector ∇I points in the direction of the steepest ascent in the intensity function. The gradient orientation

$$\theta = \tan^{-1}\left({}^{\frac{\partial I}{\partial y}} / {}_{\frac{\partial I}{\partial x}} \right) \tag{2.4}$$

points in the perpendicular direction of the local edge. While the magnitude of the gradient

$$\|\nabla I\| = \sqrt{\left(\frac{\partial I}{\partial x} \right)^2 + \left(\frac{\partial I}{\partial y} \right)^2} \tag{2.5}$$

is giving the strength of the intensity variation. Image derivatives can be approximated by applying finite difference filters to the image. However, such filters are strongly responding to noise, which makes it impossible to locate the edge. Similar to actual edges image noise is exhibiting high-frequency signal, as the pixel intensities are changing rapidly. To avoid such high-frequencies caused by noise one can smooth the image with a low-pass filter before the gradient computation. In most cases a Gaussian filter

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.6}$$

is used for smoothing due to its preferable properties: symmetry, separability, and circularity. The gradient of the smoothed image can be written as the convolution with the Gaussian derivative since convolution is associative:

$$\nabla \left[G_\sigma \star I \right] = \left[\nabla G_\sigma \right] \star I. \tag{2.7}$$

Furthermore, due to the separability of the Gaussian, one can convolve the image with the horizontal and vertical derivatives of the Gaussian kernel function.

The best known edge detector was suggested by Canny [27] who showed that the optimal smoothing filter can be well approximated by first-order derivatives of Gaussians. Furthermore, edge detection was enhanced by non-maximum suppression and hysteresis thresholding. Non-maximum suppression is a method for edge thinning, where lower edge responses in gradient direction are suppressed by the maximum value. Hysteresis thresholding is utilized to continue edges in the image. Therefore, a low and a high threshold are applied. The high threshold identifies strong edges, while the low threshold allows to continue these edges even if the continuing contour has no strong response, but is above the low threshold. More recent edge detection approaches also take into account color and texture cues and make use of learning techniques [119, 114, 42]. In [119] a supervised learning approach utilizing a large dataset of human labeled boundaries was suggested. To learn the boundaries a simple logistic regression classifier is used, which provides the predicted strength of an edge. Boosted Edge Learning (BEL) [42] is a supervised learning algorithms for edge detection which is utilizing low-level, mid-level and context information for each decision. Mairal et al. [114] devised a discriminative framework learned on sparse representations for class specific edge detection.

2.2.1 Contour Based Representations

Contours Objects can be directly represented by their contour obtained from edge extraction of training images or by a given template. Such representation is very accurate, however, direct comparison between template and extracted edges from an image is prone to errors, since variations will cause maps not to agree precisely. Therefore, comparison is more robust when measurement is based on proximity rather than exact superposition. In order to quickly compute the distance to a curve or set of points one can use two pass raster algorithm to compute the distance transform [143, 20, 39]. The distance transform of a binary edge image defines the distance of each pixel to the nearest non-zero pixel. For 2D relative translation $\boldsymbol{x}$ the distance transform $DT_Q(\boldsymbol{x})$ of a query edge map $Q = \{\boldsymbol{q}_j\}$ is defined as

$$DT_Q(\boldsymbol{x}) = \min_{\boldsymbol{q}_j \in Q} \quad d(\boldsymbol{x} - \boldsymbol{q}_j), \tag{2.8}$$

where $d(\boldsymbol{x})$ is some distance metric between pixel offsets. Two commonly used metrics include the city block distance

$$d(\boldsymbol{x}) = d(x_1, x_2) = |x_1| + |x_2| \tag{2.9}$$

and the Euclidean distance

$$d(\boldsymbol{x}) = d(x_1, x_2) = \sqrt{x_1^2 + x_2^2} = |\boldsymbol{x}| \tag{2.10}$$

Different matching approaches such as Chamfer [7] and Hausdorff [85] make use of distance transforms in comparing binary images. Furthermore, the so called generalized distance transformation [55] is extending standard distance transformation to non-binary-valued functions.

Shape Context The shape context descriptor suggested by Belongie et al. [8, 9, 11] is describing the coarse arrangement of a shape with respect to a point inside or on the object boundary. First, similar to the direct contour representation, edges are extracted from the image. Next, a relatively small number of sample points N are selected from these edges. Note, that these points do not correspond to special keypoints fulfilling special characteristics, such as high curvature or inflection points. For each sampled point, vectors to all other sampling points are computed. The set of vectors is a rich description of the given object shape, since the representation of shape becomes exact as N gets large. To make the representation more compact a histogram of the relative coordinates of the remaining points is computed, i.e. the angle of the vectors relative to the positive x-axis. To emphasize differences between nearby pixels a log-polar coordinate system is used. The number of log-radius bins and angle bins can vary according to the current application. For object recognition typically five log-polar bins and twelve angle bins are used.

Geometric Blur The geometric blur descriptor was first suggested by Berg and Malik [14]. This descriptor is a smoothed version of the signal around a feature point, blurred by a spatially varying kernel. The signal is typically assumed to be an edge image and the blur is accounting for the geometric distortion of a shape. In [13] a subsampled version of the original geometric blur descriptor was suggested for object recognition. Edge detectors are used to produce four channels of oriented edge responses and are smoothed using a Gaussian kernel. Since the geometric blur of a signal is usually rather smooth far from a feature point the descriptor can be subsampled at a sparse set of points. The final descriptor is a concatenation of the subsampled geometric blur descriptor computed at a certain point in each of the four orientation channels.

2.2.2 Histograms of Oriented Gradients

Scale-invariant feature transform (SIFT) The scale-invariant feature transform descriptor was suggested by Lowe [110, 111]. First, a Gaussian filter is applied to remove noise from the image. Then, key locations in scale space are identified that are invariant with respect to image translation, scaling, and rotation. This can be done by finding maxima or minima of a difference-of-Gaussian function applied in scale space. For efficient computation an image pyramid is build with resampling between each level. This method is particularly stable for characterizing the image, since it locates keypoints at regions and scales of high variation. At each of this keypoints a feature vector is extracted that describes the local image region sampled relative to its scale-space coordinate frame. Particularly, the image gradient magnitudes and orientations are sampled around the keypoint location at the scale of the keypoint. In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the keypoint orientation. After weighting the gradients with a Gaussian functions they are accumulated into orientation histograms of 4×4 subregions with 8 orientation bins. To avoid boundary effects, trilinear interpolation is used to propagate the value of each gradient sample into adjacent histogram bins. Finally, the vector is normalized

to unit length to make it less invariant to affine illumination changes and large gradient magnitudes in the feature vector are thresholded and renormalized.

Histograms of Oriented Gradients (HOG) Histograms of Oriented Gradients (HOG) where first suggested by Dalal and Triggs [37] for human detection and were extended by [56] for general object detection. For determining the HOG descriptor the object window is divided into small spatial regions called cells and for each cell a local histogram of gradient directions is computed. Results presented in [37] showed that the simple centered 1-D $[-1, 0, 1]$ mask and its transpose without smoothing are the preferable method for the gradient computation to build HOG features. After computing the gradient orientation, histograms using 9 orientation bins are computed over a 8×8 pixel sized cells. Each pixel in a cell is contributing to its corresponding histogram with a vote according to the strength of edge magnitude. To reduce aliasing, votes are interpolated bilinearly between the neighboring bin centers in both orientation and location, i.e. each pixel is also contributing to its neighboring bins in orientation and location. To ensure invariance to illumination changes a contrast normalization is performed using overlapping blocks, which consist of 2×2 cells. Each block is normalized separately and the HOG descriptor is concatenating each cell normalized with each of the blocks it belongs to. Therefore, each cell contributes several components to the final feature vector. For block normalization an L2 normalization or an L2 normalization with additional truncation, to limit the maximal value used in the final HOG descriptor, is applied.

When applying HOG to general object detection it was found in [56] that some object categories improve using contrast sensitive features, while others benefit from contrast insensitive features as suggested by [37]. For this reason the HOG descriptor was extended to compute histograms of gradients using 9 bins, called contrast insensitive, and histograms of gradients using 18 bins, called contrast sensitive. Normalization is performed as in [37] using overlapping blocks. However, instead of using each of the normalized cell values separately in the feature vector the values are summed up and additionally 4 dimensions capturing the overall gradient energy are added.

2.3 Object Modeling Schemes

For recognition on the category level one can distinguish between holistic and part-based object models. The next Section will give a brief overview over both types of object models and introduce the concept of hierarchical object models, as mid-level representations will be discussed for these different object modeling schemes in Section 2.4 and subsequent chapters of this thesis.

2.3.1 Holistic and Part-Based Object Models

Holistic object models describe the whole object in a single model which can handle all variabilities or utilize different holistic object models to describe possible poses of an object. In [131] a general object detector is presented using a support vector machine

trained on an overcomplete dictionary of Haar wavelets. Viola et al. [170] suggested a holistic pedestrian detection system using AdaBoost to train a chain of progressively more complex region rejection rules based on Haar-like wavelets and space-time differences. Dalal and Triggs [37] suggested a holistic pedestrian detection system with a rather simple architecture that significantly outperformed early person detectors. Gavrila [68] suggested a real-time pedestrian detection system using a template tree to efficiently represent and match the variety of shape exemplars. Besides this advances in pedestrian detection using holistic object models also more generic object detections systems have been suggested. Deselaers and Ferrari [40] presented a global self-similarity descriptor for object detection and classification. Furthermore, shape based template matching was extended to fast directional chamfer matching in [108] and applied for object detection.

To overcome the limitations of holistic object models when dealing with articulated objects, part-based models have been suggested. The first part-based model was suggested by [64] for face recognition. The suggested model is linking humanly defined parts, i.e. eyes, head, nose, mouth, left, and right edge of the face, based on a flexible spatial arrangement. Part-based models are ranging from fully connected shape models, where all parts are connected directly to each other, e.g. constellation model [54, 59], to non-connected part models, where all parts are mutually independent, e.g. Bag of Features [36, 165, 166]. Between this two extreme cases a lot of different connectivity structures have been developed, such as star shape models [35, 60, 63, 103], k-fan models [35] and tree structures [58, 188]. In a star shaped model all parts are connected to one center part. K-fan models are a generalization of the star shaped model to k center parts, i.e. all parts are connected to the k center parts. Tree structured models are connecting each part with all other parts by a single path. Due to the different connectivity structures these different methods vary greatly in their computational complexity. Assuming that N is the number of feature detections and P is the number of parts, the constellation model has an exponential complexity $O(N^P)$, while the Bag of Features model only has a complexity of $O(NP)$. Therefore, recent approaches favor star shaped models [56, 63] and Bag of Features [165, 166], typically in combination with a spatial pyramid [100], due to their efficiency. One of the most important part-based object models these days is the Deformable Part Model (DPM) [56]. The DPM is a star shaped model, where object parts are positioned relative to the center of the whole object. The exact position of the parts is flexible and determined by latent variables. Its success has drawn attention from the entire vision community towards this tool, and subsequently, it has become an integral component of many classification and segmentation tasks. In 2010 it received the lifetime achievement award at the PASCAL VOC challenge.

2.3.2 Hierarchical Object Models

Hierarchical object models are biologically inspired by the visual cortex of the brain, which is responsible for processing visual information. Especially, Convolutional Neural Networks (CNNs), such as the Neocognitron [66], HMAX [150] and LeNet-5 [102] are emulating the behavior of the biological processes. While the first convolutional network, the Neocognitron [66], was lacking a supervised learning algorithms, LeCun et al. [101] extended the model and showed that stochastic gradient descent via backpropagation

was effective for training convolutional networks. Currently, convolutional networks are becoming widely used again, due to the availability of extremely large datasets containing 1.2 million labeled images and the usage of highly-optimized GPU implementations for 2D convolution. The popular CNN suggested by Krizhevsky et al. [97] consists of five convolutional and three fully-connected layers and uses few adaptations on the CNN of LeCun et al. [102], such as rectifying non-linearities and dropout regularization. While convolutional networks coined the term *deep learning*, also other concepts exploit hierarchies for recognition.

One of them are *compositional hierarchies* [61, 88, 96, 127, 139] that establish one or more successive representational layers by grouping parts, thus obtaining a hierarchy of successively larger and more meaningful compositions. Compared to convolutional networks, where the information from a cube-like receptive field over lower-layer feature responses is conveyed in only one value, compositional architectures are represented as a graph in which each node has only a small number of incident descendants. Therefore, it is easier to trace back what has caused the response in a compositional hierarchy than in a convolutional network. In addition, this makes inference in compositional hierarchies more controlled and structured. On the other hand, convolutional networks have the ability to automatically learn hidden representations in the network, while a lot of compositional hierarchies lack automation and rely on hand-crafted features and compositional rules. However, this drawback can be overcome by unsupervised bottom up learning strategies.

Moreover, shallow hierarchies are employed to represent spatial dependencies between model parts. In [56] a two layer hierarchy was introduced which utilized a hidden layer to represent the best matching location for each part according to the position of the object hypothesis. An extension of this model was suggested by Zhu et al. [188], which introduced an incremental concave-convex procedure, which allows to make learning of two and three layer models efficient.

2.4 Mid-Level Object Representations

Recently, a growing number of approaches are suggesting that using only very low-level features is insufficient to solve object recognition. Therefore, instead these approaches suggest the use of mid-level features to capture higher-level concepts. Such mid-level representations bridge the gap between low-level feature representations (see Section 2.2) and complex objects or scenes and, therefore, improve recognition. In the following an overview is given on several kinds of mid-level representations.

2.4.1 Higher-Order Statistics

A popular kind of descriptor utilizing higher-order statistics are *co-occurrence histograms*. Co-occurrence histograms have been suggested for different kinds of features, such as color [29, 86], orientation gradients [177, 140, 86], and texture [149, 86]. Co-occurrence histograms are counting the occurrence of pairs of values instead of the occurrence of individual values. Capturing pairs of values leads to a larger and more

detailed description of objects and, therefore, has more expressive power than standard histograms. Besides the usage of co-occurrence histograms, also other realizations of co-occurrences have been suggested. One of them is presented in [185], were a higher-level lexicon, called visual phrase lexicon, is generated by combining meaningful spatially co-occurent patterns of visual words. Such higher-level lexicon is much less ambiguous than the lower-level one. In [121], an extension of Viola and Jones popular face detector [169] is suggested. The face detector of Viola and Jones is using rectangular filters similar to Haar wavelets trained with an AdaBoost classifier that is selecting the best filters in every boosting round. The object detection framework of Mita et al. [121] incorporates the co-occurrence of multiple features at each stage of the boosting process.

Another concept of capturing higher-order statistics in images is *self-similarity*. It was introduced by Shechtman and Irani [151] to recognize correspondences despite the lack of a common underlying visual property, i.e. pixel colors, intensities, edges, gradients, or other filter responses. In their approach, self-similarities are measured locally by computing the correlation between an image patch and a larger surrounding image region. This descriptor has been adopted in the object detection and classification community[18, 30, 84, 99, 168]. While the original work matches ensembles of these descriptors, most later works use it as a feature in a bag-of-words framework. In [30, 84] the local self-similarity descriptor of Shechtman and Irani [151] is applied for image retrieval. Furthermore, the descriptor was applied for object classification and detection [18, 168, 99]. While in [18] and [168] the descriptor was applied to learn object classes, [99] applied it to attribute transfer learning.

In [40], Deselaers and Ferrari explored global self-similarity and its advantages over local self-similarity. The suggested global self-similarity descriptor captures the spatial arrangements of self-similarities within the entire image. While in [40] correlations are measured between an image patch and the entire image, Walk et al. [173] suggested a self-similarity measure between different subregions of an image. This kind of self-similarity is capturing similarities across the whole image, however, it restricts self-similarity to always a small local region.

2.4.2 Attributes

Visual attributes are human-nameable mid-level semantic properties. Attributes are typically learned in a fully-supervised way, by training a discriminative classifier for each attribute from images labeled by the attributes [25, 53, 99]. Attributes shift the goal of recognition from naming to describing. Farhadi et al. [53] suggested semantic and discriminative attributes to overcome limitations of standard recognition paradigm of naming. Instead of just naming an object by its category name, using attributes allows also to report unusual aspects of familiar objects, describe unfamiliar objects, and learn how to recognize new objects with few or no sample images. The problem of object classification, when no training examples of the target classes are available, was explored by Lampert et al. [99]. In [52] objects are described by the spatial arrangement of their attributes and the interactions between them. The system groups objects within broad domains, such as “animal” and “vehicle” instead of learning each category separately. Contrary to this more general detection, Duan et al. [43] propose

to use attributes for fine-grained object recognition. The system discovers candidate attributes that are detectable and discriminative. In order to give semantic names to the attributes human interaction is used. This way the system discovers discriminative local attributes that are both machine-detectable and human-understandable. Yuan et al. [186] proposed an efficient data-mining based approach to discover discriminative co-occurrences of attributes. Jayaraman [87] proposed a multi-task attribute learning approach that encourages each attribute classifier to use a disjoint set of image features to make its predictions. Whereas other models train each attribute classifier independently, and therefore are prone to re-using image features for correlated attributes, this approach aims to isolate distinct low-level features for distinct properties.

2.4.3 Mid-Level Patches and Parts

The learning of parts is usually integrated into the learning of a complete object or scene model. In contrast to that mid-level patches and parts are individually trained discriminative classifiers. The first approach in the direction of mid-level representation are poselets [23]. A poselet describes a particular part of the object pose under a given viewpoint. It is defined with a set of examples that are close in 3D configuration space. In order to find such examples the approach relies on additional keypoint annotations, which are providing information about the 3D configuration of an object. An example for such keypoints is given in Section 2.1.3.

Despite the highly supervised poselet approach there exist several approaches on unsupervised or weakly supervised discovery of mid-level patches. One of them is the work of Singh et al. [153], which also coined the term *mid-level patches* by their suggestion of mid-level visual primitives, which are more adaptable to the appearance distributions in the real world than the low-level features, but do not require the semantic grounding of high-level entities. Contrary to the concept of attributes (Section 2.4.2) and poselets [23, 22] the suggested mid-level patches do not explicitly aim to discover whole semantic units from labeled training data. Instead, the aim of [153] is to find mid-level patches, in an unsupervised manner, which are defined by their representative and discriminative property, i.e. that they can be detected in a large number of images with high recall and precision. On top of this idea, two other very recent approaches, utilizing mid-level parts, have been suggested for object recognition [46] and scene classification [89]. Contrary to [153], these approaches are not fully unsupervised but weakly supervised using object bounding boxes and image labels respectively. Furthermore, both approaches are utilizing exemplar support vector machines (ESVM) [117, 79] to initialize part classifiers, which are then utilized to find more positive instances and retrain the model, instead of the iterative procedure of [153], which alternates between clustering and training discriminative classifiers.

2.4.4 Compositions

The fundamental goal of compositional hierarchies is to establish one or more successive representational layers by grouping parts thus obtaining a hierarchy of successively larger

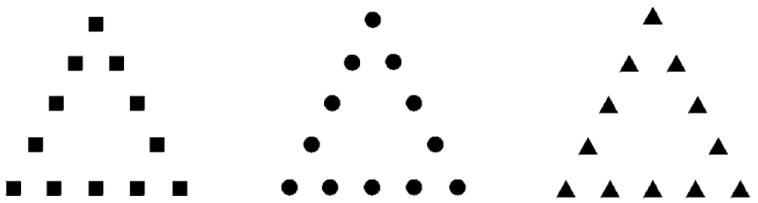


Figure 2.3: Triangle configuration made up of different basic elements. Triangle property is emerging independent of the individual parts [72].

and more meaningful compositions. This concept of compositionality is contrary to part-based models (see Section 2.3.1) which are directly modeling relations between individual parts. The fundamental idea of grouping parts into more meaningful compositions has its origin in the gestalt psychology [178, 179]. One of the most important concepts of gestalt psychology is that relations between parts are essential for the meaning of a composition. This lead to the well known statement of Wertheimer: "the whole is different form the sum of its parts" [129, p. 50]. In order to support this claim, examples with emergent properties have been created. This means that properties are not shared between the local parts and the whole gestalt. An example of this concept is given in Figure 2.3. Although the configurations are formed from different basic elements, i.e. squares, circles, and triangles, all three are exhibiting the triangle property. The fact, that the last configuration is made up of triangles doesn't make it more or less a triangle than the ones made up of squares and circles. The triangle property emerges independent of the property of the individual parts. The idea of such emergent properties was also acknowledged later by Biedermann [15] who proposed the recognition by component theory to explain object recognition. According to this theory recognition is a multi stage process in which the object is segmented into basic components which can be approximated by so called geons, i.e. simple forms such as cylinders and cones.

This general concept of compositionality has been pursued by many object recognition systems over the years [88, 128, 96, 61, 2]. Early compositional approaches for object recognition are typically decomposing the whole object into its constituents in a top-down manner [47, 105, 127]. Contrary to this [162] suggested to parse images into their constituent visual patterns, by combining top-down and bottom-up inference. In [96] object categories are decomposed into parts and shape contours using a top-down approach. In order to learn shape models discriminatively, they employ a Multiple Instance Learning algorithm in a bottom-up approach. Moreover, compositional approaches are dividing into appearance based approaches [88, 127, 128] and shape based approaches [62, 61, 2]. Jin and Geman [88] present a compositional architecture with manually built structure for license plate reading. Opposed to such hand build structures, Ommer and Buhmann [127, 128] described a composition system that automatically learns structured, hierarchical object representations in an unsupervised manner. Compositions are modeled as bags of parts with locality constraints and intermediate compositions are learned in a generative framework yielding relevant part agglomerations. Along this lines of unsupervised structures Fidler et al. [62] suggested a Learned Hierarchy of Parts (LHOP), for compositional representation of parts. While

lower layers are learned independently from the category, higher levels are specific to the category. Following this basic approach, they suggested to speed-up recognition by using a generative taxonomy of constellation object detectors in [61]. Recently, Aktas et al. [2] proposed Compositional Hierarchy of Parts (CHOP). In this approach graph theoretic tools are used to analyze, measure and employ geometric and statistical properties of parts to infer compositions.

CHAPTER 3

VISUAL RECOGNITION USING EMBEDDED FEATURE SELECTION FOR CURVATURE SELF-SIMILARITY

This chapter explores, how mid-level representations can improve object detection systems that are relying on holistic object models. Many current detection systems are based on holistic object representations learned with a discriminative classifier. The silhouette of an object is one of the most important cues for recognition and therefore most powerful object representations are capturing the shape of an object. In particular, the currently most widely used and best performing image descriptors model objects based on edge orientation histograms (Section 2.2.2). However, as described in Section 2.4 solely using low-level features is not sufficient to solve object recognition. To overcome the limitations of low-level descriptors, recently more complicated image statistics like co-occurrence and self-similarity (Section 2.4.1) became more and more popular to build more robust descriptors. While it was shown that self-similarity and co-occurrence lead to very robust and highly discriminative object representations, these second order image statistics are also pushing feature spaces to extremely high dimensions. Since the amount of training data stays more or less the same, such high dimensionality can cause the system to suffer from curse of dimensionality and overfitting. Furthermore it is noticeable, that descriptors based on edge orientation histograms are approximating objects with straight lines. However, it was shown in different studies within the perception community, that besides orientation, also curvature is an important cue when performing visual search tasks.

In [122] the modeling of object boundary contours was extended beyond the widely used edge orientation histograms by utilizing curvature information, to overcome the drawbacks of straight line approximations. However, curvature can provide even more information about the object boundary. By computing co-occurrences between discriminatively curved boundaries we build a curvature self-similarity descriptor that

provides a more detailed and accurate object description. To exploit the full capabilities of high-dimensional representations applied in object detection we developed a new embedded feature selection method for SVM, which reliable discards superfluous dimensions and therefore improves object detection performance. Figure 3.1 gives an overview over the training process of the detection system.

The remaining of the chapter is organized as follows: First, a short overview is given on regularized risk minimization and SVM (Section 3.1). Next, feature selection methods (Section 3.2) are reviewed and a novel method to capture the important dimensions from high-dimensional representations (Section 3.2.2) is described. After that, histograms of curvature [122] are introduced and a new self-similarity descriptor based on curvature is suggested to go beyond the straight line approximation of objects (Section 3.3). Moreover, Section 3.3 discusses previous work on self-similarity. In the Section 3.4, at the end of the chapter, our novel curvature self-similarity descriptor is evaluated along with the suggested feature selection method.

3.1 Regularized Risk Minimization

This section reviews the concept of regularized risk minimization [167]. It was shown that a large number of machine learning problems can be formalized as regularized risk minimization problems of the form:

$$\min_{\mathbf{w}} J(\mathbf{w}) := \lambda\Omega(\mathbf{w}) + R(\mathbf{w}). \tag{3.1}$$

where $R(\mathbf{w})$ is the empirical risk

$$R(\mathbf{w}) := \frac{1}{N}\sum_{i=1}^{N} l(y_i, f(\mathbf{x}_i)). \tag{3.2}$$

Moreover, $\{(\mathbf{x}_i, y_i)\}_{1\leq i\leq N}$ are the training data with labels $y_i \in \{-1, +1\}$ and $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^n$. l is a loss function that measures the difference between the label y_i and the prediction arising from the prediction function

$$f(\mathbf{x}) := \mathbf{w}^T\psi(\mathbf{x}) + b. \tag{3.3}$$

The function ψ represents a mapping function $\psi : \mathbb{R}^n \rightarrow \mathbb{R}^{n'}$ of $\mathbf{x}$ into a higher dimensional space $\mathbb{R}^{n'}$.

The function $\Omega(\mathbf{w})$ is serving as a regularizer, weighted by the regularization constant $\lambda > 0$. One of the most common regularizers is the squared L_2 norm:

$$\Omega(\mathbf{w}) := \|\mathbf{w}\|_2^2 = \mathbf{w}^T\mathbf{w} \tag{3.4}$$

This regularizer has gained popularity since the introduction of max-margin methods such as SVM, since $\|\mathbf{w}\|_2$ is inverse proportional to the margin. Another popular regularizer is the L_1 norm

$$\Omega(\mathbf{w}) := \|\mathbf{w}\|_1. \tag{3.5}$$

The L_1 regularization induces sparsity and is therefore preferably used in problems with high-dimensional input spaces as it automatically eliminates irrelevant dimensions.

Despite different regularizers Ω a multitude of loss functions exists. Here we will shortly review some well known loss functions that are commonly used in machine learning:

1. Hinge loss [12]

$$l(y, f(\mathbf{x})) := \max(0, 1 - yf(\mathbf{x})) \tag{3.6}$$

2. Logistic loss [33]

$$l(y, f(\mathbf{x})) := \log(1 + \exp(-yf(\mathbf{x}))) \tag{3.7}$$

3. L_2 or least mean squares loss [181]

$$l(y, f(\mathbf{x})) := (f(\mathbf{x}) - y)^2 \tag{3.8}$$

The combination of different regularizers and loss functions leads to several standard machine learning problems. For example, using an L_2 regularizer (Equation 3.4) and the hinge loss (Equation 3.6) leads to the well known Support Vector Machine (SVM) [21, 167]. Changing the loss function to the logistic loss function (Equation 3.7) leads to the problem known as logistic regression. The Lasso technique (Least Absolute shrinkage and Selection Operator) [159] can be recovered by utilizing an L_1 regularizer (Equation 3.5) with an L_2 loss function (Equation 3.8) . Exchanging the L_2 loss by the logistic loss (Equation 3.7) leads to the so called generalized LASSO [144].

3.1.1 Support Vector Machines

Support vector machines have been first introduced in [21] and is based on statistical learning theory of Vapnik [167]. Until now, it is one of the most popular supervised classification methods, which has been used in many object recognition approaches. An SVM classifier aims to find a hyperplane that best separates two classes, i.e. the hyperplane with the maximum distance between points in each class. The distance between the two classes is called margin and the SVM is maximizing it.

Finding the maximum margin between two classes corresponds to solving the optimization problem

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{s. t. :} \quad & y_i(\mathbf{w}^T\psi(\mathbf{x}_i) + b) \geq 1, \quad \forall i \in \{1, ..., N\}. \end{aligned} \tag{3.9}$$

As in many cases, this problem is given by means of a constrained optimization problem and the loss function is not explicitly defined. However, rewriting the problem without constraints leads to

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N} \max(0, 1 - y_i(\mathbf{w}^T\psi(\mathbf{x}_i) + b)). \tag{3.10}$$

Now one can see, that this problem is an instance of the regularized risk minimization problem, minimizing the hinge loss with an L_2 regularizer. This optimization problem is often converted into its dual form, which gives, in the case of support vector machines, the same solutions as the primal due to strong duality, i.e.

$$\min_{\mathbf{w},b} \max_{\alpha \geq 0} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \max_{\alpha \geq 0} \min_{\mathbf{w},b} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}). \tag{3.11}$$

The Lagrangian $\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})$ is given by

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i \left[y_i \left(\mathbf{w}^T \psi(\mathbf{x}_i) + b \right) - 1 \right]. \tag{3.12}$$

In the dual formulation the minimization problem needs to be solved first. The optimal $\mathbf{w}$ and b must satisfy the condition that the partial derivatives of $\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})$ with respect to $\mathbf{w}$ and b are zero.

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \psi(\mathbf{x}_i) = \mathbf{0} \tag{3.13}$$

implies that

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \psi(\mathbf{x}_i) = \mathbf{0}. \tag{3.14}$$

Similarly, the partial derivation with respect to b results in

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = \sum_{i=1}^{N} \alpha_i y_i = \mathbf{0}. \tag{3.15}$$

Substituting $\mathbf{w}$ in 3.12 by 3.14 results in

$$\begin{aligned} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = & \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \psi(\mathbf{x}_i)^T \psi(\mathbf{x}_j) - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \psi(\mathbf{x}_i)^T \psi(\mathbf{x}_j) \\ & - b \sum_{i=1}^{N} \alpha_i y_i + \sum_{i=1}^{N} \alpha_i \end{aligned} \tag{3.16}$$

Since $\sum_{i=1}^{N} \alpha_i y_i = \mathbf{0}$ this results in

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \psi(\mathbf{x}_i)^T \psi(\mathbf{x}_j). \tag{3.17}$$

Now, the functional only depends on the Lagrangian multipliers $\boldsymbol{\alpha}$. Moreover, one can observer, that the transformed feature vector $\psi(\mathbf{x}_i)$ is only appearing in dot products. Such dot products are commonly interpreted as kernels in the field of kernel machines [148].

Therefore applying the so called *kernel trick* [21] and exchanging the dot product by a kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \psi(\mathbf{x}_i)^T \psi(\mathbf{x}_j) \tag{3.18}$$

leads to the final dual optimization formulation

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j). \tag{3.19}$$

The training samples $\mathbf{x}_i$, for which corresponding Lagrangian multiplier α_i is larger than zero, are the *support vectors*. Solving Equation 3.19 leads to the SVM classifier

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{3.20}$$

In case of the linear kernel, one can use Equation 3.14 to write the decision function in Equation 3.20 as

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b = \mathbf{w}^T \mathbf{x} + b. \tag{3.21}$$

In 1995 Cortes and Vapnik [34] suggested a modification of the support vector machine that allows for violations of the margin to make it less sensitive to outliers. The so called soft margin SVM is alleviating the problem of outliers, that make it impossible to find a hyperplane that can separate all training samples perfectly, i.e. without margin violations. The soft margin SVM allows samples to be misclassified, i.e. to lie inside the margin or even on the wrong side of the margin and finds a hyperplane that leads to the minimal number of margin violations. To relax the optimization problem in such a way, slack variables ξ_i are introduced for each training sample $\mathbf{x}_i$. The SVM problem given in Equation 3.9 can be written as

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{i=1}^{N} \xi_i \\ \text{s. t. :} \quad & y_i(\mathbf{w}^T \psi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \forall i \in \{1, ..., N\} \\ & \xi_i \geq 0, \qquad\qquad\qquad\quad \forall i \in \{1, ..., N\} \end{aligned}$$

for a given constant C. C is a free parameter that controls the relative importance of minimizing the norm of $\mathbf{w}$ (which is equivalent to maximizing the margin) and satisfying the margin constraint for each training sample. In the case that C is small, more violations of the margin are allowed and the margin is therefore getting larger. In case that C is very large less margin violations are allowed and the margin becomes smaller. Similar to the original SVM formulation, given in Equation 3.9, strong duality (Equation 3.11) also holds for the soft-margin SVM. Conversion of the primal soft margin formulation into the

dual formulation is done similarly to hard margin SVM using the Lagrangian. The dual of the soft margin SVM is

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} : \quad & 0 \leq \alpha_i \leq C \quad \forall i. \end{aligned} \tag{3.22}$$

Note, that this is a quite similar formulation has the hard margin SVM given in Equation 3.19, but with one additional constraint on the Lagrange multipliers α_i.

3.2 Feature Selection

Guyon et al. [76] categorize feature selection methods into filters, wrappers and embedded methods. In this section we shortly describe the differences between these three approaches, review the state-of-the-art of embedded feature selection for SVMs (Section 3.2.1) and suggest a new approach for iterative dimensionality reduction for SVMs (Section 3.2.2).

Filters Filter methods are applied in a preprocessing step and are independent of the classifier. Many filter methods are so called variable ranking methods [76, 95]. Instead of providing a fixed subset of useful features, they are providing a ranking of all feature dimensions based on their relevance. A popular variable ranking method are correlation based criteria, such as the square of Pearsons correlation coefficient [133], which measures the linear dependency between two variables. This criterion has been shown to be closely related to the Fisher's criterion [67] and the T-test [73]. Besides correlation based criteria, other variable ranking methods exist, e.g. the Relief algorithm suggested by Kira et al. [94]. The Relief algorithm is utilizing the nearest neighbors to compute the rank of a variable. For each variable it first finds its nearest hit, i.e. closest example in the same class, and its nearest miss, i.e. closes example in a different class. The ranking is given by the difference between the nearest hit and the nearest miss of the feature. Therefore, the weight of a feature becomes low, if the distance to its nearest miss is smaller than the distance to its closest hit.
After ranking the individual feature dimensions, the final selection of a subset is done by choosing a threshold e.g. using cross-validation. The advantage of filters is that they are simple and efficient to compute, as it only requires to compute n scores and sorting them. Furthermore, filters are robust against overfitting. However, since feature selection is performed completely independent of the classifier the selected subset might not be suitable for the classifier.

Wrappers Contrary to filters, wrappers [76, 95] are not ranking variables according to their individual predictive power, but use the classifier to score subsets of variables according to their combined predictive power. Wrapper methods are treating the classifier as a black box, i.e. the classifier is applied without any knowledge about the classification

process and only the output is utilized for the feature selection. Therefore, the selection process is not independent of the classifier anymore, as it is the case for filter methods. Popular classifiers used to apply wrapper feature selection methods are decision trees, naive Bayes and support vector machines [76]. The feature selection itself turns out to be a search problem. If the number of variables is small an exhaustive search can be used. However, in most of the cases more efficient strategies are needed, such as best-first, branch-and-bound, simulated annealing, and genetic algorithms [95]. Greedy search methods are often preferred due to their computational efficiency and turned out to be robust against overfitting in practice [141]. One can distinguish two main strategies: forward selection and backward elimination [76]. Forward selection is greedily adding feature dimensions based on the performance of the classifier, while backward elimination is rejecting the least useful feature dimensions.

Embedded Methods Contrary to filters and wrappers, embedded feature selection methods incorporate feature selection as a part of the learning process [76, 98]. In [98] a unified framework is defined that covers many embedded methods. Furthermore, embedded methods are discussed based on how they solve the feature selection problem. A common method is to iteratively add or remove features and greedily approximate a solution. This idea is similar to those of wrapper methods, however in the case of embedded methods, the classifier and its parameters are not a black box and information about the classifier can be used in the selection process. Similar to wrapper methods there exist embedded methods that are utilizing forward selection [135, 44, Section 8.3.2] and backward elimination [77, 138]. Another approach is to relax the feature selection problem from the binary case to the continuous case and solve the feature selection problem as optimization of scaling factors [180, 112]. Furthermore, in case of linear models, the feature selection can be enforced directly on the model parameters [24, 159].

3.2.1 State-of-the-Art Embedded Feature Selection for SVMs

Since most state-of-the-art detection systems use SVM as a classifier, the focus of this work is on embedded feature selection methods for SVMs. To directly integrate feature selection into the learning process of SVMs, sparsity can be enforced on the model parameter $\mathbf{w}$. Several researchers, e.g [24], have considered replacing the L2 regularization term $\|\mathbf{w}\|_2^2$ with an L1 regularization term $\|\mathbf{w}\|_1$. Since L1 norm penalty for SVM has some serious limitations [189], Wang et al. [175] suggested the doubly regularized SVM (DrSVM), which is not replacing the L2 regularization, but adding an additional L1 regularization to automatically select dimensions during the learning process.

Contrary to linear SVM, enforcing sparsity on the model parameter $\mathbf{w}$ does reduce dimensionality for non-linear kernel functions in the higher dimensional kernel space rather than in the number of input features. To reduce the dimensionality for non-linear SVMs in the feature space one can introduce an additional selection vector $\boldsymbol{\theta} \in [0,1]^n$, where larger values of θ_i indicate more useful features. The objective is then to find the

best kernel of the form

$$K_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = K(\boldsymbol{\theta} * \mathbf{x}, \boldsymbol{\theta} * \mathbf{z}), \tag{3.23}$$

where $\mathbf{x}, \mathbf{z} \in \mathbb{R}^n$ are the feature vectors and $*$ is element-wise multiplication. These hyper-parameters $\boldsymbol{\theta}$ can be obtained via gradient descent on a generalization bound or a validation error. Another possibility is to consider the scaling factors $\boldsymbol{\theta}$ as parameters of the learning algorithm [74], where the problem was solved using a reduced conjugate gradient technique.

In this work, the scaling factors are integrated into the learning algorithm, but instead of using L2 norm constraint, like in [74], on the scaling parameter $\boldsymbol{\theta}$, an L1 norm sparsity which is explicitly discarding dimensions of the input feature vector is applied. For the linear case, the optimization problem becomes similar to DrSVM [175] where a gradient descent method is applied to find the optimal solution $\mathbf{w}^*$. To find a starting point, a computational costly initialization is applied in [175], while our selection step can start at the canonical $\boldsymbol{\theta} = \mathbf{1}$, because $\mathbf{w}$ is modeled in a separate variable.

3.2.2 Iterative Dimensionality Reduction for SVM

We are following the concept of embedded feature selection and therefore include the feature selection parameter $\boldsymbol{\theta}$ directly in the SVM classifier. The corresponding optimization problem can be expressed in the following way:

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{subject to :} \quad y_i(\mathbf{w}^T\psi(\boldsymbol{\theta} * \mathbf{x}_i) + b) \geq 1 - \xi_i \quad \wedge \quad \xi_i \geq 0 \quad \wedge \quad \|\boldsymbol{\theta}\|_1 \leq \theta_0 \end{aligned} \tag{3.24}$$

where $K(\mathbf{x}, \mathbf{z}) := \psi(\mathbf{x}) \cdot \psi(\mathbf{z})$ is the SVM kernel function. The function $\psi(\mathbf{x})$ represents the mapping of the feature vector $\mathbf{x}$ into a higher dimensional space. As discussed in Section 3.1.1 a SVM classifier is learning a hyperplane defined by $\mathbf{w}$ and b which best separates the training data $\{(\mathbf{x}_i, y_i)\}_{1 \leq i \leq N}$ with labels $y_i \in \{-1, +1\}$. We enforce sparsity of the feature selection parameter $\boldsymbol{\theta}$ by the last constraint of Equation 3.24, which restricts the L1-norm of $\boldsymbol{\theta}$ by a constant θ_0. Since SVM uses L2 normalization it does not explicitly enforce single dimensions to be exactly zero. However, this is necessary to explicitly discard unnecessary dimensions. We rewrite the problem in Equation 3.24 without additional constraints in the following way:

$$\min_{\boldsymbol{\theta}} \min_{\mathbf{w},b} \lambda\|\boldsymbol{\theta}\|_1 + \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\max(0, 1 - y_i f_{\boldsymbol{\theta}}(\mathbf{x}_i)) \tag{3.25}$$

where the decision function $f_{\boldsymbol{\theta}}$ is given by $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{w}^T\psi(\boldsymbol{\theta} * \mathbf{x}) + b$. Note, that the last constraint, where the L1-norm is restricted by a constant θ_0 is rewritten as an L1-regularization term, multiplied with the sparsity parameter λ.

Due to the complexity of problem 3.25 we propose to solve two simpler problems iteratively. We first split the training data into three sets, training $\{(\mathbf{x}'_i, y'_i)\}_{1 \leq i \leq N'}$,

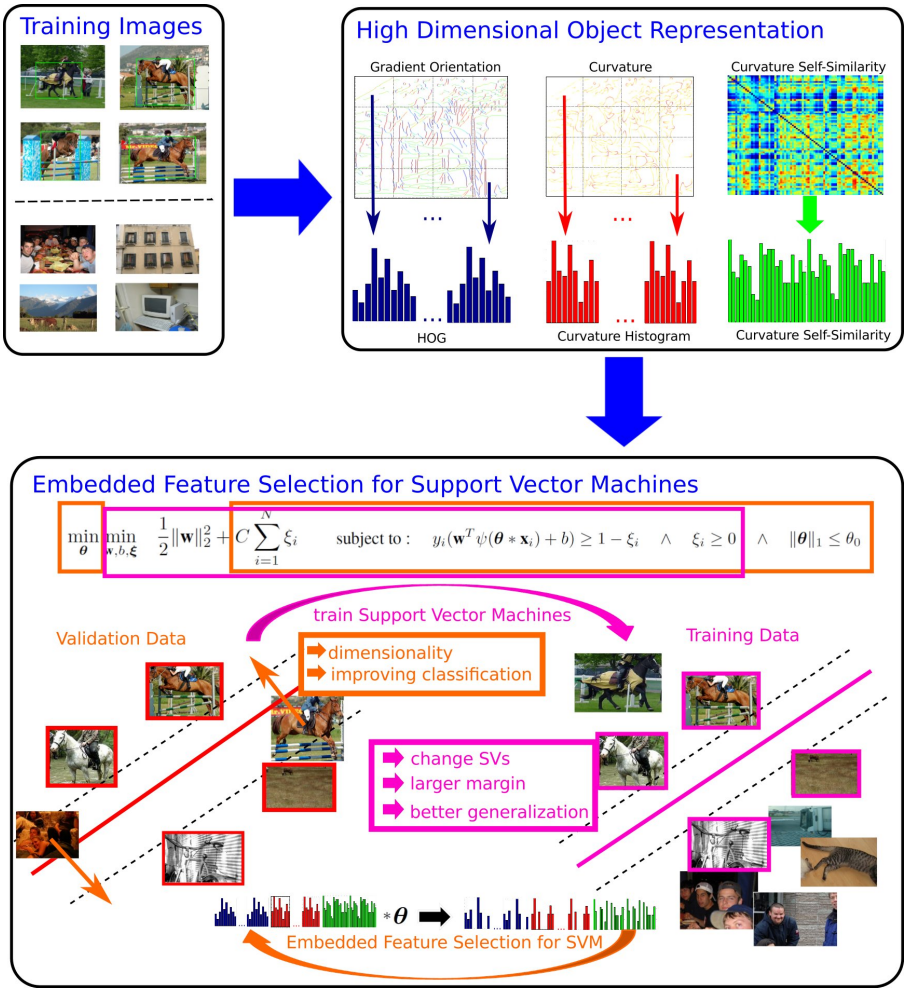


Figure 3.1: After extracting high-dimensional feature representation using HOG, curvature histograms and curvature self-similarity the optimization process is split in two steps due to its complexity. First, an SVM classifier is trained on the training data. The resulting classifier can be applied on the unseen validation data and the selection parameter θ is optimized so that the separation on the validation data is improved.

validation $\{(\mathbf{x}_i'', y_i'')\}_{1 \leq i \leq N''}$ and a hold out testset. Now, we optimize the problem according to $\mathbf{w}$ and b for a fixed selection parameter $\boldsymbol{\theta}$ using a standard SVM algorithm on the training set. Parameter $\boldsymbol{\theta}$ is optimized in a second optimization step on the validation data using an extended version of the bundle method suggested in [41]. We are performing the second step of our algorithm on a separate validation set to prevent overfitting. The optimization process is shown in Figure 3.1.

In the first step of our algorithm, the parameter $\boldsymbol{\theta}$ is fixed and the remaining problem is

Algorithm 3.1 Iterative Dimensionality Reduction for SVM.

```
converged := FALSE
θ := 1
while converged==FALSE do
    [x'_l , α , b] = trainSVM( X', Y', θ, C)
    θ* = applyBundleMethod(X'',Y'',x'_l,α,b,C)
    if θ* == θ then
        converged=TRUE;
    end if
    θ = θ*
end while
```

converted into the dual problem

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N'} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N'} \alpha_i \alpha_j y'_i y'_j K(\boldsymbol{\theta} * \mathbf{x}'_i, \boldsymbol{\theta} * \mathbf{x}'_j) \tag{3.26}$$

$$\text{subject to :} \quad 0 \leq \alpha_i \leq C, \qquad \sum_{i=1}^{N'} \alpha_i y'_i = 0$$

where the decision function $f_{\boldsymbol{\theta}}$ is given by $f_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{l=1}^{m} \alpha_l y_l K(\boldsymbol{\theta} * \mathbf{x}, \boldsymbol{\theta} * \mathbf{x}'_l) + b$, where m is the number of support vectors. Equation 3.26 is solved using a standard SVM algorithm [28, 115]. The optimization of the selection parameter $\boldsymbol{\theta}$ starts at the canonical solution where all dimensions are set to one. This is corresponding to the solution that is usually taken as a final model in other approaches. In our approach we apply a second optimization step to explicitly eliminate dimensions which are not necessary to classify data from the validation set. Fixing the values of the Lagrange multipliers $\boldsymbol{\alpha}$, the support vectors $\mathbf{x}'_l$ and the offset b, obtained by solving Equation 3.26, leads to

$$\min_{\boldsymbol{\theta}} \lambda \|\boldsymbol{\theta}\|_1 + \frac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{i=1}^{N} \max(0, 1 - y_i f_{\boldsymbol{\theta}}(\mathbf{x}''_i)). \tag{3.27}$$

which is an instance of the regularized risk minimization problem $\min_{\boldsymbol{\theta}} \lambda\Omega(\boldsymbol{\theta}) + R(\boldsymbol{\theta})$, where $\Omega(\boldsymbol{\theta})$ is a regularization term and $R(\boldsymbol{\theta})$ is an upper bound on the empirical risk. To solve such non-differentiable risk minimization problems bundle methods have recently gained increasing interest in the machine learning community. For the case that the risk function R is non-negative and convex it is always lower bounded by its cutting plane at a certain point $\boldsymbol{\theta}^i$:

$$R(\boldsymbol{\theta}) \geq < \mathbf{a}^i, \boldsymbol{\theta} > + b^i \text{ for all i} \tag{3.28}$$

where $\mathbf{a}^i := \partial_{\boldsymbol{\theta}} R(\boldsymbol{\theta}^i)$ and $b^i := R(\boldsymbol{\theta}^i) - < \mathbf{a}^i, \boldsymbol{\theta}^i >$. Bundle methods build an iteratively increasing piecewise lower bound of the objective function by utilizing its cutting planes. Starting with an initial solution it solves the problem where R is approximated by one initial cutting plane using standard solver. A second cutting plane is build at the solution of the approximated problem. The new approximated lower bound of R is now the maximum

over all cutting planes. The more cutting planes are added the more accurate gets the lower bound of the risk function.

For the general case of non-linear kernel functions the problem in Equation 3.27 is non-convex and therefore especially hard to optimize. In the special case of a linear kernel, the problem is convex and the applied bundle method converges towards the global optimum. Some efforts have been made to adjust bundle methods to handle non-convex problems [93, 41]. We adapted the method of [41] to apply L1 regularization instead of L2 regularization and employ it to solve the optimization problem in Equation 3.27. Although the convergence rate of $O(1/e)$ to a solution of accuracy e [41] does no longer apply for our L1 regularized version, we observed that the algorithm converges withing the order of 10 iterations which is in the same range as for the algorithm in [41]. An overview of the suggested iterative dimensionality reduction algorithm is given in Algorithm 3.1.

3.3 Representing Curvature Self-Similarity

Although several methods have been suggested for the robust estimation of curvature, it has been mainly represented indirectly in a contour based manner [10, 184] and to locate interest points at boundary points with high curvature value. To design a more exact object representation that represents object curvedness in a natural way we revisit the idea of [122] and design a novel curvature self-similarity descriptor based on curvature. We make use of the advantages of global self-similarity and compute all pairwise curvature similarities across the whole image. This results in a very high dimensional object representation. As mentioned before such high dimensional representations have a natural need for dimensionality reduction which we fulfill by applying our embedded feature selection algorithm outlined in Section 3.2.2. In this section we will first review curvature for object representation and the concept of self-similarity and then provide details on the computation of our novel curvature self-similarity descriptor.

3.3.1 Review: Curvature for Object Representation

Figure 3.2 shows that a straight line approximation such as histograms of oriented gradients is not detailed enough and that histograms of curvature are able to provide a more detailed description of objects. Monroy et al. [122] extend the widely used object representation based on gradient orientation histograms by incorporating a robust description of curvature. It was shown that histograms of curvature are able to capture the shape information of complex objects and yields orthogonal information to the state-of-the-art theme of histograms of oriented gradients for visual search tasks.

To estimate the curvature for planar boundaries Monroy et al. [122] used the chord-to-point distance accumulation of Han et al. [78] due to its efficiency and stability. Let B be a set of N consecutive boundary points, $B := \{p_0, p_1, p_2, ..., p_{N-1}\}$ representing one line segment. A fixed integer value l defines a line L_i between pairs of points p_i to p_{i+l}, where $i + l$ is taken modulo N. The perpendicular distance D_{ik} is computed from L_i

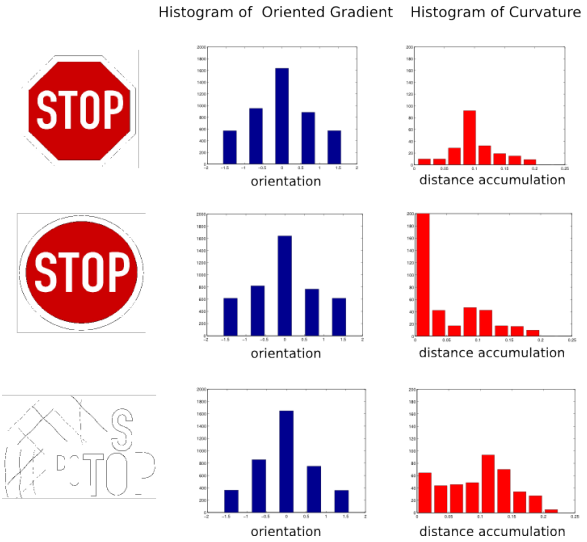


Figure 3.2: The columns from left to right show the original image, histograms of oriented gradients and histograms of curvature. The examples show, that a smooth curve cannot be distinguished from one with corners or from a set of differently oriented lines in arbitrary configuration using solely histograms of oriented gradients [122].

to the point p_k, using the euclidean distance. The distance accumulation for point p_k and a chord length l is the sum

$$h_l(k) = \sum_{i=k-l}^{k} D_{ik}. \tag{3.29}$$

In order to compute histograms of curvature the absolute value of the distance accumulation is computed for the edges provided by probabilistic boundary detector [119]. To build the final curvature descriptor the image is divided into a grid of multiple resolutions. The number of grid cells depends on the current resolution level. The first level contains only one cell and the number of cells is increasing with the level. In particular, the image has 2^s grid cells along each dimension for level s, where $s = 0$ is the coarsest level. Monroy et al. [122] use 4 levels for both HOG and curvature histograms. For each grid cell a histogram with 10 bins is computed over the absolute distance accumulation values contained in the cell. The histograms from each level are weighted according to 2^s and are concatenated to form the final feature vector that encodes local and global curvature statistics of the image.

3.3.2 Review: Self-Similarity

The idea of self-similarity was first suggested by Shechtman et al. [151] who proposed a descriptor based on local self-similarity (LSS). Instead of measuring image features directly it measures the correlation of an image patch with a larger surrounding image region. The general idea of self-similarity was used in several methods and applications [40, 90, 173, 183]. In [90] self-similarity is used to improve the Local Binary

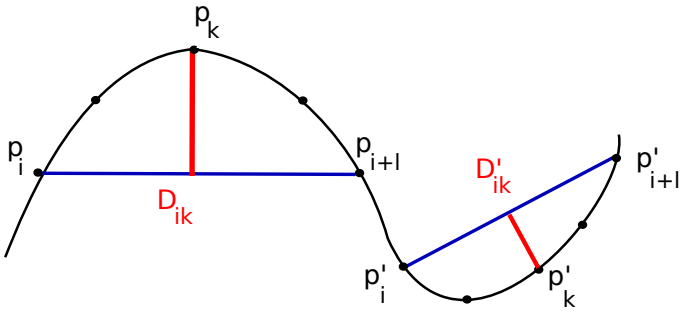


Figure 3.3: Visualization of curvature computation. D_{ik} is on the left-hand side of the vector $(p_{i+l} - p_i)$ and therefore has a positive sign, while D'_{ik} is on the right-hand side of the vector $(p'_{i+l} - p'_i)$ and therefore gets a negative sign.

Pattern (LBP) descriptor for face identification. Deselaers et al. [40] explored global self-similarity (GSS) and showed its advantages over local self-similarity (LSS) for object detection. Furthermore, Walk et al. [173] showed that using color histograms directly is decreasing performance, while using color self-similarity (CSS) as a feature is more appropriate. Besides object classification and detection, self-similarity was also used for action recognition [90] and turned out to be very robust to viewpoint variations.

3.3.3 Curvature Self-Similarity Descriptor

To describe complex objects, it is not sufficient to build a self-similarity descriptor solely based on curvature information, since self-similarity of curvature leaves open many ambiguities. To resolve these ambiguities we add 360 degree orientation information to get a more accurate descriptor. We are using 360 degree orientation, since curved lines cannot be fully described by their 180 degree orientation. This is different to straight lines, where 180 degree orientation gives us the full information about the line. Consider a half circle, with an arbitrary tangent line on it. The tangent line has an orientation between 0 and 180 degrees. However, it does not provide information on which side of the tangent the half circle is actually located, in contrast to a 360 degree orientation. Therefore, using a 180 degree orientation yields to high similarities between a left curved line segment and a right curved line segment.

As a first step we extract the curvature information and the corresponding 360 degree orientation of all edge pixels in the image. To estimate the curvature we follow the approach presented in [122] and use the distance accumulation method of Han et al. [78], which accurately approximates the curvedness along given 2D line segments (see Section 3.3.1). While Monroy et al. [122] only used the absolute curvature value we are also using the sign of the curvature to compute the 360 degree orientation. The distance D_{ik} is positive if p_k is on the left-hand side of the vector $(p_{i+l} - p_i)$, and negative otherwise (see Figure 3.3 and Figure 3.6). To get the 360 degree orientation information we compute the gradient of the probabilistic boundary edge image [119] and extend the resulting 180 degree gradient orientation to a 360 degree orientation using the sign of the curvature.

Contrary to the original curvature feature proposed in [122], where histograms of curvature are computed using differently sized image regions, we build our basic curvature

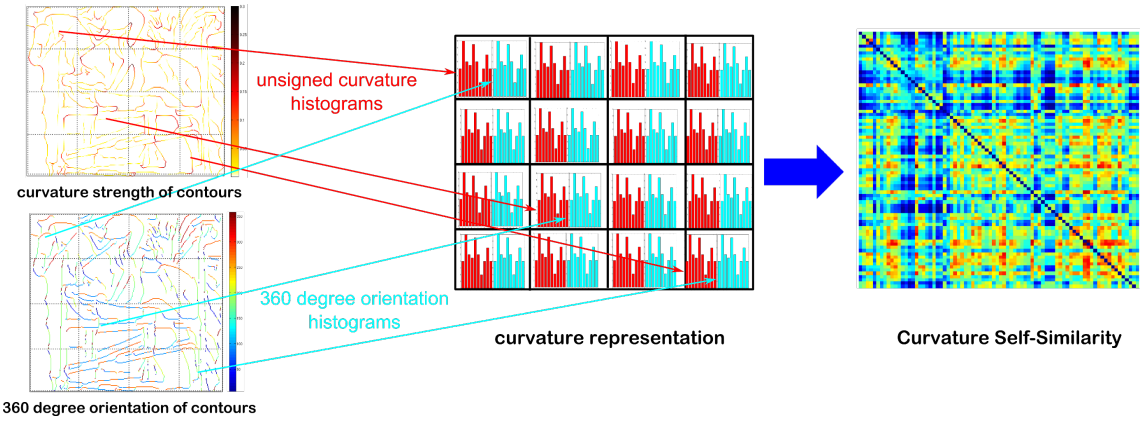


Figure 3.4: In the left column of the figure on the top the absolute curvature value of an example image is shown. On the bottom the corresponding 360 degree orientation of all edge pixels is visualized. After dividing these two input signals in grids of 8×8 pixels, histograms are build for each cell. The curvature and orientation histograms are concatenated and similarities are computed, using histogram intersection, to build the final self similarity representation.

feature using equally sized cells to make it more suitable for computing self-similarities. We divide the image into non-overlapping 8×8 pixel cells and build histograms over the curvature values in each cell. Next, we do the same for the 360 degree orientation and concatenate the two histograms. This results in histograms of 28 bins, 10 bins representing the curvature and 18 bins representing the 360 degree orientation. There are many ways to define similarities between histograms. We follow the scheme that was applied to compute self similarities between color histograms [173] and use histogram intersection as a comparison measure to compute the similarities between different curvature histograms in the same bounding box. Histogram intersection is given by

$$d_{\text{hist}}(\mathbf{u}, \mathbf{v}) = \sum_{j=1}^{n} \min(u_j, v_j) \tag{3.30}$$

where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{28}$. Furthermore, we apply an L2-normalization to the final self-similarity vector. Figure 3.4 gives an overview of the feature construction process. The computation of self-similarities between all curvature-orientation histograms results in an extremely high-dimensional representation. Let D be the number of cells in an image, then computing all pairwise similarities results in a D^2 large curvature self-similarity matrix. Some examples are shown in Figure 3.5. Since, the similarity matrix is symmetric we use only the upper triangle which results in a $(D \cdot (D-1)/2)$-dimensional vector. This representation gives a very detailed description of the object. The higher dimensional a descriptor gets, the more likely it contains noisy and correlated dimensions. Furthermore, it is also intuitive that not all similarities extracted from a bounding box are helpful to describe the object. To discard such superfluous dimensions we apply our embedded feature selection method to the proposed curvature self-similarity representation.

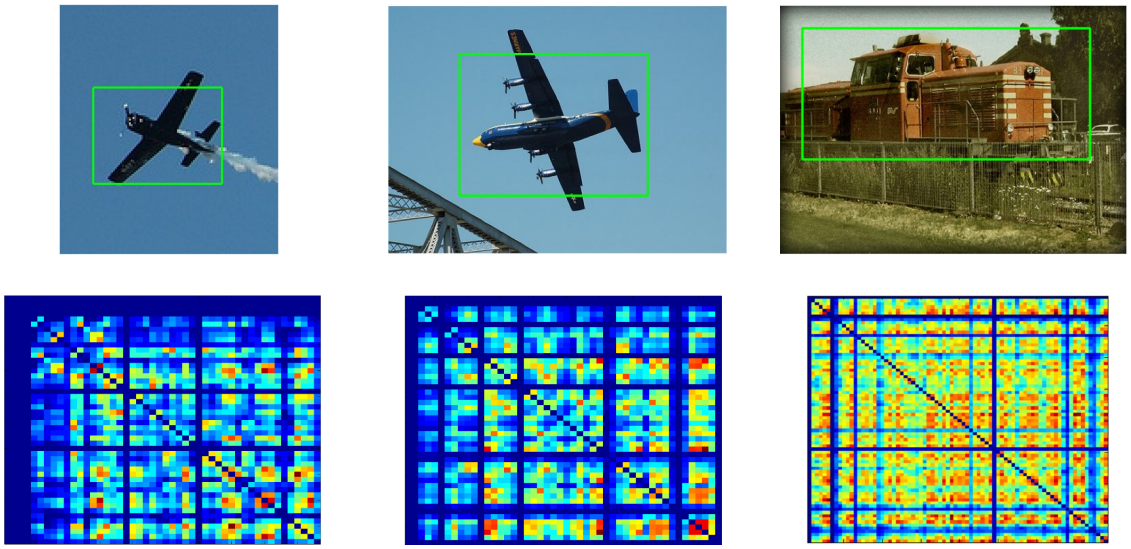


Figure 3.5: Our visualization shows the original images along with their curvature self-similarity matrices displaying the similarity between all pairs of curvature histogram cells. While curvature self-similarity descriptor is similar for the same object category it looks quite different to other object categories

3.4 Experiments

We evaluate our curvature self-similarity descriptor in combination with the suggested embedded dimensionality reduction algorithm for the object detection task on the PASCAL dataset [48]. Our experiments show, that curvature self-similarity is providing complementary information to straight lines, while our feature selection algorithm is further improving performance by fulfilling its natural need for dimensionality reduction.

The common basic concept shared by many current detection systems are high-dimensional, holistic representations learned with a discriminative classifier, mostly an SVM [167]. In particular the combination of HOG [37] and SVM constitutes the basis of many powerful recognition systems and it has laid the foundation for numerous extensions like, part based models [56, 123, 147, 188], variations of the SVM classifier [56, 161] and approaches utilizing context information [82, 155]. These systems rely on high-dimensional holistic image statistics primarily utilizing straight line approximations. In this paper we explore a orthogonal direction to these extensions and focus on how one can improve on the basic system by extending the straight line representation of HOG to a more discriminative description using curvature self-similarity. At the same time our aim is to reduce the dimensionality of such high-dimensional representations to decrease the complexity of the learning procedure and to improve generalization performance.

The experimental section is organized as follows: First an overview is given over the PASCAL VOC dataset (Section 3.4.1). The next part of the experiments (Section 3.4.2) adjust the selection parameter λ of the iterative dimensionality reduction technique via cross-validation. Furthermore, performance of the feature selection algorithm is compared to L2 regularized SVM [28, 115] and DrSVM [175]. In the final part (Section 3.4.3) we evaluate the suggested curvature self-similarity feature after applying our novel

feature selection method to it.

3.4.1 PASCAL Visual Object Classes

The PASCAL Visual Object Classes (VOC) [49] is a challenging benchmark for object category classification and detection. The PASCAL challenge has been organized from 2005 until 2012. Each year a new dataset was released together with annotations and standard evaluation procedure. Until 2007 the annotation of the testing data has been released after the challenge. This is due to the fact that since 2009 the dataset was only augmented with new images instead of providing a whole set of new images each year. Instead of releasing the testset annotation an evaluation server was set up for measuring performance on the testset. This also makes sure that parameters of approaches published after the challenge can not be adjusted to the testset. The downside is that only a small number of evaluations are allowed for each suggested detection system. However, to show the individual strengths of the two contributions suggested in this chapter we need to perform a number of evaluations. Since this is not supported by the PASCAL VOC evaluation server we follow the best practice guidelines and use the VOC 2007 dataset.

The PASCAL VOC 2007 dataset contains images collected from flickr. In total 9963 images containing 24640 annotated objects from twenty classes have been collected. Furthermore, the dataset provides a fixed split into 50% training/validation and 50% testing data. The twenty classes can be divided into four main topics: vehicles, animals, household objects and people. The dataset is particularly challenging due to a wide variation of viewpoints and lighting conditions, high intra-class variability, low inter-class variability, and a high amount of occlusions and truncations.

For evaluation a ranked list of object bounding boxes together with an associated confidence score is submitted for each class. For all detected bounding boxes B_d in an image the overlap a_0 with the annotated groundtruth bounding boxes B_{gt} is computed :

$$a_0 = \frac{area(B_d \cap B_{gt})}{area(B_d \cup B_{gt})}. \tag{3.31}$$

$B_d \cap B_{gt}$ is the intersection between the ground truth and the detected bounding box and $B_d \cup B_{gt}$ is their union. A detected bounding box is considered as correct detection if the overlap a_0 is larger than 50%. Furthermore, if multiple detections are dedicated to a ground truth box only the highest ranking box is considered as positive, while the remaining detections are double detections and are counted as false positives. To avoid such double detections a non-maximum suppression is typically performed to remove overlapping boxes.

From the number of true positives (TP), i.e. correctly detected objects, and false positives (FP), i.e. a non existing object was detected, one can compute precision and recall. Precision is the amount of samples that have been detected correctly among all detected objects

$$\text{precision} = \frac{\#TP}{\#TP + \#FP}, \tag{3.32}$$

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow
linSVM	66.1	80.0	53.0	53.1	70.7	73.8	75.3	61.2	63.8	70.7
DrSVM	59.1	77.6	53.5	49.9	64.4	71.6	75.8	50.8	56.1	64.5
linSVM + FS	69.7	80.3	55.5	56.2	71.8	74.0	75.9	63.2	64.8	71.0
FIKSVM	80.1	74.8	57.1	59.3	63.3	73.9	77.3	77.3	69.1	66.4
FIKSVM + FS	80.4	74.9	57.5	62.1	66.7	73.9	78.0	80.1	70.6	69.9

	table	dog	horse	mbike	pers	plant	sheep	sofa	train	tv	**mean**
linSVM	71.4	57.2	76.5	83.0	72.9	47.7	55.1	61.1	70.4	73.1	66.8
DrSVM	59.9	53.9	70.9	76.5	72.3	47.7	66.3	69.0	67.7	79.7	64.3
linSVM + FS	72.0	57.8	77.2	83.3	73.0	49.7	56.7	62.4	70.7	73.8	68.0
FIKSVM	64.1	61.7	74.6	70.9	79.4	47.5	62.0	59.8	76.9	69.3	68.1
FIKSVM + FS	67.6	64.6	79.7	74.2	79.6	53.0	64.2	64.6	77.1	69.8	70.4

Table 3.1: Average precision of our iterative feature reduction algorithm for linear and non-linear kernel function using our final feature vector consisting of HOG+Curv+CurvSS. For linear kernel function we compare our feature selection (linSVM+FS) to L2 normalized linear SVM (linSVM) and to the doubly regularized SVM (DrSVM) [175]. For non-linear kernel function we compare the fast intersection kernel SVM (FIKSVM) [115] with our feature selection (FIKSVM+FS).

while recall measures the amount of recognized positives among all positive samples

$$\text{recall} = \frac{\#TP}{\#P}. \tag{3.33}$$

By varying the threshold on the confidence score a precision-recall curve can be computed. Performance is then summarized in a single number called average precision. The average precision is the area under the precision-recall curve.

3.4.2 Evaluation of Feature Selection

All experiments in this section are performed using our final feature vector consisting of HOG, curvature (Curv) and curvature self-similarity (CurvSS). We apply our iterative dimensionality reduction algorithm in combination with linear L2 regularized SVM classifier (linSVM) [28] and non-linear fast intersection kernel SVM (FIKSVM) by Maji et al. [115].

Histogram intersection (cf. Equation 3.30) is often used as a comparison measurment between histograms. However, due its positive definiteness [126] it is also a suitable kernel for SVMs. In [115] it is shown how the runtime complexity of the intersection kernel SVM can be reduced from linear to logarithmic in the number of support vectors. Therefore, the so called FIKSVM is widely used and evaluation is relatively fast compared to other non-linear kernels. Nevertheless, computational complexity is still an issue on

the PASCAL dataset as the number of support vectors grows linearly with the amount of training data [156]. This is why on this database linear kernels are typically used [56, 155].

Because of the high computational complexity of DrSVM and FIKSVM, we compare to these methods on a smaller train and test subset obtained from the PASCAL training and validation data in the following way. All training and validation data from the PASCAL VOC 2007 dataset are used to train an SVM using our final object representation on all positive samples and randomly chosen negative samples. The resulting model is used to collect hard negative samples. The set of collected samples is split up into three sets: training, validation and test. Out of the collected set of samples every tenth sample is assigned to the hold out test set which is used to compare the performance of our feature selection method. The remaining samples are randomly split into training and validation set of equal size which are used to perform the feature selection. The reduction algorithm is applied on 5 different training/validation splits which results in five different sets of selected features. For each set we train an L2 norm SVM on all samples from the training and validation set using only the remaining dimensions of the feature vector. Then we choose the feature set with the best performance on the hold out test set. To find the best performing selection parameter λ, we repeat this procedure for different values of λ.

The performance of our dimensionality reduction algorithm is compared to the performance of linSVM and DrSVM [175] for the case of a linear kernel. Since DrSVM is solving a similar optimization problem as our suggested feature selection algorithm for a linear kernel this comparison is of particular interest. We are not comparing performance to DrSVM in the non-linear case since it is performing feature selection in the higher dimensional kernel space rather than in the original feature space. Instead we compare our feature selection method to that of FIKSVM for the non-linear case. Our feature selection method reduces the dimensionality of the feature by up to 55% for the linear case and by up to 40% in the non-linear case, while the performance in average precision is constant or increases beyond the performance of linSVM and FIKSVM. On average our feature selection increases performance about 1.2% for linSVM and 2.3% for FIKSVM on the hold-out testset. The DrSVM is actually decreasing the performance of linSVM by 2.5% while discarding a similar amount of features. All in all our approach improves the DrSVM by 3.7% (see Table 3.1).

Our results confirm that our feature selection method reduces the amount of noisy dimensions of high-dimensional representations and therefore increases the average precision compared to a linear and non-linear SVM classifier without applying any feature selection. For the linear kernel we showed furthermore that the proposed feature selection algorithm achieves gain over the DrSVM.

3.4.3 Object Detection using Curvature Self-Similarity

In this section we provide a structured evaluation of the parts of our final object detection system. We use the HOG of Felzenszwalb et al. [56, 57] as baseline system, since it is the basis for many powerful object detection systems. All detection results are measured in terms of average precision performing object detection on the PASCAL VOC 2007 dataset. To the best of our knowledge neither curvature nor self-similarity was used

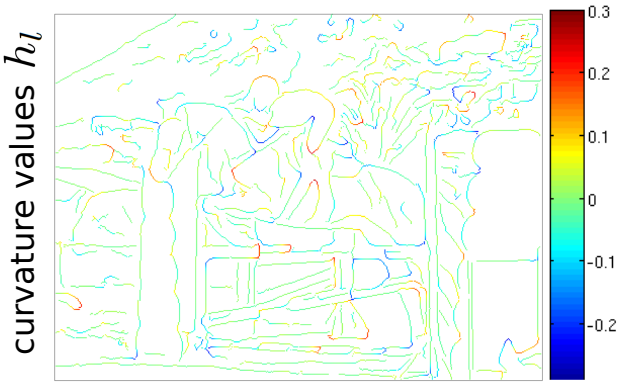


Figure 3.6: Based on meaningful edge images one can extract accurate curvature information which is used to build our curvature self-similarity object representation.

Figure 3.7: A significant number of images from PASCAL VOC feature contour artifacts i.e, due to their size, low resolution, or compression artifacts. The edge maps are obtained from the state-of-the-art probabilistic boundary detector [119]. It is evident that objects like the sheep are not defined by their boundary shape and are thus beyond the scope of approaches based on contour shape.

to perform object detection on a dataset of similar complexity as the PASCAL dataset so far. Deselaers et al. [40] evaluated their global self-similarity descriptor (GSS) on the simpler classification challenge on the PASCAL VOC 2007 dataset, while the object detection evaluation was performed on the ETHZ shape dataset. However, it was shown in [122] that including curvature already solves the detection task almost perfectly on the ETHZ dataset. Furthermore, [122] outperforms the GSS descriptor on three categories and reached comparable performance on the other two. Thus we evaluate on the more challenging PASCAL dataset.

Since the proposed approach models the shape of curved object contours and reduces the dimensionality of the representation, we expect it to be of particular value for objects that are characterized by their shape and where their contours can be extracted using state-of-the-art methods (see Figure 3.6). However, a significant number of images form PASCAL VOC are corrupted due to noise or compression artifacts (see Figure 3.7). Therefore state-of-the-art edge extraction fails to provide any basis for contour based approaches on these images and one can therefore only expect a significant gain on categories where proper edge information can be computed for a majority of the images.

Our training procedure makes use of all objects that are not marked as difficult from

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow
HOG of [57]	19.0	*44.5*	2.9	4.2	13.5	37.7	39.0	8.3	11.4	15.8
HOG	20.8	43.0	2.1	5.0	*13.7*	37.8	38.7	6.7	12.1	16.3
HOG+Curv	23.0	42.6	*3.7*	6.7	12.4	38.6	39.9	7.5	10.0	16.9
HOG+Curv+FS	25.4	42.9	*3.7*	6.8	13.5	38.8	40.0	8.1	12.0	17.1
HOG+Curv+ CurvSS	28.6	39.1	2.3	6.8	12.9	40.3	38.8	9.3	11.1	13.9
HOG+Curv+ CurvSS+FS	*28.9*	43.1	3.5	*7.0*	13.6	*40.6*	*40.4*	*9.6*	*12.5*	*17.3*

	table	dog	horse	mbike	pers	plant	sheep	sofa	train	tv	**mean**
HOG of [57]	10.5	2.0	43.5	29.7	24.0	3.0	*11.6*	17.7	28.3	32.4	20.0
HOG	9.8	2.2	42.4	29.5	24.3	3.8	11.5	17.6	29.0	33.4	20.0
HOG+Curv	13.0	3.7	46.0	30.5	25.5	4.0	8.7	18.7	32.3	33.6	20.9
HOG+Curv+FS	15.6	3.7	46.4	*30.8*	25.7	4.0	11.3	19.1	32.3	33.6	21.5
HOG+Curv+ CurvSS	16.3	6.2	48.0	27.5	27.2	4.2	9.3	20.5	35.9	*34.8*	21.7
HOG+Curv+ CurvSS+FS	*16.7*	*6.4*	*48.5*	30.6	*27.3*	*4.8*	*11.6*	*20.7*	*36.0*	*34.8*	*22.7*

Table 3.2: Detection performance in terms of average precision of the HOG baseline system, HOG and curvature (Curv) before and after discarding noisy dimensions using our feature selection method (FS) and our final detection system consisting of HOG, curvature (Curv), the suggested curvature self-similarity (CurvSS) with and without feature selection (FS) on the PASCAL VOC 2007 dataset. Note, that we use all data points to compute the average precision as it is specified by the default experimental protocol since VOC 2010 development kit. This yields lower but more accurate average precision measurements.

the training and validation set. We evaluate the performance of our system on the full testset consisting of 4952 images containing objects from 20 categories using a linear SVM classifier [28]. Due to the large amount of data in the PASCAL database the usage of intersection kernel for object detection becomes comparable intractable. Results of our final system consisting of HOG, curvature (Curv), curvature self-similarity (CurvSS) and our embedded feature selection method (FS) are reported in terms of average precision in Table 3.2. We compare our results to that of HOG [57] without applying the part based model. Additionally we show results of our own HOG baseline system which is using standard linear SVM [28] instead of the latent SVM used in [57]. Furthermore, we show results with and without feature selection to show the individual gain of the curvature self-similarity descriptor and our embedded feature selection algorithm.

The results show that the suggested self-similarity representation in combination with feature selection improves performance on most of the categories. All in all this results in an increase of 2.7% in average precision compared to the HOG descriptor. One can observe that curvature information in combination with our feature selection algorithm is already improving performance over the HOG baseline and that adding curvature self-similarity additionally increases performance by 1.2%. The gain obtained by applying our feature selection (FS) depends obviously on the dimensionality of the feature vector; the higher the dimensionality the more can be gained by removing noisy dimensions. For HOG+Curv applying our feature selection is improving performance by 0.6% while the gain for the higher dimensional HOG+Curv+CurvSS is 1%. The results underline that curvature information provides complementary information to straight lines and that feature selection is needed when dealing with high dimensional features like self-similarity.

3.5 Discussion

We have observed that high-dimensional representations cannot be sufficiently handled by linear and non-linear SVM classifiers. An embedded feature selection method for SVMs has therefore been proposed, which has been demonstrated to successfully deal with high-dimensional descriptions and therefore increases the performance of linear and intersection kernel SVM. Moreover, the proposed curvature self-similarity representation has been shown to add complementary information to widely used orientation histograms.histograms.

CHAPTER 4

MAX-MARGIN REGULARIZATION FOR REDUCING ACCIDENTALNESS IN CHAMFER MATCHING

In the previous chapter the benefit of mid-level representations has been shown for holistic object models, by modeling self-similarities between image regions of an object hypothesis. As the hypothesis are divided into cells this can be reinterpreted as a rigid part-based model, where similarities have been computed between the individual parts. This chapter discusses, how mid-level representations are also of great use for more flexible part-based object detection approaches.

As discussed in Section 2.2.1 chamfer matching is a widely used technique for object detection. Due to its simplicity and efficiency it has been employed in a variety of applications to match whole object boundaries, as well as partial object contours. Despite these advantages chamfer matching has a serious drawback when contours are matched in cluttered image regions. Contour matches in cluttered regions have a high accidentalness and can not be distinguished from matches on the actual object. Recent research made some attempts to improve specificity by including orientation information [69, 38, 152, 108] in the distance function. Furthermore, this issue was addressed by learning the relevance of model points and gives higher weight to more important model points (see Section 4.2.1). While this extensions help to fit the template more accurately to the object contours, dense clutter as shown in Figure 4.5 is still a serious problem of this approach.

This work is addressing the problem of dense clutter by developing a novel mid-level representation. Mid-level representations are providing a more detailed descripton of the object and are therefore bridging the gap between the low-level feature, in this case the foreground contour of an object part, and the high-level concept of an object. Co-occurrences between different low-level features have been shown to lead to very detailed and robust image descriptors for object detection. Following this idea of

co-occurrences, the flexible co-placement of generic background contours are learned to reduce the accidentalness of matches in dense background clutter. The co-occurrence of generic background contours is integrated together with current improvements on matching the foreground model, i.e. orientation information [108] and weighted model points (see Section 4.2.1) into a single max-margin framework.

In the remaining of this chapter, first state-of-the-art chamfer matching approaches are reviewed (Section 4.1), next the concept of non-accidentalness will be discussed (Section 4.2), then the learning procedure for the suggested chamfer regularization will be explained (Section 4.3), and finally an extensive evaluation of the suggested framework is performed (Section 4.5).

4.1 State-of-the-art Chamfer Matching

Chamfer matching (CM) is a popular shape matching algorithm due to its speed and robustness [158]. Therefore, it has been used in a large number of applications in computer vision. It was first introduced by Barrow et al. [7] to match two sets of contour fragments. Since then chamfer matching has been widely applied and has been a successful technique for detecting complete objects or their parts. In [19] hierarchical chamfer matching was suggested where edge points are matched in a coarse-to-fine-manner. Later, chamfer matching was used to build powerful detectors as proposed in [70, 104, 107]. Leibe et al. [104] combine local features with global shape cues obtained from chamfer matching to verify and refine hypotheses. In [70], Gavrila and Munder have applied chamfer matching for real-time pedestrian detection and tracking. Lin et al. [107] have proposed a hierarchical part-template matching approach, for detection and segmentation, which measures shape information in terms of chamfer matching scores.

Besides the usage of standard chamfer matching, several works engage the question, how chamfer matching can be enhanced to make it less sensitive to clutter. In the following we will first review standard chamfer matching, oriented chamfer matching (OCM) [152] and directional chamfer matching (DCM) [108], two approaches that investigated in including orientation information, and normalized oriented chamfer matching (NOCM) [113], an improvement of oriented chamfer matching for dense clutter.

Chamfer Matching Lets assume that each object is represented by a collection of contours of its different parts. Let $P = \{\mathbf{p}_i\}$ and $Q = \{\mathbf{q}_j\}$ be the pixels of an object part and query edge maps respectively. For a given location $\mathbf{x}$ of the object part in the query image, chamfer matching aims to find the best $\mathbf{q}_j \in Q$ for each $\mathbf{p}_i \in P$. The chamfer distance is defined as

$$d_{CM}^{(P,Q)}(\mathbf{x}) = \frac{1}{|P|} \sum_{\mathbf{p}_i \in P} \min_{\mathbf{q}_j \in Q} |(\mathbf{p}_i + \mathbf{x}) - \mathbf{q}_j| \tag{4.1}$$

Chamfer matching is robust against small rotations, misalignments, occlusions, and deformations. The matching cost can be efficiently computed in linear time using distance

transformation given in Equation 2.8

$$d_{CM}^{(P,Q)}(\mathbf{x}) = \frac{1}{|P|} \sum_{\mathbf{p}_i \in P} DT_Q(\mathbf{p}_i + \mathbf{x}) \tag{4.2}$$

In practice the distance is often thresholded to make it more robust against missing edges in the query image Q:

$$d_{CM,\tau}^{(P,Q)}(\mathbf{x}) = \frac{1}{|P|} \sum_{\mathbf{p}_i \in P} \min(DT_Q(\mathbf{p}_i + \mathbf{x}), \tau). \tag{4.3}$$

Furthermore, thresholding with the constant τ allows normalization to the range $[0, 1]$:

$$d_{CM,\tau}^{(P,Q)}(\mathbf{x}) = \frac{1}{\tau|P|} \sum_{\mathbf{p}_i \in P} \min(DT_Q(\mathbf{p}_i + \mathbf{x}), \tau). \tag{4.4}$$

Oriented Chamfer Matching Shotton et al. [152] suggested an improved matching scheme called oriented chamfer matching (OCM) that takes into account the orientation mismatch between pixels. Exploiting the edge orientation improves the robustness, since it is unlikely that clutter edges align in orientation and position. Therefore, the cost function is extended by an explicit cost for the orientation mismatch, given by the difference in orientation between edges in the template P and the edge map Q

$$d_{orient}^{(P,Q)}(\mathbf{x}) = \frac{2}{\pi|P|} \sum_{\mathbf{p}_i \in P} |\phi(\mathbf{p}_i) - \phi(ADT_Q(\mathbf{p}_i + \mathbf{x}))| \tag{4.5}$$

where $\phi(\mathbf{p}_i)$ be the edge orientation of the edge point $\mathbf{p}_i$ and ADT_Q be the argument distance transform. The argument distance transform gives the locations of the closest point in Q

$$ADT_Q(\mathbf{x}) = \arg\min_{\mathbf{q}_j \in Q} |\mathbf{x} - \mathbf{q}_j| \tag{4.6}$$

The suggested oriented chamfer matching uses a linear combination of the distance and the orientation term

$$d_{OCM}^{(P,Q)}(\mathbf{x}) = (1 - \lambda) \cdot d_{CM,\tau}^{(P,Q)}(\mathbf{x}) + \lambda \cdot d_{orient}^{(P,Q)}(\mathbf{x}). \tag{4.7}$$

The parameter λ is controlling the importance of the orientation term.

Directional Chamfer Matching In [108] an alternative approach for incorporating edge orientation has been proposed which solves the matching problem in an augmented space. Instead of modeling the orientation mismatch as a separate term as in OCM each edge pixel is augmented with a direction term. This method is called directional chamfer matching and its distance function is defined as

$$d_{DCM}^{(P,Q)}(\mathbf{x}) = \frac{1}{|P|} \sum_{\mathbf{p}_i \in P} \min_{\mathbf{q}_j \in Q} |(\mathbf{p}_i + \mathbf{x}) - \mathbf{q}_j| + \lambda |\phi(\mathbf{p}_i + \mathbf{x}) - \phi(\mathbf{q}_j)|. \tag{4.8}$$

Similar to the OCM formulation λ denotes the weighting factor between location and orientation terms. It was shown that the suggested directional chamfer matching (DCM) achieves a superior performance compared to oriented chamfer matching. Moreover, the approach is significantly reducing the matching time from linear to sublinear by using 3D distance transforms and integral images.

Normalized Oriented Chamfer Matching Another improvement of chamfer matching was suggested in [113]. While this work is build upon oriented chamfer matching, the focus of this work is not on improving the matching of the template, but on comparing the matching scores of the template to the matching scores of auxiliary contours, so called normalizers. The idea is that the normalizer contour matches as good as the foreground template in a cluttered image region. Therefore, a comparison between the oriented chamfer matching score of the template and a random normalizer is used to recognize if the template matches to clutter. For a target template P, a set of normalizers is created $N = \{\eta_k | k = 1, ...K\}$ and the ratio

$$R^{(P,Q,\eta_k)}(\mathbf{x}) = \frac{d^{(P,Q)}_{OCM}(\mathbf{x})}{d^{(\eta_k,Q)}_{OCM}(\mathbf{x})} \tag{4.9}$$

is computed. To get a suitable set of normalizers, a manually selected set of tuples of contour fragments is proposed and trained in a boosting framework where all the ratios are evaluated and used as weak learners

$$g^{(P,Q,\eta_k)}(\mathbf{x}) = \begin{cases} 1 & R^{(P,Q,\eta_k)}(\mathbf{x}) < t_k \\ 0 & \text{otherwise} \end{cases}. \tag{4.10}$$

The threshold t_k is chosen in the boosting framework to minimize the misclassification error and the final strong classifier is given as a weighted linear combination of the weak classifiers resulting in the normalized oriented chamfer matching score

$$G^{(P,Q)}(\mathbf{x}) = \sum_{k=1}^{K} \omega_k \cdot g^{(P,Q,\eta_k)}(\mathbf{x}) = d^{(P,Q)}_{NOCD}(\mathbf{x}). \tag{4.11}$$

Note, that while oriented chamfer matching is utilizing a distance, normalized oriented chamfer matching utilizes a similarity instead.

4.2 Modeling Accidentalness

The concept of non-accidentalness [109, 182] suggests that the significance of relations between parts or features is mainly dependent on the extend to which such a configuration could have appeared by accident. This concept was first suggested and exploited in the area of grouping and object recognition. In this work, accidental matches of the foreground template are detected by learning the interdependence (Section 4.2.1) of model points and by learning the co-occurrence of generic background contours (Section 4.2.2).

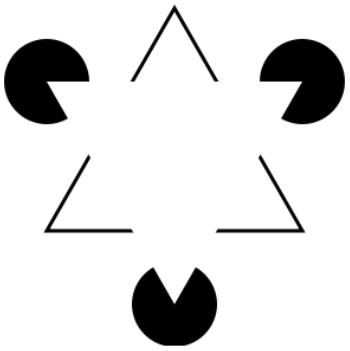


Figure 4.1: We are percepting this image as a white triangle partially occluding three black circles and another triangle with a black boundary. However, there is not a single edge in the image defining the white triangle. The contours are illusionary [92].

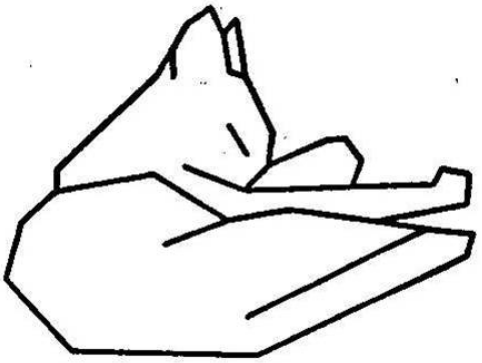


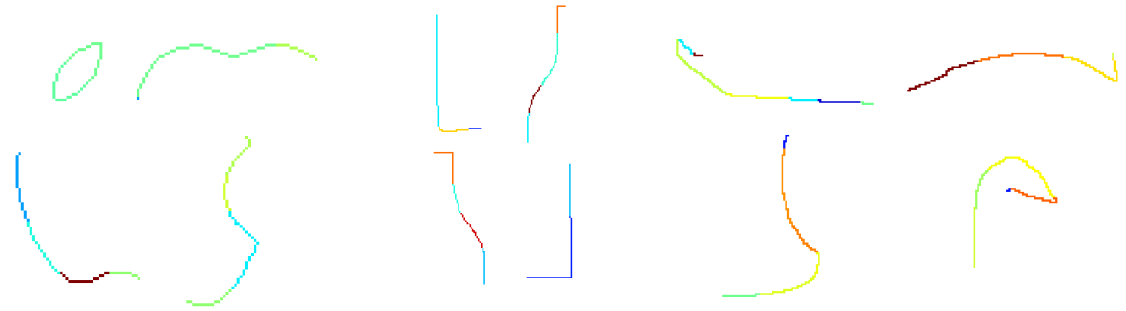
Figure 4.2: The shown cat is abstracted by selecting the 38 points of highest curvature and connecting them with straight edges [5].

4.2.1 Review: Interdependence of Model Points

Standard chamfer matching is utilizing object contours which only measure the mere sum of location differences of contour pixels. Extensions of chamfer matching such as OCM [152] and DCM [108] focus on adding orientation information to improve the matching quality of the foreground template. However, in both approaches, the score for an object hypothesis is obtained by summing over all the template pixels in the distance transform of the query image (see Equation 4.7 and 4.8). Therefore, these approaches measure the presence of individual model points in a query image independently. However, not all the pixels on an object part are equally important for detecting objects, as shown in the famous Kanizsa triangle shown in Figure 4.1. Provided only contour fragments around the corners, the whole triangle can be easily recognized. Similarly, Biederman [15] presents perceptual experiments with degraded contours that demonstrate the varying importance of different points on object contours. Another example is Attneave's cat [5] shown in Figure 4.2, where for instance, points of high curvature are proposed as the most useful features for recognition.

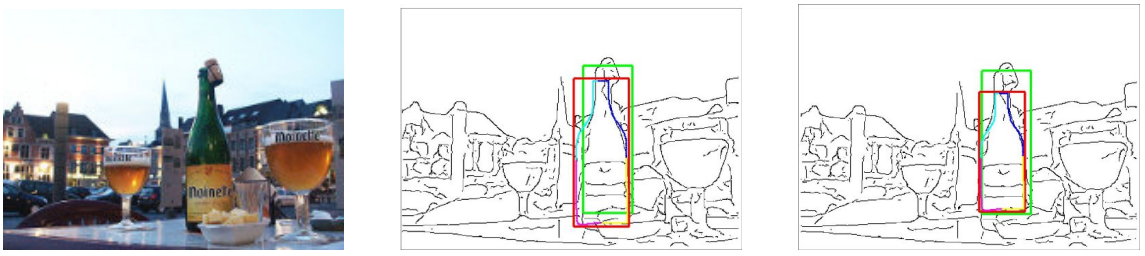
This issue is addressed by learning the relevance of model points which gives higher weight to more important model points. Such interdependence of model points is increasing the specificity of the model contour by learning the relative importance of all model points instead of treating them as independent. The weights modeling the interdependence between individual points of an object part are determined by learning discriminative weights of their co-occurrence, i.e. of their matching costs.

$$t_i^{(P,Q)}(\mathbf{x}) = \min_{\mathbf{q}_j \in Q} |(\mathbf{p}_i + \mathbf{x}) - \mathbf{q}_j| + \lambda |\phi(\mathbf{p}_i + \mathbf{x}) - \phi(\mathbf{q}_j)| \tag{4.12}$$



a) b) c)

Figure 4.3: Relative pixels weights for a) applelogos, b) bottles and c) swans learnt with a linear max-margin classifier. Red indicates high weight and blue low weight.



a) b) c)

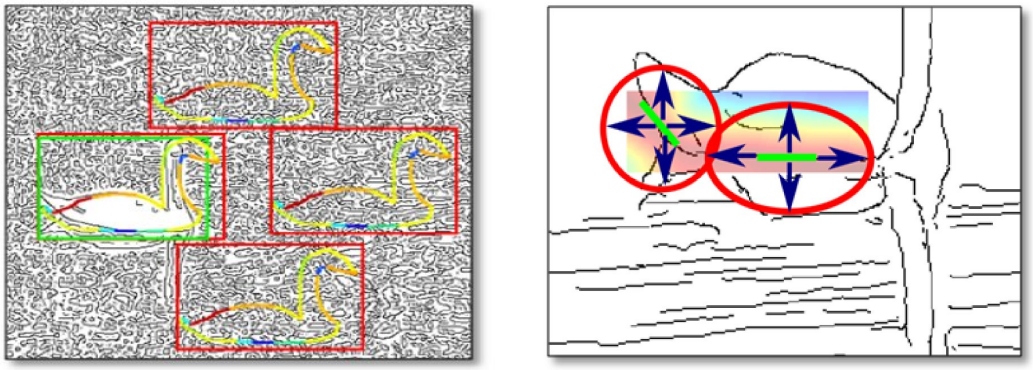
Figure 4.4: Learning discriminative weights for the co-occurrences of $t_i^{(P,Q)}(\mathbf{x})$ improves the matching score of shape templates as shown in the example here. The original image, the result obtained from directional chamfer matching, and the result obtained from foreground reweighting are shown in panels a,b and c respectively. The groundtruth bounding box is shown in green and the top scoring object hypotheses are shown in red.

Since adjacent pixels of an object part are statistically dependent, the line representation is utilized for the templates from [108] and discriminative weights are learned for each line of the object part. Thus, all the pixels which lie on the same line are assigned the same weight. Let $\bar{t}_l$ denote the matching cost of line l fitted to the object part.

$$\bar{t}_l = \sum_{i \in l} t_i^{(P,Q)}(\mathbf{x}) \tag{4.13}$$

The discriminative learning algorithm that discovers the weights for the co-occurrences of lines is described in Section 4.3. Figure 4.3 shows the relative importance of various pixels of the foreground template learnt using a linear SVM.

Other related work, for instance on saliency [91] and interest point detection [14], is not suitable for integration into chamfer matching. Moreover, interest points are detected based on each training image separately, whereas for the integration in a max-margin framework the importance of points of an object part needs to be based on joint consideration of all the training images.



(a) (b)

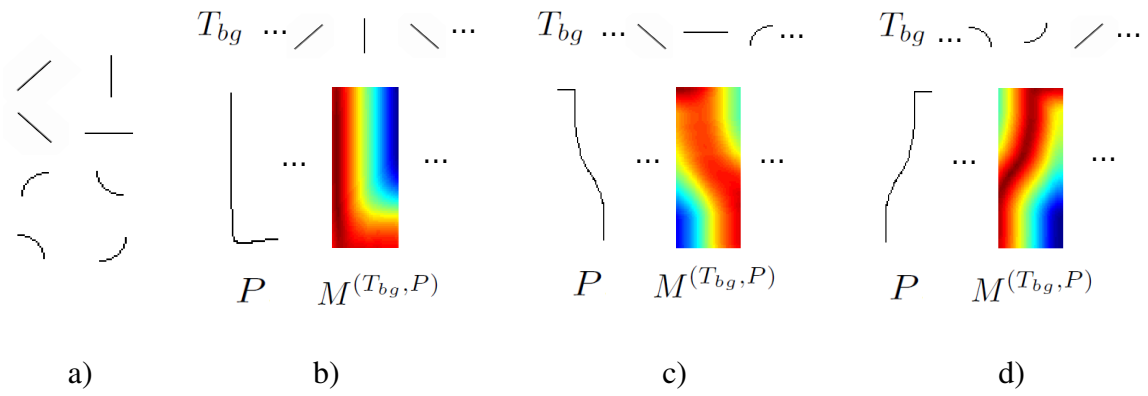
Figure 4.5: Dense clutter is one of the main drawbacks of chamfer matching since it can cause numerous spurious matches of the model contour as shown in a). b) shows how background contours are placed according to the foreground contour mask to measure accidentalness and reduce such spurious matches.

4.2.2 Background Contours for Modeling Accidentalness

In [158] Thayananthan et al. have compared shape context [11] and chamfer matching of templates for object detection in cluttered images. They reported that chamfer matching is more robust to clutter than shape context. Nevertheless, false positives in cluttered background were still found to be the major downside of chamfer matching (see Figure 4.5 a)).

Increasing the specificity of the model contour matches, by adding orientation information [108, 152] and learning the importance of foreground contour pixels (Section 4.2.1) can only partially solve the problems arising from background clutter as shown in Figure 4.8. Consequently, we need to measure the accidentalness of an object part matching in the background clutter. However, most previous work focuses on improving the matching of the foreground model contour such as OCM, DCM and the interdependence of model points. An exception is the recently suggested NOCM [113] approach which is focusing on this issue and aims to reduce chamfer matches in clutter. NOCM is normalizing template matches with manually combined normalizer contours to alleviate the impact of dense clutter on the matching result. The normalizers are placed at the center of the template matches. However, to sufficiently model complex background, it is important to combine simple contours via flexible placement going beyond the manual combinations of normalizers. We measure the accidentalness of a match to clutter by learning the co-placement of background contours dependent on the foreground.

The introduced background contours are a set of simple, generic contour segments (see Figure 4.6 a) that typically match equally well to background clutter and the correct part contour. Since each single background contour segment has a very low specificity we learn discriminative co-occurrence patterns which have very low accidentalness. By going for flexible spatial arrangements of background contours, we avoid manually combining tuples of normalizers consisting of one or two contours to form hand designed complex background templates as in [113]. Furthermore, we measure the amount of clutter only in the neighborhood of model contours, where clutter actually interferes with the matching of



a) b) c) d)

Figure 4.6: a) shows a set of simple background contours T_{bg}. These background contours are used to regularize the chamfer response of a part P. b)-d) show the masks $M^{(T_{bg},P)}$, described in Equation 4.14, obtained from placing the background contours at the top relative to the object part contour on the left.

the model contour, while in [113] background contours are placed at a fixed single location (the center of the model contour). The importance of the second point is illustrated by the following example. Consider a U-shaped object part being matched to a query image. Clutter from the query image that is situated within the U does not interfere with the object part. Only clutter that is close to the contour of the U will have an impact. Thus, Latecki et al. [113] miss out on measuring the susceptibility of the model contour to clutter and instead measure clutter simply at the center of the object. To make sure that background contours T_{bg} are placed on the foreground contour P, where accidental matches typically occur, we create a mask for every combination of a foreground part and a background contour

$$M^{(T_{bg},P)}(\mathbf{x}) = 1 - d_{DCM}^{(T_{bg},P)}(\mathbf{x}) \tag{4.14}$$

These masks give high weight to regions where the background contour matches well on the part contour and low weight otherwise. Figure 4.6 shows the resultant masks for three different foreground bottle parts in combination with different background contours.

To describe the background matching costs for a hypothesis in a robust way we build weighted histograms over chamfer matching costs (see Figure 4.7). Let $\bar{\mathbf{x}}$ be one specific placement of the foreground object part P on the query image Q. Furthermore we define $B(\bar{\mathbf{x}})$ to be the bounding box region of P centered at $\bar{\mathbf{x}}$. For each foreground hypothesis we build weighted histograms $h^{(T_{bg},Q)}$ over the directional chamfer matching costs $d_{DCM}^{(T_{bg},Q)}$ in the corresponding bounding box region. The weights introduced in Equation 4.14 are used to weight the histogram votes according to their position relative to the foreground object part. Each histogram consists of K bins where $\mathcal{M}_k$ is the range of the kth bin and $k = 1, ..., K$. We define a histogram bin $h_k^{(T_{bg},Q)}$ as

$$h_k^{(T_{bg},Q)} = \sum_{\substack{\mathbf{x}\in B(\bar{\mathbf{x}}) \\ d_{DCM}^{(T_{bg},Q)}(\mathbf{x})\in\mathcal{M}_k}} M^{(T_{bg},P)}(\mathbf{x}), \tag{4.15}$$

for each background contour T_{bg} on a certain position of the foreground object part P in the query image Q.

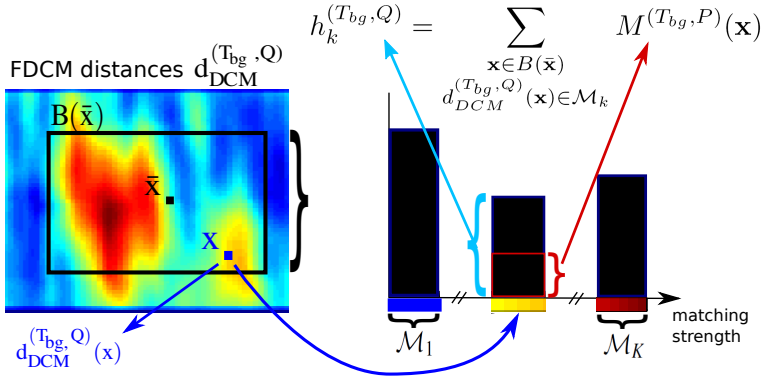


Figure 4.7: Background histograms $h_k^{(T_{bg},Q)}$ are constructed from the fast directional chamfer matching score maps. The sample scoremap on the left shows high matching scores of the background contour in the query image in red and low ones in blue. Each point $\mathbf{x}$ within the bounding box $B(\bar{\mathbf{x}})$ casts a vote with the corresponding weight of the mask $M^{(T_{bg},P)}(\mathbf{x})$. The vote is added to the histogram bin range $\mathcal{M}_k$ that corresponds to the directional chamfer matching score $d_{DCM}^{(T_{bg},Q)}(\mathbf{x})$.

4.3 Learning Chamfer Regularization

In order to exploit both the advantages of better foreground modeling of the template and the robustness against dense background clutter the suggested framework is integrating i) fast directional chamfer matching, ii) the co-occurrence of points on the foreground object part, and iii) the accidentalness of a match by means of co-occurrence patterns of background contours, into a single discriminative approach. Therefore, directional chamfer matching is regularized by learning the characteristic co-occurrence of foreground object part pixels and the joint placement of background contours using a support vector machine (SVM).

4.3.1 Learning Co-occurrences for Foreground and Background

In order to combine the line-matching costs $\bar{t}_l$ (Equation 4.13) and the weighted background histograms h_k (Equation 4.15) into a single max-margin framework, a new object representation is constructed by concatenating all line-matching costs L and all background histogram bins K of an object part for each object hypothesis j

$$\boldsymbol{f}_j = [\bar{t_1} \ldots \bar{t_L}\ h_1 \ldots h_K]. \tag{4.16}$$

The resulting feature vector for hypothesis j is denoted as $\boldsymbol{f}_j$.

$\mathcal{K}$ defines a kernel such that $\mathcal{K}(\boldsymbol{f}_i, \boldsymbol{f}_j)$ represents the similarity between feature vectors $\boldsymbol{f}_i, \boldsymbol{f}_j$. To model the joint co-occurrences of foreground and background contours we need to utilize a non-linear kernel that captures the relationship between foreground and

background pairs, triples, quadruples and so on. From the polynomial kernel

$$\mathcal{K}(\boldsymbol{f}_i, \boldsymbol{f}_j) = (\boldsymbol{f}_i^T \boldsymbol{f}_j + c)^2 = (\sum_{m=1}^{L+K} \boldsymbol{f}_i(m)\boldsymbol{f}_j(m) + c)^2. \tag{4.17}$$

of degree 2 one can easily determine that the mapping function ψ comprises all possible second order terms. It is straight forward, that a polynomial kernel of degree d comprises all possible combinations between feature dimensions up to degree d.

While polynomial kernels have finite mapping functions ψ the mapping function of the radial basis function (RBF) kernel has infinitely many dimensions. It is defined as

$$\mathcal{K}(\boldsymbol{f}_i, \boldsymbol{f}_j) = \exp\left(\frac{\|\boldsymbol{f}_i - \boldsymbol{f}_j\|^2}{2\sigma^2}\right) = \exp\left(-\gamma\|\boldsymbol{f}_i - \boldsymbol{f}_j\|^2\right). \tag{4.18}$$

Using the Taylor expansion one can determine the mapping function ψ of the RBF kernel. Since the Taylor expansion is a infinite set of features corresponding to polynomial terms it comprises an infinite amount of feature combinations. Applying the kernel trick [21] it is not necessary to explicitely represent the mapping ψ. Therefore, a kernel resulting in an infinite mapping function can be utilized. As the mapping function ψ is infinite the max-margin classification problem needs to be solved in its dual form

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \exp\left(-\gamma\|\boldsymbol{f}_i - \boldsymbol{f}_j\|^2\right) \\ \text{s.t.:} \quad & 0 \leq \alpha_i \leq C \quad \forall i. \end{aligned} \tag{4.19}$$

N is the number of training samples, b is the offset, C is the penalty and α_i are the Lagrangian multipliers. The solution to the classification problem given in Equation 4.19 maximizes the margin between positive and negative hypotheses in the transformed space. The resulting classifier

$$\delta(\boldsymbol{f}_j) = \sum_{i}^{N} \alpha_i \mathcal{K}(\boldsymbol{f}_j, \boldsymbol{f}_i) + b \tag{4.20}$$

has learned non-linear relationships between the features and models the joint co-occurrences of foreground and background contours. Since the mapping function ψ is infinite the explicit weighting $\boldsymbol{w}$ of the individual feature dimensions cannot be computed explicitly.

A crucial point in the training of every learning algorithm is the selection of training data. Since positive training data are typically rare, while negative training data are abundant, the selection of training data typically means a selection of negative training data. A common approach is to train the classifier using all positive samples and an initial random set of negative examples. This initial classifier is then used in a sliding window mode on all other images. Hypotheses with a high classification score are collected and added to the current set of negative support vectors. After retraining the classifier this procedure is repeated until some convergence criteria is met. Another common approach is to use another classifier or an objective measurement to generate hypotheses. In the

suggested framework the fast directional chamfer matching approach is used to generate good training hypotheses. We run the directional chamfer matching code [108] on the training images and label a hypothesis j positive $y_j = 1$, if it has an overlap greater than 80% with the groundtruth and a hypothesis with an overlap smaller than 40% is labeled as negative $y_j = -1$.

4.4 Object Detection using Regularized Chamfer Matching

In the previous section, we have described how the relevance of model points and the accidentalness, measured using background contours, can be jointly learned. Let us now utilize the combined model of foreground relevance and background accidentalness from Equation 4.20 to improve upon the directional chamfer matching cost function given in Equation 4.8. This improved, regularized chamfer distance $d^{(P,Q)}_{RDCM}(\mathbf{x})$ again measures the distortion cost of an object part. Let the j-th object hypothesis $\boldsymbol{f}_j$, which is described by the feature vector from Equation 4.16, be the placement of object part P at location $\mathbf{x}$ in the query image Q. Since a non-linear radial basis kernel is employed, the regularized chamfer distance is obtained using the dual SVM parameters, obtained by solving the dual SVM optimization problem from Equation 4.20,

$$d^{(P,Q)}_{RDCM}(\mathbf{x}) = 1 - \Big(\sum_i \alpha_i \mathcal{K}(\boldsymbol{f}_j, \boldsymbol{S}_i) + b \Big). \tag{4.21}$$

Each object part matched to a query image casts a vote with weight d_{RDCM} as computed in Equation 4.21 for different placements of the part in the query image. The votes from various parts are collected in a Hough accumulator and non-max suppression is performed to obtain final candidate hypotheses for objects.

4.5 Experiments

To demonstrate the utility of the proposed discriminative chamfer regularization, we evaluate our approach on benchmark datasets for chamfer matching. Since we integrated our regularization into the publicly available code of [108], the results reported in [108] have been used as the baseline. To demonstrate the advantage of our regularization over learning the normalization for chamfer distances [113], a comparison is made with the results documented in [113]. We also compare with a sophisticated learning and inference approach applied on object contours [187]. Furthermore, an analysis of the running time overhead caused by discriminative chamfer regularization compared to the running time of the chamfer matching approach of [108] is presented.

To extract the edge maps from input RGB images, we utilize the probabilistic boundary detector of [119]. The dual SVM optimization problem given in Equation 4.20 is solved using the support vector machine implementation of [28]. To measure the performance of our detection system, we employ the standard PASCAL overlap criterion according to

which a detection is correct if the ratio of intersection and overlap between groundtruth bounding box and the detected bounding box is larger than 50% (cf. Section 3.4.1).

The contribution of the proposed background regularization is presented in Section 4.5.3. As a baseline directional chamfer matching [108] is evaluated. Next the performance of foreground reweighting of the template pixels as in Equation 4.12 is evaluated and finally the performance obtained by the combined foreground and background regularization as in Equation 4.20 is determined. Section 4.5.4 compares the proposed regularization with other state-of-the-art extensions to chamfer matching such as [113, 187].

4.5.1 Datasets

For our experimental evaluation, we use the TUD Pedestrians, TUD Cows, and the ETHZ Shape datasets. These are the benchmark datasets for chamfer matching and approaches such as [108, 113, 152] report their results on one or more of these datasets.

TUD Pedestrians

The TUD Pedestrian dataset is a very challenging due to significant variation in clothing and articulation. Moreover, the background is rather complex and increases the chance of accidental matches. The TUD Pedestrian dataset [4] provides two training sets with 210 and 400 side-view pedestrians. Following the protocol of [113], we use the training set containing 400 images for training and the testset containing 250 images with 311 fully visible people for testing. Note, that the test images are significantly more challenging than the 400 training images. Since the dataset doesn't provide shape templates we are following the protocol of [113] and use the segmentation data given for the 400 training images to create part matching templates. Specifically we use 5 randomly selected segmentation masks from the training images and use them as model shape templates.

Cow Dataset

The Cow dataset from the PASCAL Object Recognition Database Collection [103] consists of 111 images in which cows appear with quite different articulation. The dataset is not providing a fixed separation of training and test data. Due to comparability we are following the protocol of Latecki et al. [113] to divide the dataset into training and testing sets. The first 55 images are used for training, and the remaining 56 images are used as testset. Next, the second half of the data is used for training and the first 55 images are used for testing. This way performance can be evaluated on the whole dataset. Similar to the TUD Pedestrian dataset 5 segmentation masks from the training images are obtained as the shape templates.

ETHZ Shape Dataset

The ETHZ shape dataset is designed for testing object class detection algorithms. It contains 255 images and features five diverse shape-based classes (apple logos, bottles,

giraffes, mugs, and swans). It is highly challenging, as the objects appear in a wide range of scales, have high intra-class shape variation and appear in cluttered background. We are following the standard protocol for the dataset and use one half of the images of all classes for training and the remaining images for testing. Additionally one hand-drawn example is provided with the dataset along with each category which is used as a shape template. For the object categories applelogos, bottles and swans,the template is decomposed into four parts while for the categories giraffes and mugs the full template was utilized.

4.5.2 Running Time

To obtain the initial matches for the templates, we run the publicly available directional chamfer matching code of [108] using the default parameters for all the datasets. In our experimental evaluations, we have observed that computing the distance transformation of a query image for each angular quantization is the most time consuming part in the code of [108]. The proposed chamfer regularization added only a marginal overhead to the computation time. For instance, only 2 second overhead is observed per image from TUD Cow dataset. On the other hand, computations for the baseline performance [108] took about 15 seconds per image. Thus, our approach turns out to be easily integrable into a state-of-the-art chamfer matching approach, without adding significant overhead in terms of running time.

4.5.3 Evaluating Background Regularization

We are evaluating the object detection performance of the suggested regularized chamfer matching approach using *average precision*. The average precision is the area under the curve of the precision/recall curve. The precision/recall curve is computed from a method's ranked output. Recall is defined as the proportion of all positive examples ranked above a given rank. Precision is the proportion of all examples above that rank which are from the positive class.

Evaluation of regularized chamfer matching on the ETHZ dataset shows that for 4 out of 5 object categories it is helpful to measure the accidentalness of a match utilizing the suggested background regularization. In particular, this method is helpful for a challenging category like Giraffes with articulations and background clutter. We observe 6% improvement in average precision of foreground reweighting over directional chamfer matching. Extending foreground regularization to avoid accidental matches using background regularization is improving the performance by another 9% in average precision.

Table 4.1 compares the baseline directional chamfer matching, which constitutes the basis of our approach, with the different components of our discriminative chamfer regularization. In particular, first the performance of foreground regularization method in combination if directional chamfer matching is evaluated. Next the performance of the final detector integrating the suggested background regularization together with directional chamfer matching and foreground reweighting is evaluated. The experiments show that foreground regularization alone improves performance in terms of average

	Applelogos	Bottles	Giraffes	Mugs	Swans	**mean**
DCM [108]	60.8	85.5	27.0	10.1	33.1	43.3
FG Regularization	62.0	86.9	36.3	**27.3**	33.8	49.3
Combined Regularization	**81.8**	**90.4**	**43.0**	**27.3**	**47.3**	**58.0**

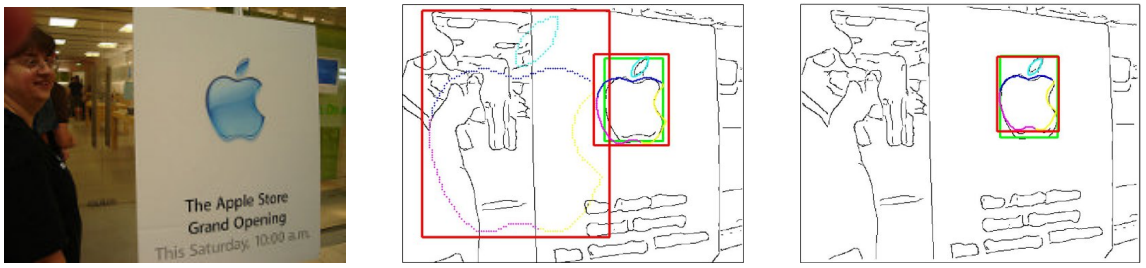
Table 4.1: Comparison of **average precision** for the ETHZ Shape classes. We compare DCM [108] which constitutes the basis of our approach with the extension from Section 4.2.1 and our final learning of regularized chamfer matching. All the detections are evaluated based on PASCAL overlap criterion with the groundtruth object annotations.

	Pedestrians	Cows
DCM [108]	3.0	88.1
FG Regularization	6.8	89.2
Combined Regularization	**11.2**	**91.9**

Table 4.2: Comparison of **average precision** for two datasets namely, TUD Pedestrians, Cows. We compare DCM [108] which constitutes the basis of our approach with the extension from Section 4.2.1 and our final learning of regularized chamfer matching. All the detections are evaluated based on PASCAL overlap criterion with the groundtruth object annotations.

precision on all of these object categories compared to directional chamfer matching. Applying the background regularization in addition to foreground reweighting suppresses false positives in cluttered background and, thus, yields a significant further gain.

For the TUD Pedestrian dataset the images in the testing set are provided at a very high resolution which yields very low average precision for the directional chamfer matching which is around 3%. The low baseline can be attributed to the high resolution of the test images, since it is known that chamfer matching is sensitive to all the fine details in the edge map. While foreground regularization shows significant improvement (3.8%) over the baseline in average precision, adding the background regularization brought a further gain of 4.4% in average precision. For the Cow dataset directional chamfer matching yields very good performance around 88% average precision. Nevertheless, the combined detector still improves the performance about 4% by exploiting the advantages of foreground and background regularization. These results show that measuring the accidentalness of a match using the suggested background regularization is of equal important as improving the foreground template by reweighting the pixels of the template. This is due to the fact, that the two approaches are solving different problems of the chamfer matching approach. While foreground reweighting is further improving the template, which improves alignment of object detections with the groundtruth (see Figure 4.4), the background normalization avoids accidental matches in dense clutter. The example in Figure 4.8 shows that foreground regularization is not always able to suppress false positives in cluttered background and how background regularization can handle such cases. All in all, our combined detector using foreground and background



a) b) c)

Figure 4.8: This example shows how combined foreground and background regularization Equation 4.20 can remove false positive detections which could not be eliminated by foreground reweighting alone. Panel a) shows the original image, b) the result obtained by using foreground reweighting and c) the results from the combined foreground and background regularization.

	Cows	Pedestrians
OCM [152]	73.9	35.2
NOCM [113]	91.0	70.0
HDT [187]	88.2	-
Regularized Chamfer Matching	**98.3**	**80.0**

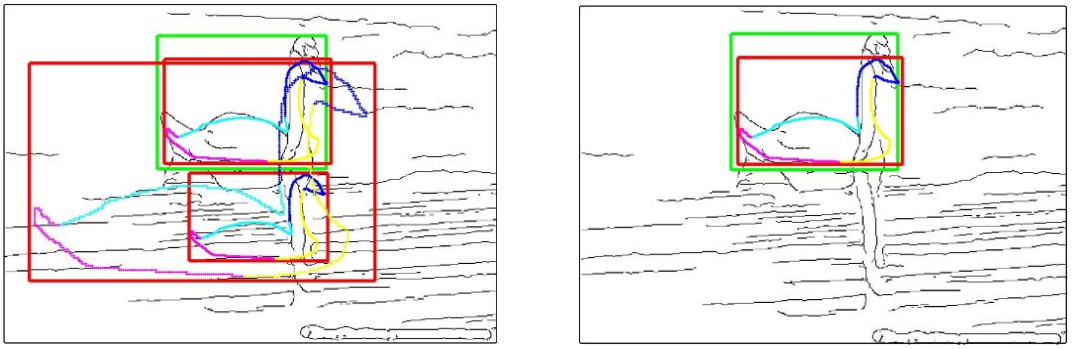
Table 4.3: Comparison in terms of **detection rate** at 10% precision (in %) on the Cow dataset and the TUD Pedestrian dataset with OCM, NOCM and HDT.

regularization achieves significant gain on all of the seven categories compared to directional chamfer matching and foreground regularization. Additional detection results comparing the regularized chamfer matching to directional chamfer matching are provided in Figure 4.9.

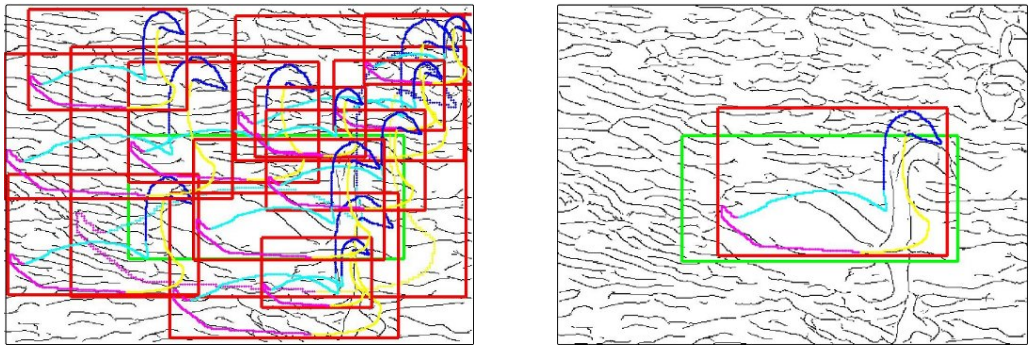
4.5.4 Comparison with State-of-the-Art Extensions to Chamfer Matching

We compare our combined foreground and background regularization with other state-of-the-art extensions to chamfer matching such as the normalized oriented chamfer matching by Ma et al. [113] (NOCM) and the hierarchical deformable template model (HDT) by Zhu et al. [187].

In [113] Latecki et al. have reported results on two datasets: the TUD Pedestrian dataset [4] and the Cow dataset [103]. In [187] Zhu et al. have evaluated their method on the Cow dataset. Both approaches report their results in terms of detection rate at 10% precision. In the previous section, we have reported the gain obtained by our regularization in terms of average precision, since it is taking into account the area under the precision recall curve instead of just one point on the performance curve and therefore is a more robust measure. Nevertheless, to compare ourselves with [113, 187], we need to report results in terms of detection rate at 10% precision.



a)



b)

Figure 4.9: Panel a) and b) show detection results for two examples. The left image of each panel shows results obtained by directional chamfer matching. The right image of each panel shows the improved detection result after applying chamfer regularization. The groundtruth bounding box is shown in green and the top scoring object hypotheses are shown in red.

Table 4.3 shows the results for the Cow dataset and the TUD Pedestrian dataset. The results indicate that chamfer regularization significantly improves performance on the Cow dataset compared to HDT and NOCM. For TUD Pedestrians we gain 10% in detection rate compared to NOCM. All in all, our results confirm that the regularized chamfer matching method significantly improves over state-of-the-art extensions to chamfer matching.

4.6 Discussion

This contribution extends the well established and widely used chamfer matching technique, particularly by overcoming its susceptibility to clutter. Our results confirm, that making the template more specific by learning the co-occurrence of model points is increasing the specificity of the template. However, to avoid matches in dense clutter only improving the template is not enough. To suppress false positive matches in background clutter measuring the accidentalness of match is crucial. By placing generic contours on

the model contour and learning to distinguish the typical co-occurrence of these contours on cluttered background compared to actual objects, performance improves significantly. Furthermore, foreground and background regularization are integrated in a max-margin learning framework which is based on state-of-the art directional chamfer matching.

CHAPTER 5

RANDOMIZED MAX-MARGIN COMPOSITIONS FOR VISUAL RECOGNITION

As already discussed in Section 2.3.1 part-based models currently constitute one of the most popular and powerful paradigms for the challenging problem of category-level object detection. In the previous chapter it was shown, how mid-level representations are a useful scheme to further improve such part-based models. However, while chamfer matching is a very efficient approach, parts are detected based on a simple distance measure and therefore have less discriminative power than parts that are learned by a discriminative classifier. Furthermore, the approach suggested in the previous chapter is organizing parts in a flat star-model. This chapter is investigating in the usage of discriminatively trained parts which are arranged in a more powerful hierarchical framework utilizing random compositions.

Such discriminative parts can be learned in different frameworks. Typically, powerful discriminative approaches, such as the deformable part model [56], are combining a small number of parts based on their appearance and location. This framework typically restricts such discriminative methods as [56, 188] to only few parts, as opposed to weaker spatial models such as bag-of-features [36], Hough voting [116], or generative methods such as [51, 97, 139].

In contrast we aim for a large number of specific but weak parts (on the order of 1000 per category) that are trained in the spirit of currently popular paradigm of mid-level patches and parts (Section 2.4.3). Each part is trained on only a small region of a single positive sample against negatives. In contrast to other part-based methods, such as [46, 89, 117, 153], we compensate for the weakness of specialized, local, and frail parts by grouping them into stronger compositions that exhibit improved generalization ability. Compositionality [88, 127] is a powerful mid-level representation (Section 5.2.1) that reduces the representational complexity to render learning of structured models feasible.

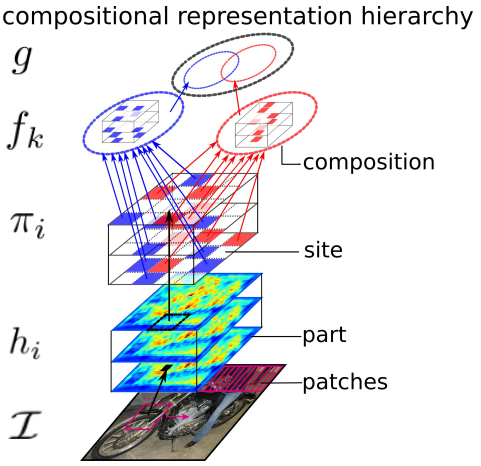


Figure 5.1: The compositional representation hierarchy shows the individual stages of the classifier hierarchy: part classifiers h_i, max-pooled responses π_i, compositions f_k and final non-linear object classifier g.

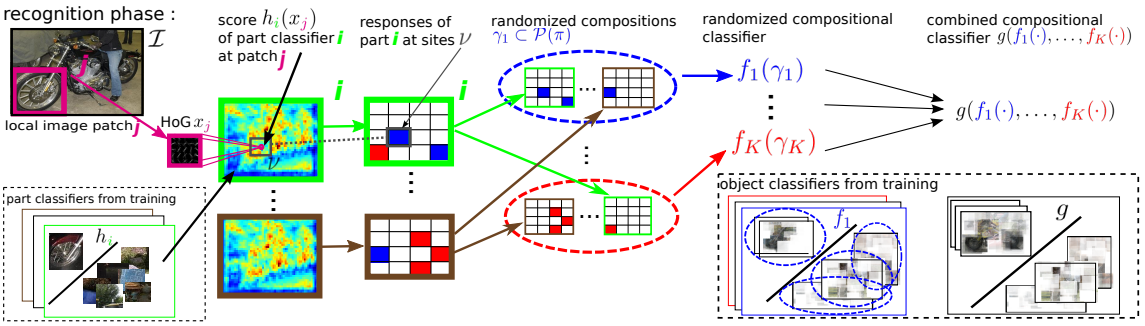


Figure 5.2: Shows the detection procedure of our randomized max-margin compositions. Part classifiers responses are pooled at different locations before aggregating them in randomized, discriminatively trained compositions. All compositions then join in a final combined classifier $g(\cdot)$.

We deviate from the common rationale of compositional hierarchies [61, 88, 96, 128, 139] that establish meticulously arranged, semantically meaningful compositions. Rather we show that multiple overlapping *randomized* compositions trained using a max-margin approach generalize significantly better to new category instances compared to the original parts and thus yield improved performance. Compositions are then all combined by a final non-linear decision function in a third layer of this hierarchy of discriminative classifiers, with part classifiers and the compositional classifiers in the two preceding stages. Figure 5.1 gives an overview of the classifier hierarchy and Figure 5.2 summarizes the detection procedure.

We thoroughly evaluate the individual contributions and crucial modeling decisions of our model. Experiments are conducted on the well-established, competitive benchmark detection challenges of PASCAL VOC 2007, using the VOC 2010 evaluation server [49], and on the challenging MITIndoor scene recognition dataset of [137]. Our randomized max-margin compositions (RM^{2}C) show, to the best of our knowledge the currently best

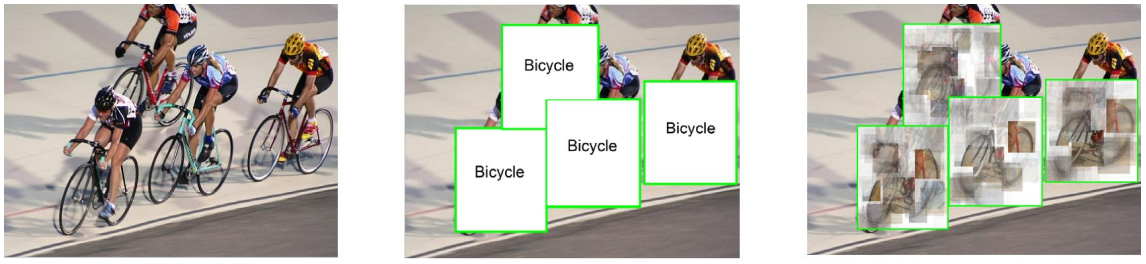


Figure 5.3: Object detection and parsing with randomized max-margin compositions (RM^{2}C). The discriminative approach not only detects objects, but also activates compositions according to the classification function $g(F(\mathcal{I}))$. Compositions in turn activate parts i (we plot the corresponding positive training patch x_p) by weighting them according to the decision function f_k.

performance using only HOG features for single class object detection, i.e. without any postprocessing exploiting interactions of multiple object classifiers trained for different classes.

Moreover, the experimental analysis underlines the necessity of large numbers of specific parts because of their mutual unrelatedness and low generalization ability. We also observe that randomly sampling compositions significantly outperforms individual parts, a location based part grouping, and a clustering based on visual similarity. Finally, we show that our approach not only localizes object bounding boxes, but that, although being discriminative, it parses their content to thoroughly explain a test object with the randomized compositional model (cf. Figure 5.3). We then propose a novel evaluation setup for part-based models on PASCAL VOC 2010 that allows measuring the accuracy of arbitrary individual parts. This new experimental protocol is crucial to thoroughly evaluate the intermediate components of hierarchical part-based methods.

The remaining of this chapter is organized as follows: First an overview is given on state-of-the-art part-based models for object and scene recognition (Section 5.1), then a novel compositional approach to discriminative part-based recognition is suggested in Section 5.2 and finally experimental evaluation for object (Section 5.3) and scene recognition (Section 5.4) is executed.

5.1 Part-Based Models

In this section a short overview is given on state-of-the-art part-based recognition approaches for objects and scenes and will illustrate the contributions of the suggest randomized max-margin compositions.

5.1.1 Object Recognition

A popular and powerful approach for discriminative part-based object recognition is the deformable part model (DPM) suggested by Felzenszwalb *et al*. [56]. The model trains a

latent support vector machine to discover the hidden locations of a fixed number of parts. Zhu *et al*. [188] extended this idea and suggested a deeper hierarchy of parts which is trained using a structural SVM. Recently Song *et al*. [154] suggested a discriminative and-or tree model to automatically learn the configuration of parts. Since the spatial configuration needs to be learned in the training phase, the number of parts is quite restricted. This results in a small set of very general parts that typically correspond to a whole aspect. Contrary to this, our framework is able to handle a very large number of specialized parts. Due to the great number of parts, our approach can not only detect object bounding boxes but also provides a parsing of its content (see Figure 5.3). Endres *et al*. [46] are avoiding a structured model and use a simple method that pools part responses over proposed object regions with a boosting classifier. Similarly to our approach they start by using part-based exemplar SVM [117]. However, one of the main challenges solved in [46] is how to refine these simple but specialized classifiers to get a smaller more general set of part-classifiers. In addition there has been work on incorporating strong supervision to train part-based object detection models, such as [6] and [23], and on different classifiers such as Random Forests [26].

Furthermore, our approach is related to compositional hierarchies [88, 127] which have been proposed to bridge the large gap between local features or parts and the whole object. The fundamental goal is to establish one or more successive representational layers by grouping parts, thus obtaining a hierarchy of successively larger and more meaningful compositions [61, 88, 96, 139]. In contrast to this delicate assembly of compositions, which is common to these approaches, we show that randomized discriminative compositions are ideal for robust aggregation of specialized parts, thus yielding significant performance improvements.

5.1.2 Scene Recognition

Part-based approaches are recently also becoming more popular for scene classification. Pandey *et al*. [130] adapted the deformable part model for scene classification. On the other hand, there are holistic representations such as object bank [106] that require a supervised training of object classifiers. Similar to the discriminative training of intermediate compositions in [128], Singh et al. [153] train mid-level patch classifiers. Juneja et al. [89] followed this idea but started from individual exemplar SVM classifiers which are used to mine more positive samples instead of performing an unsupervised clustering as in [153]. Since in [153] and [89] parts are discovered in an unsupervised manner they need to solve the problem of finding a good positive training set for parts using clustering, positive mining etc. which is as difficult as the scene classification problem itself. Therefore, our aim is not to make parts more general, but rather to train compositions that generalize better than the specialized part classifiers they aggregate.

5.2 A Compositional Approach to Discriminative Part-Based Recognition

Let us assume for now that we have semi-local features and part classifiers that are specifically trained for individual instances of an object category. We discuss the training of these parts in Section 5.2.3 and provide the classifiers on the project site [1]. Due to the specific nature of such parts, a large number of them is necessary to capture all relevant characteristics of complex object categories. However, training a powerful discriminative model, *e.g.*, a non-linear classifier, on a limited training set, is not feasible based on the high-dimensional combination of a large number of parts. To avoid overfitting we aggregate parts in fewer, overlapping compositions, each capturing a previously learned, random set of parts. These compositions, that can be shared across instances of a category, are all gathering different observations due to the random selection of parts and thus generalize better to novel samples. Section 5.2.1 presents our compositional model before discussing part classifiers and their training in the following sections.

5.2.1 Randomized Max-Margin Compositions

Assume we have already trained a large set of part classifiers (typically around $P = 1000$ per category), which will be described in Section 5.2.3. For some image site ν the classifier of part i is evaluated densely within this region and the detection scores are pooled yielding a response $\pi_i(\nu) \in \mathbb{R}$ as will be discussed in Section 5.2.2 . At each image site all parts are evaluated. The common approach is then for all sites $\nu \in \mathcal{I}$ on a regular grid within an object bounding box $\mathcal{I}$ to concatenate all part responses. Given the large number of parts this would yield a very high dimensional representation (on average far beyond $20\,000$-D). In light of the curse of dimensionality, learning object models with this high dimensional representation on a limited set of positive training samples (for PASCAL VOC typically on the order of 100) is inappropriate. One might speculate that there is significant redundancy when a large number of part classifiers is applied to an object, so that grouping related parts or subspace methods could significantly reduce dimensionality. However, since each part classifier represents a single positive object region (Section 5.2.3), we observe that their responses are highly uncorrelated (cf. Figure 5.4). Consequently, applying principle component analysis, 90% of the original dimensionality retains only about 40% of the variance. The $\pi_i(\cdot)$ are essentially trained to act as specialists, each specifically trained for an individual part instance from training. Therefore, we propose to group the responses of all parts i at sites ν to create K groups of part responses $K << P$. Each comprises a large number of part responses and thus generalizes better than individual parts to the large number of instances from an object category. More precisely, let $\pi := \{\pi_i(\nu), \forall i, \nu\}$ and $\mathcal{P}(\pi)$ be the powerset of all responses then we seek K compositions $\gamma_k \subset \mathcal{P}(\pi)$. When applying a composition to a candidate object bounding box $\mathcal{I}$, we obtain a $|\gamma_k|$-dimensional response $\gamma_k(\mathcal{I})$. Following upon the part classifiers, the groups establish a second level in a classifier hierarchy. To render the learning problem feasible, this second level is comprised by linear classifiers

[1] hci.iwr.uni-heidelberg.de/COMPVIS/research/RM2C

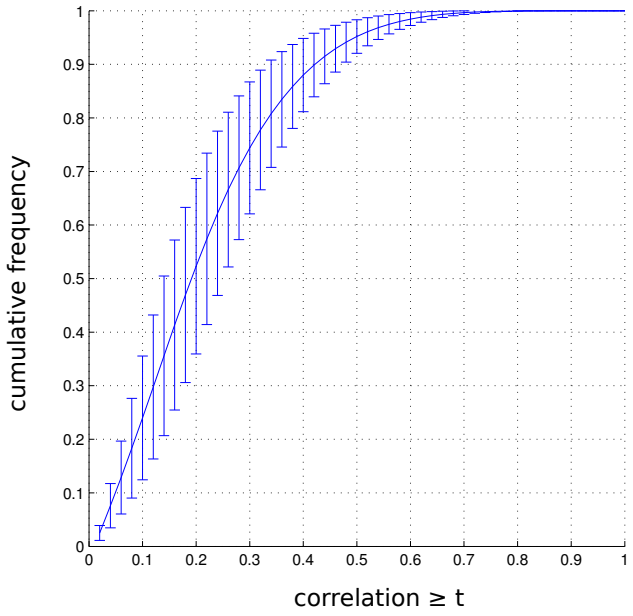


Figure 5.4: Maximal absolute correlation of a part to any other part evaluated over all categories of VOC 2007. Most parts are highly uncorrelated, i.e. 95% of parts have a correlation of less than $.5$ to any other.

$f_k(\cdot)$ trained with hinge loss in a max-margin fashion,

$$\min_{\mathrm{w}} \frac{1}{2}\|\mathrm{w}\|_2^2 + C \sum_{\mathcal{I}\in\mathcal{T}} \max(0, 1 - y_{\mathcal{I}} f_k(\gamma_k(\mathcal{I}))) \tag{5.1}$$

where $f_k(\gamma_k(\mathcal{I})) = \mathrm{w}_k^T \gamma_k(\mathcal{I}) + \mathrm{b}_k$, $\mathcal{T}$ denotes the set of training bounding boxes and $y_{\mathcal{I}} \in \{-1, 1\}$ is the class label of the bounding box $\mathcal{I} \in \mathcal{T}$. Now the questions remains, how to obtain the γ_k. From the experiment in Figure 5.4 we see that the appearance-based part responses $\pi_i(\nu)$ at locations ν are uncorrelated. Without any extra annotation as in [23] we can from this experiment already suspect that an unsupervised grouping of parts based on their appearance and location will not be desirable. And indeed, combining parts based on similarity in appearance and location using agglomerative clustering (Wards method) does not yield a significant improvement of groups compared to their constituent parts. We experimented with different grouping strategies and measured the performance of the first level compositional classifiers $f_k(\cdot)$ in terms of average precision on a validation set. Figure 5.5 shows the cumulative frequency of group classifiers $f_k(\cdot)$, i.e., the fraction of classifiers that succeed a certain average precision. When grouping parts based on their location we observe little gain over the baseline of singleton part groups. An agglomerative clustering based on visual similarity yields a larger improvement over the individual part performance. To achieve a further significant gain we propose to randomize the formation of compositions. Therefore mutually overlapping part response vectors γ_k are drawn randomly from $\mathcal{P}(\pi)$. To simplify their subsequent combination, we demand all γ_k to have a fixed size $|\gamma_k| = L$. Crossvalidation has shown L=3000 (part,location) pairs to yield optimal performance, but the fluctuation within reasonable

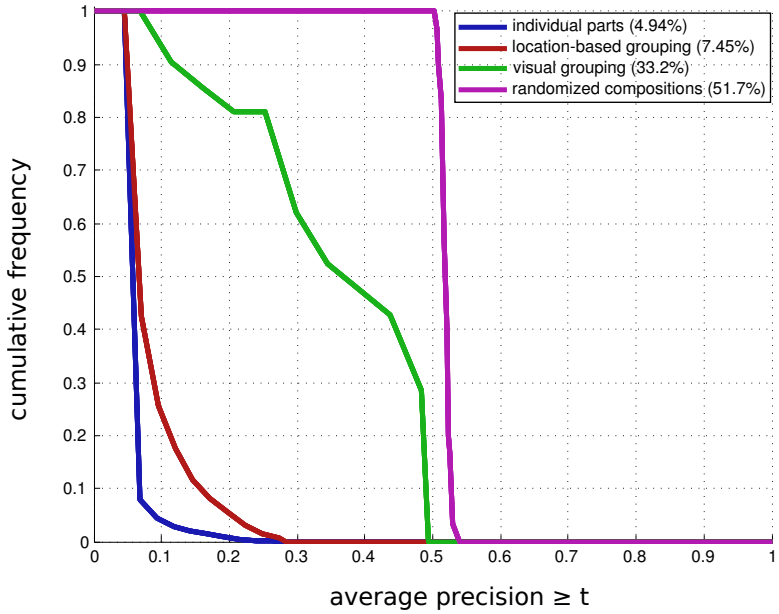


Figure 5.5: Comparing different grouping strategies for assembling compositions on VOC 2007 *bicycle*. Cumulative frequency of group classifiers $f_k(\cdot)$ w.r.t. their average precision.

range was insignificant. Figure 5.5 shows that randomized compositions generalize significantly better than clustering parts based on their visual similarity. One might conclude that randomization avoids overfitting by not using visual information twice, i.e., for defining the part classifiers and for clustering them based on visual similarity.

Now we have a manageable number of compositions, each being significantly more informative than the large number of initial parts. Thus, training a non-linear classifier $g(f_1(\cdot), \ldots, f_K(\cdot))$ that establishes a third level in the already existing hierarchy of classifiers becomes feasible. Let $F(\cdot) = (f_1(\cdot), ..., f_K(\cdot))^\top$ be the low dimensional feature descriptor that concatenates the K decision values (we use $K = 50$) $f_k(\cdot)$ of the second level group classifiers. The final third level classifier is then trained by optimizing

$$\max_{\alpha} \sum_{\mathcal{I} \in \mathcal{T}} \alpha_{\mathcal{I}} - \frac{1}{2} \sum_{\mathcal{I} \in \mathcal{T}} \sum_{\mathcal{I}' \in \mathcal{T}} \alpha_{\mathcal{I}} \alpha_{\mathcal{I}'} y_{\mathcal{I}} y_{\mathcal{I}'} \kappa(F(\mathcal{I}), F(\mathcal{I}')) \tag{5.2}$$

with the radial basis function (RBF) kernel given by

$$\kappa(F(\mathcal{I}), F(\mathcal{I}')) = \exp\left(-\frac{\|F(\mathcal{I}) - F(\mathcal{I}')\|_2^2}{2\sigma^2} \right). \tag{5.3}$$

The decision function is

$$g(F(\mathcal{I})) = \sum_{\mathcal{I}' \in \mathcal{T}} \alpha_{\mathcal{I}'} y_{\mathcal{I}'} \kappa(F(\mathcal{I}), F(\mathcal{I}')). \tag{5.4}$$

5.2.2 Part Responses on Image Sites

Evaluating a part classifier i only once per image site ν leads to noisy results, since the regular spatial grid of sites is too coarse to deal with local deformations. If a part in an

image would be shifted or scaled, so that it is not aligned with a site ν we might miss it. Therefore, we follow common practice and sample local features x_j densely using a sliding window at all locations/scales $j \in \nu$ within sites. To get the sites we use regular grids of size 1×1, 2×2 and 4×4. As feature we use HOG and for the j we use the location/scale pyramid of [56]. As a result we obtain classifier scores $h_i(x_j)$ for each part (cf. Section 5.2.3). The part response to a site is then defined by max pooling over all locations/scales within ν,

$$\pi_i(\nu) = \max_{j \in \nu} h_i(x_j). \tag{5.5}$$

This aggregation of part responses on a spatial grid has been shown to work well in different vision problems [100, 106, 153].

5.2.3 Learning Parts without Part Annotation

Learning part models without annotation of parts is a challenging problem. Without extra annotation, the task of finding corresponding parts in different object bounding boxes turns out to be as difficult as finding the object itself, since the locality of parts leads to ambiguities. Thus parts are typically detected conjointly, linked by a spatial model that enforces spatial consistency. However, when learning a part, we have neither an object model provided nor any other parts. Thus, finding all instances of a part in all training images is daunting. And indeed it was shown that clustering based on the distance of features (*e.g.* HOG) is not very reliable [79, 153]. The problem is then that incorrect groups of parts at this initial stage will lead to mistakes that accumulate during later stages. We therefore train part models with just a single positive sample and a set of negatives as suggested by [117]. To obtain the positive part samples we randomly select a large number of patches at different locations and scales within training bounding boxes. All parts together should exhibit a good coverage of all training images. Therefore, we do not want to get very similar patches with high overlap in the same bounding box and therefore restrict the overlap between sampled patches in the bounding box to be less than 20%. Additionally we restrict the number of parts per box and sample a maximum of 20 parts. Note, that significantly less parts maybe sampled if the object bounding box is very small. Now we have one positive sample x_p per part, and similar to [117] we perform negative mining on up to 2500 images to obtain a set of negatives $\mathcal{N}$. The corresponding classification function h_i is

$$\min_{\omega} \frac{1}{2}\|\omega_i\|_2^2 + C_1 \max(0, 1 - h_i(x_p)) + C_2 \sum_{x \in \mathcal{N}} \max(0, 1 + h_i(x)) \tag{5.6}$$

were $h_i(x) = \omega_i^T x + \beta_i$. The part features x are HOG descriptors [37] using 25 cells that are fitted to the part as in ESVM [117]. The number of pixels per cell depends on the scale on which the part was sampled. The minimum cell size is 4 pixels. In our framework the trained exemplar SVMs act as specialized parts. One might think that a part classifier trained on one positive sample is overfitting badly and therefore performance of the individual parts might be very poor compared to more general parts using a larger set of positive training samples. To get an idea of the quality we are evaluating the individual performance of the part classifiers in the next section.

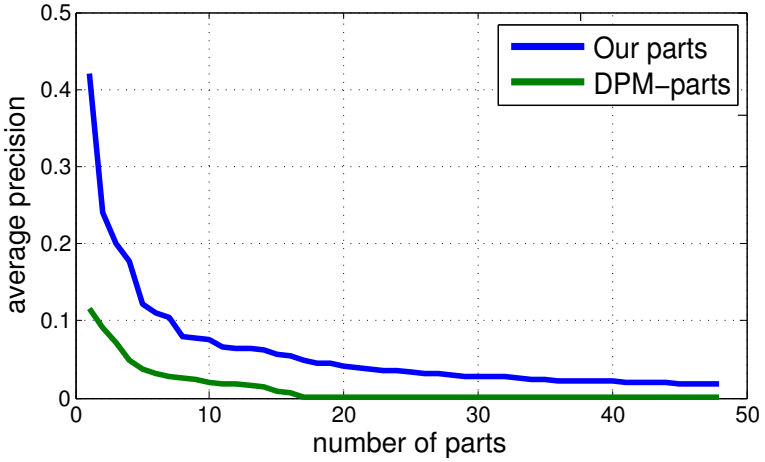


Figure 5.6: Performance comparison of the 48 DPM parts with our randomly sampled parts (also a subset of 48) in terms of average precision, see Section 5.2.4

Recognition Phase To perform object detection in a novel test image (see Figure 5.2) we first need to extract HOG descriptors x_j and run part classifiers $h_i(x_j)$. Then we pool part responses using Equation 5.5 into $\pi_i(\nu)$ before running the composition classifiers $f_k(\cdot)$. Responses from the composition classifiers are concatenated to the final feature vector which is evaluated using $g(F(\cdot))$ to combine all compositions using the non-linear classifier.

5.2.4 Part Evaluation

To evaluate the performance of our part classifiers we are using the keypoint annotation of [22] for the PASCAL 2010 dataset. However, in contrast to poselets this is here merely for our subsequent evaluation and not for training. Since our parts are trained in an unsupervised manner using HOG features we are comparing the performance of our parts to those of the Deformable Part Model (DPM) [56] which are using a similar setup. In contrast to our parts the DPM parts are much more general since they are trained on all training images from an aspect of a category.

To evaluate the detection performance of individual parts we first need to generate ground-truth on which we can test. In contrast to [23] there are no annotations specific to our parts, but the idea is to measure how much a part shifts between training and testing relative to the existing keypoint annotation of [23]. For the positive training sample x_p that defines the part we therefore measure its euclidean distances to all keypoints within the object bounding box. During detection we again compute the distances to the same keypoints. Comparing the training and test vector of keypoint distances thus defines a similarity measure. Now we can rank parts according to their mean average precision, i.e., how good they are in detecting a similar object region as they were trained upon, where similarity is measured with respect to annotated semantic landmarks from [23]. Figure 5.6 compares the 48 DPM parts [56] with the randomly sampled parts from Section 5.2.3 (also a subset of 48). We observe that in the large pool of weak parts there is still a sufficient number of parts that have favorable detection performance compared to the DPM parts.

5.3 Object Recognition Results for PASCAL

In our experiments we are providing object recognition and scene classification results on three of the most challenging datasets. For object recognition we are evaluating our approach on PASCAL VOC 2007 and 2010 . The scene classification results are evaluated on the MITIndoor dataset [137]. Our experimental results show competitive performance to recent state-of-the-art part based approaches on all datasets. We follow the standard training and testing protocols for the PASCAL detection challenge only using provided bounding box annotation on the object category level. Additionally we are showing qualitative results in terms of a back-rendering of our training parts in the detection box to visualize how our model is explaining objects (cf. Figure 5.10 and 5.11).

5.3.1 Implementation Details

Training

Since we are training classifiers on a part level and on an object level we need to split the training data, to avoid over-fitting. Considering our part classifiers are trained in an exemplar fashion over-fitting is not an issue on the rare positive samples as one part classifiers is only over-fitting in one image at a certain location and scale. The sampling of positive patches described in Section 5.2.3 can therefore be performed on the whole trainval set. Since each part classifier is performing a negative mining, the part classifiers might over-fit when we are applying them on the same negative images again to get the response maps. Therefore we are only using 2500 negative images from the PASCAL training data for the negative mining procedure of the part classifiers. To train the object classifiers (i.e. $f_k(\cdot)$ and $g(\cdot)$) we use all the positives form the trainval set and all negative images remaining after training the part classifiers. To get a set of hard negative samples we apply the deformable part model with a low threshold (-1.1) and use the resulting false detections. Note, that we use the same models and parameters for hypothesis generation at detection time. For training the SVM classifiers we use LIBSVM to train non-linear classifiers and otherwise LIBLINEAR [50].

Part Selection

Since we are sampling an over complete set of parts the number of parts can be extremely large for classes with a lot of objects like the person category. This raises the question if all of these parts are actually needed. Therefore we perform an experiment where we use an increasing number of parts (in steps of 100 parts) for training and evaluate the performance on the validation set. Note, that since we evaluate on the validation set only the training data are used to train our framework. We order the parts according to their strength based on the absolute weights of a linear SVM classifier trained on the maximum response of each part per training sample. For each of our evaluations we are using the best N parts for training. Figure 5.7 shows that the mean average precision is saturating around 1000 parts. This confirms that a large number of part classifiers is actually needed. One could think that the reason this high number of parts are needed is because the individual

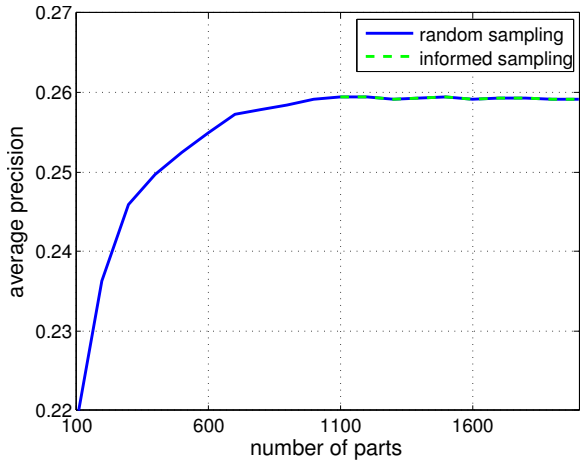


Figure 5.7: Mean average precision of all classes of the PASCAL VOC2010 dataset, on the validation set, training our model with different number of parts added randomly (blue) and using false negatives (green).

performance of our exemplar-based parts is very weak. However, as we were discussing in detail in Section 5.2.4 and is shown in Figure 5.6 a subset of our part classifiers is even performing better than the DPM parts. Based on these results we are selecting the subset of parts for each category with the highest performance on the validation set.

Now, although our object detection system performs well using random parts the questions remains if parts can be sampled in a more sophisticated manner using guidance of the current system. Therefore we perform cross-validation on the training data to identify which positive samples are especially hard (false negatives) for our object detection system. To get a better description of these false negatives we are getting additional positive patches from them and train new part classifiers. Since this new part classifiers describe positive samples that have been classified incorrectly the coverage of the positive samples should improve and therefore the overall object classifier should exhibit better generalization ability. However, in Figure 5.7 we can see, that adding additional parts randomly is as good as adding new parts from false negatives. Qualitative coverage results of our randomly sampled part classifiers are provided in Figure 5.8.

Detecting with all these classifiers may seem very time consuming. However, the filter operation is just a single dot-product for all the part classifiers. Creating the response maps for 1000 part classifiers takes around 13 seconds. For comparison the DPM [56] takes 7 seconds to create response maps for 54 object and part classifiers. The reason for the comparably small overhead of our system is that the time needed to build HOG features and extract detection windows for an image is significantly higher than the detection time. Therefore, the more filters are used the more favorable it is to first extract HOG features for all windows and perform a single matrix multiplication than performing a separate convolution for each filter as done by the DPM.

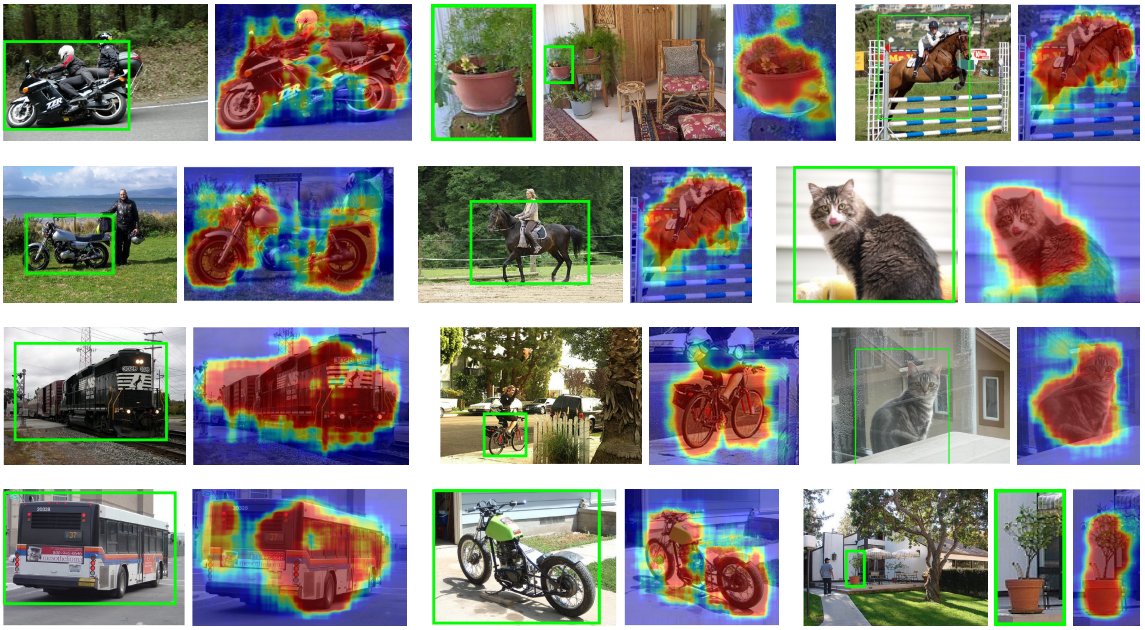


Figure 5.8: The coverage image on the right shows which pixels are covered by part detections. For each part the best detection in the hypothesis is selected and all pixels covered by the part detection box are weighted according to the detection score. For each pixel the weights of all part scores are summed up and normalized.

5.3.2 Comparison with other Methods

Since we suggest a part-based approach the focus of our evaluation is to compare with other part-based approaches. There exist several methods such as [31, 155] that focus on how the responses of several classifiers can be used to improve overall detection performance. These methods can be applied in a post-processing step for any part based method. Therefore part-based methods are evaluated without context in common literature.

PASCAL VOC 2007

Our final approach (RM^{2}C) is also incorporating parts that are root filters. Our results show that the suggested approach already gives state-of-the art performance without applying larger parts corresponding to objects (RM^{2}C w/o obj.). Additionally we are comparing our approach to three other part-based approaches. All approaches are utilizing HOG features as a low level representation. The detection results are summarized in Table 5.1. Our method outperforms all other approaches on 17 out of 20 categories. Significant improvements are reached on articulated objects as dogs (8.1%), cats (6.5%) and birds (2.5%). However, also more rigid objects with high intra class variability benefit from our specialized part-classifier compositions as aeroplanes (4.5%) and tvmonitors (2.5%). In mean we are gaining 1.9% over the And-Or Tree (AOT), 2.9% over the Deformable Part Model (DPM) and 7% over the Latent Hierarchical Structures (LHS).

	DPM rel5 [56]	LHS [188]	AOT [154]	RM^{2}C w/o obj.	RM^{2}C
aeroplane	33.2	29.4	35.3	37.0	**37.7**
bicycle	60.3	55.8	60.2	58.3	**61.4**
bird	10.2	9.4	9.4	12.0	**12.7**
boat	16.1	14.3	16.6	14.7	**17.6**
bottle	27.3	28.6	29.5	22.9	**29.9**
bus	54.3	44.0	53.0	51.3	**55.1**
car	**58.2**	51.3	57.1	51.7	56.3
cat	23.0	21.3	23.0	23.7	**29.5**
chair	20.0	20.0	22.9	21.7	**24.6**
cow	24.1	19.3	27.7	25.0	**28.2**
table	26.7	25.2	28.6	29.0	**30.7**
dog	12.7	12.5	13.1	20.6	**21.2**
horse	58.1	50.4	58.9	51.4	**59.5**
motorbike	48.2	38.4	49.9	46.1	**51.5**
person	**43.2**	36.6	41.4	36.3	40.3
pottedplant	12.0	15.1	**16.0**	12.7	14.3
sheep	21.1	19.7	22.4	22.3	**23.9**
sofa	36.1	25.1	37.2	35.1	**41.6**
train	46.0	36.8	48.5	43.9	**49.2**
tvmonitor	43.5	39.3	42.4	41.8	**46.0**
mean	33.7	29.6	34.7	32.9	**36.6**

Table 5.1: Performance comparison using average precision (AP) for the PASCAL VOC2007 dataset. For abbreviations see Section 5.3.2

Furthermore one can observe from recall-precision curves given in Figure 5.9 that the suggested method is increasing precision for most classes in areas with intermediate recall while the precision for low and high recall areas is similar to that of the DPM.

Since the number of random compositions (K=50) is rather small one could suspect that the variance of the detection performance is high. However, measuring the variance of the mean average precision of five different random composition samplings showed a favorable variance of about 0.1%.

PASCAL VOC 2010

Additionally, we are providing results on the PASCAL VOC2010 dataset were we outperform other approaches on 12 out of 20 classes (see Table 5.2). Our approach performs particularly well for classes that can be considered as very difficult due to the huge intra-class variations as birds, boats and potted plants were the improvement is up to 4.1% in terms of average precision. The comparison with the Boosted Collection of Parts (BCP) is particularly interesting, since due to their usage of exemplar parts it is the most similar approach to our compositional part-model. We are showing superior performance

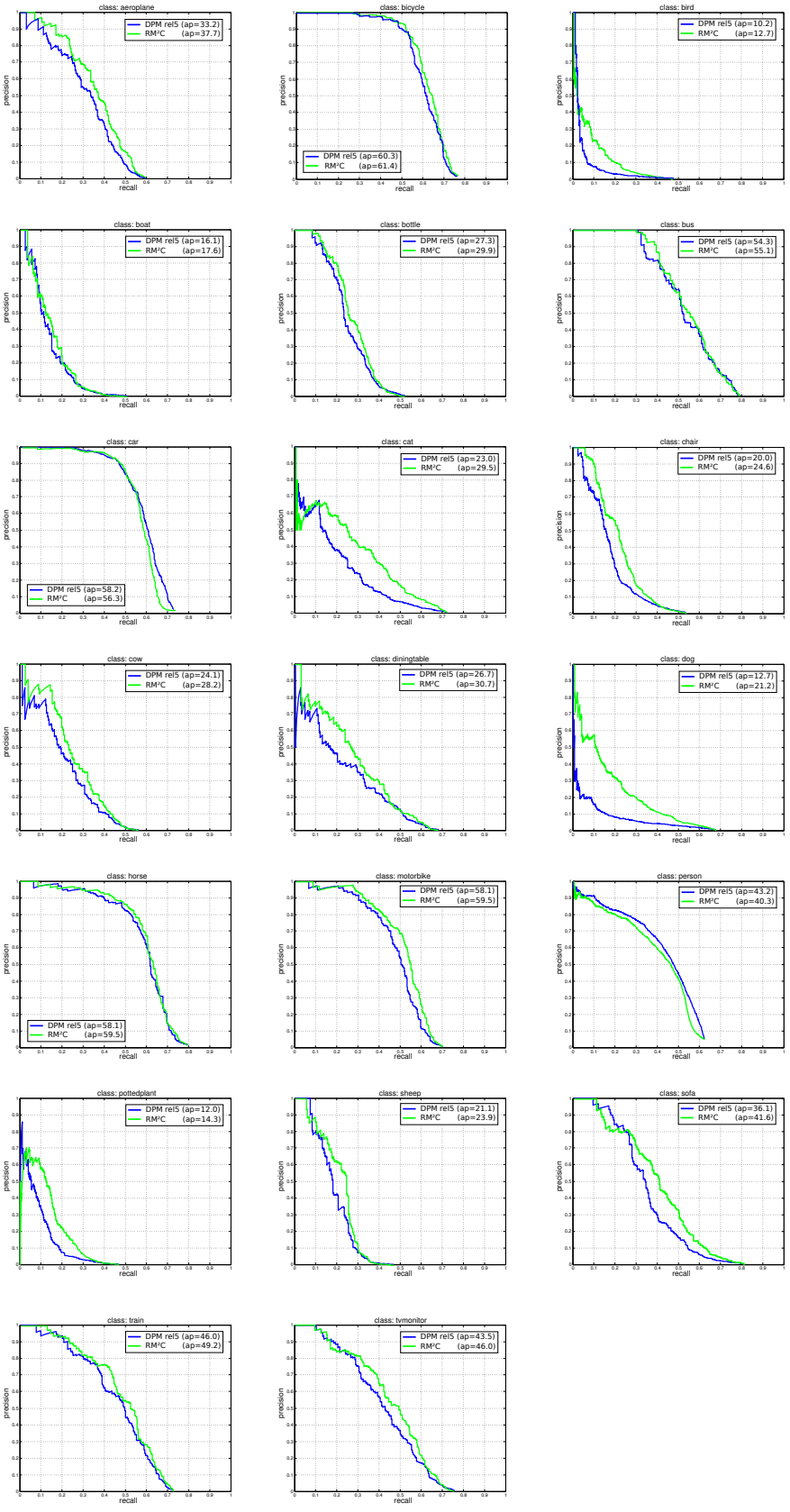


Figure 5.9: Recall-precision curves for the deformable part model (DPM) and the final RM^{2}C detector for PASCAL VOC 2007 dataset.

	DPM rel5 [56]	Poselets [23]	BCP [46]	AOT [154]	RM^{2}C
aeroplane	45.6	33.2	44.3	44.6	**49.8**
bicycle	49.0	**51.0**	35.2	48.5	**50.6**
bird	11.0	8.5	9.7	10.8	**15.1**
boat	11.6	8.2	10.1	12.9	**15.5**
bottle	27.2	**34.8**	26.3	22.9	28.5
bus	50.5	39.0	44.6	47.5	**51.1**
car	43.1	**48.8**	32.0	41.6	42.2
cat	23.6	22.2	**35.3**	21.6	**30.5**
chair	17.2	-	4.4	**17.3**	**17.3**
cow	23.2	20.6	17.5	23.6	**28.3**
table	10.7	-	**15.0**	11.5	**12.4**
dog	20.5	18.5	**27.6**	22.9	**26.0**
horse	42.5	**48.2**	36.2	40.9	**45.6**
motorbike	44.5	44.1	42.1	45.3	**51.8**
person	41.3	**48.5**	30.0	37.9	41.4
pottedplant	8.7	9.1	5.0	9.6	**12.6**
sheep	29.0	28.0	13.7	**30.4**	**30.4**
sofa	18.7	13.0	18.8	25.3	**26.1**
train	40.0	22.5	34.4	39.0	**44.0**
tvmonitor	34.5	33.0	28.6	31.2	**37.6**
mean	29.6	-	34.7	29.4	**32.8**

Table 5.2: Performance comparison using average precision (AP) for the PASCAL VOC2010 dataset. Note that our approach outperforms Poselets comparing the mean of the 18 classes, where the detection results are provided, by 5.3%.

on 17 out of 20 classes, improving the average precision by 7.8%. While poselets are giving the best performance on 5 out of 20 classes they also perform more than 10% worse than our detection system on 5 other classes. Note, that we are outperforming the poselets even though this approach uses additional ground truth annotation in the form of keypoints for training while ours only depends on bounding box annotations at object level. Comparing the mean over the 18 classes on which results for the poselets are available we outperform them by 5.3%. All in all we are gaining 3.2% in terms of mean average precision over the DPM which is the best performing approach we are comparing to.

5.3.3 Object Parsing Results

Besides the quantitative object detection results we are also providing qualitative results for object detection and parsing of the suggested randomized max-margin compositions in Figure 5.10 and Figure 5.11. In the following, first details of the reconstruction process will be given and then reconstruction results will be discussed.

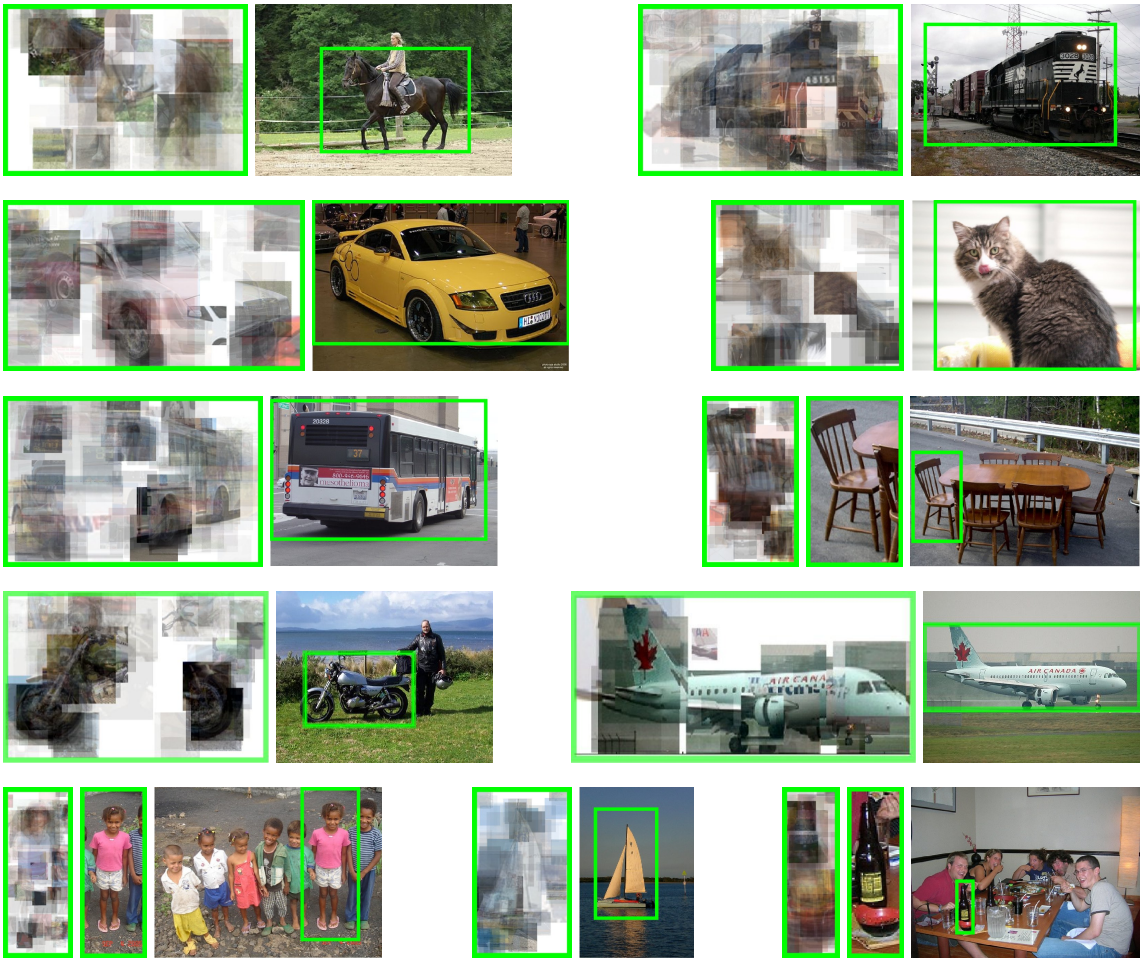


Figure 5.10: Object detection and parsing with randomized max-margin compositions. For a true positive detection in a test image we show (from left to right) the parsing result provided by our algorithm, for small detections the corresponding cropped out image region and the detection (green box) in the full-sized test image.

Reconstruction Process

Figure 5.10 and Figure 5.11 show the result of applying the recognition process. While results in Figure 5.10 show the reconstruction of different classes Figure 5.11 shows the ability of our approach to handle intra-class variations and viewpoint changes.

First, we extract HOG descriptors x_j and run part classifiers $h_i(x_j)$ from Equation 5.6. Then, we pool part responses into $\pi_i(\nu)$ using Equation 5.5 before running the composition classifiers $f_k(\cdot)$ (Equation 5.1). Finally, we evaluate $g(F(\cdot))$ (Equation 5.2) to combine all compositions using the non-linear classifier.

The parsing results at the left then show the responses of compositions and their constituent parts. The non-linear classifier $g(F(\cdot))$ weights the compositional classifiers $f_k(\cdot)$ which in turn activate their constituent part classifiers $\pi_i(\nu)$ with weights w_k^T. The weights indicate the importance for separating positive and negative training samples. For

each part i and site ν, we go to the location of the part x_j in the test image that wins the pooling of Equation 5.5. At this location in the test bounding box $\mathcal{I}$ we then place the single positive patch x_p from the training data that defines the i-th part classifier. The transparency of this training patch is the importance that the compositional model assigns to it, i.e., it is proportional to $g(F(\mathcal{I}))\, f_k(\gamma_k(\mathcal{I}))\, w_{k,i}\, \pi_i(\nu)$.

Observations

In Figure 5.10 parsing tries to explain test data using training samples. Due to large intra-class variation in PASCAL VOC, test samples are quite diverse from training data and we observe generalization artifacts: e.g. the radiator grill of the car is changing the brand from "Audi" to "Mercedes"; the t-shirt of the little girl (last row, left) is changing its color from pink to blue; the train in the first row slightly changes in style from a modern locomotive to a steam train.

Figure 5.11 shows the importance the model assigns to different object regions. For the two motorbikes (second row) the middle part of the motorbikes, which is often covered by a sitting person is assigned lower weight. This region is less reliable than the rest of the bike, as it is covered by bike riders in some samples, thus leading to larger variability. Similar behavior can be seen for the bicycles and horses. Moreover, we see that if a query region does not fit to what is expected, there is not arbitrary hallucination, but rather the region is down weighted. Whereas the right tire of the first bicycle is completely reconstructed, the middle of the other tire is down weighted. This is due to the large discrepancy to what the model has learned from training data for this region of a bike.

5.4 Scene Classification Results

5.4.1 MITIndoor Scene Recognition Dataset

The MITIndoor [137] dataset is consisting of 67 indoor scene categories and was the first large indoor scene classification benchmark dataset. It was collected because most outdoor scenes classification approaches perform poorly on indoor scenes. The difference between outdoor and indoor scene classification is that outdoor scenes can be well characterized by global spatial models while such models are not appropriate for all indoor scenes. For the classification of most indoor scenes one needs to capture global and local information. The 15620 images contained in the dataset were collected from Flickr, Goolge, Altavista and the LabelMe dataset.

We are using the protocol given in [137]. For each scene class a one versus all classifier is trained and results are combined into a single prediction by taking the maximum classification score. Each individual scene classifier is trained on a fixed set of 80 positive training samples, while positives from the other classes are used as negatives. For testing, each scene class provides 20 test images resulting in a testset of 1340 images.

We provide results, as in [137], in terms of classification accuracy which is defined by the number of correct classifications divided by the number of samples obtained by averaging

Figure 5.11: Object detection and parsing with randomized max-margin compositions as in Figure 5.11. Each row shows two example detections for the same category thus illustrating how the model deals with large intra-class variabilities and viewpoint changes.

the diagonal of the confusion matrix:

$$\text{acc} = \frac{1}{67} \sum_{i=1}^{67} \frac{\#TP_i}{\#P_i}. \tag{5.7}$$

Furthermore, performance is measured in terms of mean average precision which is used as an additional measurement in [89].

5.4.2 Comparison with State-of-the-Art

We compare our performance to 7 different classification approaches (see Table 5.3). The focus of our evaluation is the comparison to other methods that are using semantical part classifiers based on HOG features for scene classification. Therefore, the most important comparisons are in the lower half of Table 5.3, since these approaches are

Method	Acc. (%)	Mean AP
Object Bank [106]	37.60	-
RBoW [132]	37.93	-
DPM+GIST-color+SP [130]	43.10	-
Patches+GIST+SP+DPM [130]	49.40	-
IFV+BoP [89]	63.10	63.18
Mid-Level Patches [153]	38.10	-
BoP [89]	46.10	43.55
RM^2C	51.34	46.70

Table 5.3: Average classification performance on the MITIndoor Dataset. Upper half of the table shows diverse approaches for scene classification while the lower half focuses on approaches using semantic parts and are therefore most similar to our approach.

methodologically most similar to the one we are suggesting. Our results show that we outperform Mid-Level Patches [153] by 13% and the Bag of Parts (BoP) by 5% in terms of classification accuracy. The improved fisher vectors (IFV) can be combined with all part based approaches to boost performance as it was done by IFV+BoP [89]. Since we are outperforming the individual performance of BoP, it should be expected that the combination with fisher vectors would outperform their combined approach. However, the aim of this experiment was to compare our method with other related part based approaches.

5.5 Discussion

We have proposed a compositional approach that can integrate large numbers of weak parts in a strong discriminative model. Contrary to the main theme of the field, we randomly sample instance specific parts and randomly aggregated them in compositions that are trained using a max-margin procedure. The approach has shown favorable performance on standard benchmark datasets for object detection and scene classification and the potential of its constituents has been evaluated individually.

CHAPTER 6

CONCLUSIONS

This thesis dealt with the task of visual object and scene recognition using mid-level representations. Different mid-level representations were suggested, that were build on different low-level features, to improve state-of-the-art object detection and scene classification frameworks. As the representations of object and scenes is closely linked to the learning algorithms this thesis developed novel mid-level representations and learning strategies that are suitable for the appendent mid-level representation.

This work showed that high dimensional object descriptors arising from higher order statistics, such as self-similarity, cannot be handled sufficiently well by support vector machines without applying additional feature selection for noise reduction. Specifically, this was shown for the novel curvature self-similarity descriptor suggested in this thesis. Furthermore, it was found that the novel curvature self-similarity descriptor provides complementary information to the widely used orientation histograms and to simple curvature histograms. As support vector machines are not readily able to deal with such high dimensional descriptors on a limited amount of training data, a new embedded feature selection method for support vector machines was devised. Experimental results verified the premise that the performance of high-dimensional object descriptors is improving, when appropriate learning algorithms are applied that are able to eliminate superfluous dimensions. Therefore this approach provides an improvement to the widely used framework utilizing histograms of oriented gradients and support vector machines for object detection.

Furthermore, this thesis investigated to overcome the susceptibility of chamfer matching to background clutter which is a serious drawback of this popular method. While other approaches typically focus on improving the foreground template this work focused on explicitly modeling the background to avoid accidental matches in clutter. The flexible co-placement of generic background contours was learned and integrated with current extensions of chamfer matching that aim to improve the matching of the foreground model. To model accidentalness, background contours were placed on the foreground model contour and characteristic co-occurrences between these contours were learned

using non-linear radial basis function kernel. Experimental results on standard shape based datasets for object recognition showed significant performance improvement and qualitative results showed a large reduction of false positives in dense clutter.

Another interesting finding of this work is that it is beneficial to use a large number of specific part classifiers for object detection and scene classification. This is contrary to the common approach of utilizing a small set of generic parts learned together in a discriminative framework. Individual part classifiers were trained using an exemplar support vector machine framework, i.e. the part classifiers were trained on a single positive and are therefore very specific but also weak. To improve generalization ability of these local and specific parts they were grouped into stronger compositions. Compositions were shown earlier to reduce computational complexity and render learning of structured models feasible. However, in this thesis it was found, that the common approach of combining parts into meticulously arranged, semantically meaningful compositions was outperformed by grouping parts into randomized compositions. A non-linear discriminative classifier is combining the compositions in the last layer of the hierarchy. Additionally, a novel framework for part evaluation was suggested for thoroughly evaluating the intermediate components of the hierarchical part based model. Evaluation showed that randomly sampling individual part classifiers and learning them in an exemplar fashion is leading to satisfactory number of parts that have good detection performance, compared to more general parts learned by the deformable part model. This result was not only confirmed quantitatively but also qualitatively, as the suggested framework provides a detailed parsing of the detected test objects.

In conclusion, this thesis devised different kinds of mid-level representations and utilized them successfully for object and scene recognition. Mid-level representations improved recognition performance in several object detection frameworks utilizing different low-level features and outperform current state-of-the-art approaches on challenging benchmark datasets.

LIST OF PUBLICATIONS

The following scientific articles have been published based on this dissertation

- Eigenstetter, A., Takami, M., and Ommer, B. Randomized Max-Margin Compositions for Visual Recognition. *Conference on Computer Vision and Pattern Recognition* (2014)
- Eigenstetter, A., and Ommer, B. Visual Recognition using Embedded Feature Selection for Curvature Self-Similarity. *Conference on Neural Information Processing Systems* (2012)
- Eigenstetter, A., Yarlagadda, P., and Ommer, B. Max-Margin Regularization for Reducing Accidentalness in Chamfer Matching. *Asian Conference on computer Vision* (2012)

BIBLIOGRAPHY

[1] AGIN, G. J., AND BINFORD, T. O. Computer description of curved objects. *IEEE Transactions on Computers C-25*, 4 (1967), 439–449.

[2] AKTAS, U. R., OZAY, M., LEONARDIS, A., AND WYATT, J. L. A graph theoretic approach for object shape representation in compositional hierarchies using a hybrid generative-descriptive model. *Proceedings of the European Conference on Computer Vision* (2014).

[3] ALEXE, B., DESELAERS, T., AND FERRARI, V. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 11 (2012), 2189–2202.

[4] ANDRILUKA, M., ROTH, S., AND SCHIELE, B. People-tracking-by-detection and people-detection-by-tracking. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2008).

[5] ATTNEAVE, F. Some informational aspects of visual perception. *Psychological review 61*, 3 (1954).

[6] AZIZPOUR, H., AND LAPTEV, I. Object detection using strongly supervised deformable part models. *Proceedings of the European Conference on Computer Vision* (2012).

[7] BARROW, H. G., TENENBAUM, J. M., BOLLES, R. C., AND WOLF, H. C. Parametric correspondence and chamfer matching: Two new techiques for image matching. *International Joint Conference Artifical Intelligence* (1977), 659–663.

[8] BELONGIE, S., AND MALIK, J. Matching with shape contexts. *IEEE Workshop on Contentbased Access of Image and Video Libraries* (2000).

[9] BELONGIE, S., MALIK, J., AND PUZICHA, J. Shape context: A new descriptor for shape matching and object recongition. *Proceedings of the Conference on Advances in Neural Information Processing Systems* (2000).

[10] BELONGIE, S., MALIK, J., AND PUZICHA, J. Matching shapes. *Proceedings of the IEEE International Conference on Computer Vision* (2001).

[11] BELONGIE, S., MALIK, J., AND PUZICHA, J. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 24 (2002), 509–521.

[12] BENNETT, K. P., AND MANGASARIAN, O. L. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods Software 1* (1992), 23–34.

[13] BERG, A. C., BERG, T. L., AND MALIK, J. Shape matching and object recognition using low distortion correspondence. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2005).

[14] BERG, A. C., AND MALIK, J. Geometric blur for template matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2001), pp. 607–614.

[15] BIEDERMAN, I. Recognition-by-components: A theory of human image understanding. *Psychological Review 94*, 2 (1987), 115–147.

[16] BISHOP, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

[17] BODEN, M. A. *Mind As Machine: A History of Cognitive Science.* Oxford University Press, 2006.

[18] BOIMAN, O., SHECHTMAN, E., AND IRANI, M. In defense of nearest-neighbor based image classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2008).

[19] BORGEFORS, G. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 10*, 6 (1988), 849–865.

[20] BORGEFORS, G. Distance transformations in digital images. *Computer Vision, Graphics and Image Processing 34*, 3 (1996), 227–248.

[21] BOSER, B., GUYON, I. M., AND VAPNIK, V. N. A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (1992), 144–152.

[22] BOURDEV, L., MAJI, S., AND MALIK, J. Detection, attribute classification and action recognition of people using poselets (in submission). In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[23] BOURDEV, L., AND MALIK, J. Poselets: Body part detectors trained using 3D human pose annotations. *Proceedings of the IEEE International Conference on Computer Vision* (2009).

[24] BRADLEY, P. S., AND MAGASARIAN, O. L. Feature selection via concave minimization and support vector machines. *International Conference of Machine Learning* (1998).

[25] BRANSON, S., WAH, C., SCHROFF, F., BABENKO, B., WELINDER, P., PERONA, P., AND BELONGIE, S. Visual recognition with humans in the loop. *Proceedings of the European Conference on Computer Vision* (2010).

[26] BREIMAN, L. Random forests. *Machine Learning* (2001).

[27] CANNY, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*, 6 (1986), 679–698.

[28] CHANG, C.-C., AND LIN, C.-J. LIBSVM: A library for support vector machines. *Association for Computing Machinery Transactions on Intelligent Systems and Technology 2* (2011), 27:1–27:27.

[29] CHANG, P., AND KRUMM, J. Object recognition with color cooccurrence histograms. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (1999).

[30] CHATFIELD, K., PHILBIN, J., AND ZISSERMAN., A. Efficient retrieval of deformable shape classes using local self-similarities. *International Conference on Computer Vision, Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment* (2009).

[31] CHEN, G., DING, Y., XIAO, J., AND HAN, T. Detection evolution with mulit-order conextual co-occurrence. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013).

[32] CHEN, X., MOTTAGHI, R., LIU, X., FIDLER, S., URTASUN, R., AND YUILLE, A. Detect what you can: detecting and representing objects using holistic models and body parts. *CVPR:L* (2014).

[33] COLLINS, M., SCHAPIRE, R. E., AND SINGER, Y. Logistic regression, adaboost and bregman distances. *Proceedings of the 13th Annual Conference on Computational Learning Theory.* (2000), 158–169.

[34] CORTES, C., AND VAPNIK, V. Support-vector networks. *Machine Learning 20* (1995), 273–297.

[35] CRANDALL, D. J., FELZENSWALB, P. F., AND HUTTENLOCHER, D. P. Spatial prios for part-based recognition using statistical models. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2005).

[36] CSURKA, G., DANCE, C., FAN, L., WILLAMOWSKI, J., AND BRAY, C. Visual categorization with bags of keypoints. *European Conference on Computer Vision International Workshop on Statistical Learning in Computer Vision* (2004).

[37] DALAL, N., AND TRIGGS, B. Histograms of oriented gradients for human detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2005).

[38] DANIELSSON, O., CARLSSON, S., AND SULLIVAN, J. Automatic learning and extraction of multi-local features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009).

[39] DANIELSSON, P. E. Euclidean distance mapping. *Computer Graphics and Image Processing 14*, 3 (1980), 227–248.

[40] DESELAERS, T., AND FERRARI, V. Global and efficient self-similarity for object classification and detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2010).

[41] DO, T.-M.-T., AND ARTIÉRES, T. Large margin training for hidden markov models with partially observed states. *International Conference on Machine Learning* (2009).

[42] DOLLAR, P., TU, Z., AND BELONGIE, S. Supervised learning of edges and object boundaries. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2006).

[43] DUAN, K., PARIKH, D., CRANDALL, D., AND GRAUMAN, K. Discovering localized attributes for fine-grained recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2012).

[44] DUDA, R. O., HART, P. E., AND STORK, D. G. *Pattern Classification*. John Wiley and Sons, 2001.

[45] E. A. VAN DE SANDE, K., GEVERS, T., AND SNOEK, C. G. M. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 9 (2010), 1582–1596.

[46] ENDRES, I., SHIH, K. J., JIAA, J., AND HOIEM, D. Learning collections of part models for object recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013).

[47] EPSHTEIN, B., AND ULLMAN, S. Feature hierarchies for object classification. *Proceedings of the IEEE International Conference on Computer Vision* (2005).

[48] EVERINGHAM, M., GOOL, L. V., WILLIAMS, C. K. I., WINN, J., AND ZISSERMAN, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[49] EVERINGHAM, M., VAN GOOL, L., WILLIAMS, C. K. I., WINN, J., AND ZISSERMAN, A. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision 88*, 2 (2010), 303–338.

[50] FAN, R.-E., CHANG, K.-W., HSIEH, C.-J., WANG, X.-R., AND LIN, C.-J. Liblinear: A library for large linear classification. *Journal of Machine Learning Research 9* (2008), 1871–1874.

[51] FARABET, C., COUPRIE, C., NAJMAN, L., AND LECUN, Y. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 8 (2013).

[52] FARHADI, A., ENDRES, I., AND HOIEM, D. Attribute-centric recognition for cross-category generalization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2010).

[53] FARHADI, A., ENDRES, I., HOIEM, D., AND FORSYTH, D. Describing objects by their attributes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009), 1778–1785.

[54] FEI-FEI, L., FERGUS, R., AND PERONA, P. A bayesian approach to unsupervised one-shot learning of object categories. *Proceedings of the IEEE International Conference on Computer Vision* (2003).

[55] FELZENSWALB, P. F., AND HUTTENLOCHER, D. P. Distance transforms of sampled functions. Tech. rep., Cornell Computing and Information Science, 2004.

[56] FELZENSZWALB, P., GIRSHICK, R., MCALLESTER, D., AND RAMANAN, D. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 9 (2010), 1627–1645.

[57] FELZENSZWALB, P. F., GIRSHICK, R. B., AND MCALLESTER, D. Discriminatively trained deformable part models, release 4. http://www.cs.brown.edu/ pff/latent-release4/.

[58] FELZENSZWALB, P. F., AND HUTTENLOCHER, D. P. Pictorial structures for object recognition. *International Journal of Computer Vision* (2005).

[59] FERGUS, R., PERONA, P., AND ZISSERMAN, A. Object class recognition by unsupervised scale-invariant learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2003).

[60] FERGUS, R., PERONA, P., AND ZISSERMAN, A. A sparse object category model for effcient learning and exhaustive recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2005).

[61] FIDLER, S., BOBEN, M., AND LEONARDIS, A. A coarse-to-fine taxonomy of constellations for fast multi-class object detection. *Proceedings of the European Conference on Computer Vision* (2010).

[62] FIDLER, S., AND LEONARDIS, A. Towards scalable representations of visual categories: Learning a hierarchy of parts. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2007).

[63] FIDLER, S., MOTTAGHI, R., YUILLE, A., AND URTASUN, R. Bottom-up segmentation for top-down detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013).

[64] FISCHLER, M. A., AND ELSCHLAGER, R. A. The representation and matching of pictorial structures. *IEEE Transactions on Computers C-22*, 1 (1973), 67–92.

[65] FREEMAN, W. T., AND ROTH, M. Orientation histograms for hand gesture recognition. *International Workshop on Automatic Face and Gesture- Recognition* (1995).

[66] FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36 (1980), 193–202.

[67] FUREY, T., CRISTIANINI, N., DUFFY, N., BEDNARSKI, D. W., SCHUMMER, M., AND HAUSSLER, D. Support vector machine classification and validation of cancer tissue samples using microarray experssion data. *Bioinformatics 16* (2000), 906–914.

[68] GAVRILA, D. A bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 8 (2007).

[69] GAVRILA, D. M. Multi-feature hierarchical template matching using distance transforms. *Proceedings of the International Conference on Pattern Recognition* (1998), 439–444.

[70] GAVRILA, D. M., AND MUNDER, S. Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision 73*, 1 (2007), 41–49.

[71] GOLDSTEIN, E. B. *Sensation and Perception*. Wadsworth, 2014.

[72] GORDON, I. *Theories of visual perception*. Psychology Press, 2004.

[73] GOSSET, W. S. The probable error of a mean. *Biometrika 6*, 1 (1908), 1–25. Originally published under the pseudonym "Student".

[74] GRANDVALET, Y., AND CANU, S. Adaptive scaling for feature selection in SVMs. *Proceedings of the Conference on Advances in Neural Information Processing Systems* (2003).

[75] GUPTA, A., EFROS, A. A., AND HEBERT, M. Blocks world revisited: Image understanding using qualitative geometry and mechanics. *Proceedings of the European Conference on Computer Vision* (2010).

[76] GUYON, I., AND ELISSEEFF, A. An introduction to variable and feature selection. *Journal of Machine Learning Research 3* (2003), 1157–1182.

[77] GUYON, I., WESTON, J., BARNHILL, S., AND VAPNIK, V. Gene selection for cance classification using support vector machines. *Journal of Machine Learning Research 3* (2003), 1439–1461.

[78] HAN, J. H., AND POSTON, T. Chord-to-point distance acccumulation and planar curvature: a new approach to discrete curvature. *Pattern Recognition Letters 22*, 10 (2001), 1133 – 1144.

[79] HARIHARAN, B., MALIK, J., AND RAMANAN, D. Discriminative decorrelation for clustering and classification. *Proceedings of the European Conference on Computer Vision* (2012).

[80] HEDAU, V., HOIEM, D., AND FORSYTH, D. Recovering the spatial layout of cluttered room. *Proceedings of the IEEE International Conference on Computer Vision* (2009).

[81] HEDAU, V., HOIEM, D., AND FORSYTH, D. Thinking inside the box: Using appearance models and context based on room geometry. *Proceedings of the European Conference on Computer Vision* (2010).

[82] HEITZ, G., AND KOLLER, D. Learning spatial context: Using stuff to find things. *Proceedings of the European Conference on Computer Vision* (2008).

[83] HOIEM, D., EFROS, A., AND HEBERT, M. Putting objects in perspective. *International Journal of Computer Vision 80*, 1 (2008), 3–15.

[84] HÖRSTER, E., AND LIENHART, R. Deep networks for image retrieval on large-scale databases. *Association for Computing Machinery, Multimedia* (2008).

[85] HUTTENLOCHER, D., KLANDERMAN, G., AND RUCKLIDGE, W. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*, 9 (1993), 850–863.

[86] ITO, S., AND KUBOTA, S. Object classification using heterogeneous co-occurrence features. *Proceedings of the European Conference on Computer Vision* (2010).

[87] JAYARAMAN, D., SHA, F., AND GRAUMAN, K. Decorrelating semantic visual attributes by resisting the urge to share. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014).

[88] JIN, J., AND GEMAN, S. Context and hierarchy in a probabilistic image model. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2006).

[89] JUNEJA, M., VEDALDI, A., JAWAHAR, C. V., AND ZISERMAN, A. Blocks that shout: Distinctive parts for scene classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013).

[90] JUNEJO, I. N., DEXTER, E., LAPTEC, I., AND PERÉZ, P. Cross-view action recognition from temporal self-similarities. *Proceedings of the European Conference on Computer Vision* (2008).

[91] KADIR, T., AND BRADY, M. Saliency, scale and image description. *International Journal of Computer Vision 45* (2001).

[92] KANIZSA, G. Margini quasi-percettivi in campi con stimolazione omogenea. *Rivista di Psicologia 49*, 1 (1955), 7–30.

[93] KARMITSA, N., FILHO, M. T., AND HERSKOVITS, J. Globally convergent cutting plane method for nonconvex nonsmooth minimization. *Journal of Optimization Theory and Applications 148*, 3 (2011), 528 – 549.

[94] KIRA, K., AND RENDELL, L. The feature selection problem: Traditional methods and a new algorithm. *AAAI'92 Proceedings of the tenth national conference on Artificial intelligence* (1992), 129–134.

[95] KOHAVI, R., AND JOHN, G. Wrappers for feature selection. *Artificial Intelligence 97* (1997), 273–324.

[96] KOKKINOS, I., AND YUILLE, A. Inference and learning with hierarchical shape models. *International Journal of Computer Vision 93* (2011), 201–255.

[97] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Proceedings of the Conference on Advances in Neural Information Processing Systems* (2012).

[98] LAL, T. N., CHAPELLE, O., WESTON, J., AND ELISSEEFF, A. *Studies in Fuzziness and Soft Computing*. I. Guyon and S. Gunn and N. Nikravesh and L. A. Zadeh, 2006.

[99] LAMPERT, C., NICKISCH, H., AND HARMELING, S. Learning to detect unseen object classes by between-class attribute transfer. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009).

[100] LAZEBNIK, S., SCHMID, C., AND PONCE, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2006).

[101] LECUN, Y., BOSER, B., DENKER, J., HENDERSON, D., HOWARD, R., HUBBARD, W., AND JACKEL, L. Backpropagation applied to handwritten zip code recognition. *Neural Computation 1*, 4 (1989), 541–551.

[102] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of IEEE 11*, 86 (1998), 2278–2324.

[103] LEIBE, B., LEONARDIS, A., AND SCHIELE, B. Combined object categorization and segmentation with and implicit shape model. *European Conference on Computer Vision, Workshop on Statistical Learning in Computer Vision* (2004).

[104] LEIBE, B., SEEMANN, E., AND SCHIELE, B. Pedestrian detection in crowded scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2005).

[105] LEVINSHTEIN, A., SMINCHISESCU, C., AND DICKINSON, S. J. Learning hierarchical shape models from examples. *Energy Minimization Methods in Computer Vision and Pattern Recognition 3757* (2005), 251–267.

[106] LI, L., SU, H., XING, E. P., AND FEI-FEI, L. Object bank: A high-level image representation for scene classification & semantic feature sparsification. *Proceedings of the Conference on Advances in Neural Information Processing Systems* (2010).

[107] LIN, Z., DAVIS, L. S., DOERMANN, D., AND DEMENTHON, D. Hierarchical part template matching for human detection and segmentation. *Proceedings of the IEEE International Conference on Computer Vision* (2007).

[108] LIU, M., TUZEL, O., A.VEERARAGHAVAN, AND CHELLAPPA, R. Fast directional chamfer matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2010).

[109] LOWE, D. G. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, 1985.

[110] LOWE, D. G. Object recognition from lcoal scale-invariant features. *Proceedings of the IEEE International Conference on Computer Vision* (1999).

[111] LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.

[112] LUNTZ, A., AND BRAILOVSKY, V. On the estimation of characters obtained in statistical procedure of recognition. *Technicheskaya Kibernetica* (1996).

[113] MA, T., YANG, X., AND L.LATECKI. Boosting chamfer matching by learning chamfer distance normalization. In *Proceedings of the European Conference on Computer Vision* (2010).

[114] MAIRAL, J., LEORDEANU, M., BACH, F., HEBERT, M., AND PONCE, J. Discriminative sparse image models for class-specific edge detection and image

interpretation. *Proceedings of the European Conference on Computer Vision* (2008).

[115] MAJI, S., BERG, A. C., AND MALIK, J. Classification using intersection kernel support vector machines is efficient. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2008).

[116] MAJI, S., AND MALIK, J. Object detection using a max-margin hough transform. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009).

[117] MALISIEWICZ, T., GUPTA, A., AND EFROS, A. Ensemble of exemplar-svms for object detection and beyond. *Proceedings of the IEEE International Conference on Computer Vision* (2011).

[118] MARR, D. *Vision : A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., 1982.

[119] MARTIN, D., FOWLKES, C., AND MALIK, J. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 5 (2004), 530 – 549.

[120] MCCARTHY, J., MINSKY, M. L., ROCHESTER, N., AND SHANNON, C. E. Proposal for the dartmouth summer research project on artificial intelligence.

[121] MITA, T., KANEKO, T., STENGER, B., AND HORI, O. Discriminative feature co-occurrence selection for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 7 (2008), 1257–1269.

[122] MONROY, A., EIGENSTETTER, A., AND OMMER, B. Beyond straight lines - object detection using curvature. *IEEE International Conference on Image Processing* (2011).

[123] MONROY, A., AND OMMER, B. Beyond bounding-boxes: Learning object shape by model-driven grouping. *Proceedings of the European Conference on Computer Vision* (2012).

[124] MURPHY, K. P. *Machine Learning: a probabilistic perspective*. MIT press, 2012.

[125] NEVATIA, R., AND BINFORD, T. O. Description and recognition of curved objects. *Artificial Intelligence 8* (1977), 77–98.

[126] ODONE, F., BARLA, A., AND VERRI, A. Building kernels from binary strings for image matching. *IEEE Transactions on Image Processing 14*, 2 (2005), 169–180.

[127] OMMER, B., AND BUHMANN, J. M. Learning compositional categorization models. *Proceedings of the European Conference on Computer Vision* (2006).

[128] OMMER, B., AND BUHMANN, J. M. Learning the compositional nature of visual object categories for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 3 (2010), 501–516.

[129] PALMER, S. E. *Theoretical approaches to vision*. MIT Press, 1999, ch. 2, pp. 45–92.

[130] PANDEY, M., AND LAZEBNIK, S. Scene recognition and weakly supervised object localization with deformable part-based models. *Proceedings of the IEEE International Conference on Computer Vision* (2011).

[131] PAPAGEORGIOU, C., AND POGGIO, T. A trainable system for object detection. *International Journal of Computer Vision 38*, 1 (2000), 15–33.

[132] PARIZI, S. N., OBERLIN, J., AND FELZENSWALB, P. F. Reconfigurable models for scene recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2012).

[133] PEARSON, K. Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London 58* (1895), 240–242.

[134] PENTLAND, A. P. Perceptual organization and the representation of natural form. *Artificial Intelligence 28*, 3 (1986), 293–331.

[135] PERKINS, S., LACKER, K., AND THEILER, J. Grafting: Fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research 3* (2003), 1333–1356.

[136] PONTIL, M., ROGAI, S., AND VERRI, A. Recognizing 3-d objects with linear support vector machines. *Proceedings of the European Conference on Computer Vision* (1998).

[137] QUATTONI, A., AND TORRALBA, A. Recognizing indoor scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009).

[138] RAKOTOMAMONJY, A. Variable selection using svm-based criteria. *Journal of Machine Learning Research 3* (2003), 1357–1370.

[139] RANZATO, M., MNIH, V., SUSSKIND, J. M., AND HINTON, G. E. Modeling natural images using gated MRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 9 (2013), 2206–2222.

[140] REN, H., HENG, C.-K., ZHENG, W., LIANG, L., AND CHEN, X. Fast object detection using boosted co-occurrence histograms of oriented gradients. *International Conference on Image Processing.* (2010).

[141] REUNANEN, R. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research 3* (2003), 1371–1382.

[142] ROBERTS, L. Machine perception of three-dimensional solides. *PhD thesis, MIT* (1963).

[143] ROSENFELD, A., AND PFALTZ, J. L. Sequential operations in digital picture processing. *Journal of the Association for Computing Machinery 13*, 4 (1966), 471–494.

[144] ROTH, V. The generalized lasso. *IEEE Transactions on neural networks* (2003).

[145] RUSSEL, S., AND NORVIG, P. *Artificial Intelligence : A Modern Approach.* Prentice Hall, 2010.

[146] SCHNEIDERMAN, H., AND KANADE, T. A statistical method for 3d object detection applied to faces and cars. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2000).

[147] SCHNITZSPAN, P., FRITZ, M., ROTH, S., AND SCHIELE, B. Discriminative structure learning of hierarchical representations for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009).

[148] SCHOELKOPF, B., AND SMOLA, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[149] SCHWARTZ, W., KEMBHAVI, A., HARWOOD, D., AND DAVIS, L. Human detection using partial least squares analysis. *Proceedings of the IEEE International Conference on Computer Vision* (2009).

[150] SERRE, T., WOLF, L., BILESCHI, S., AND RIESENHUBER, M. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 3*, 29 (2007), 411–426.

[151] SHECHTMAN, E., AND IRANI, M. Matching local self-similarities across images and videos. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2007).

[152] SHOTTON, J., BLAKE, A., AND CIPOLLA, R. Multi-scale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 7 (2008), 1270–1281.

[153] SINGH, S., GUPTA, A., AND EFROS, A. A. Unsupervised discovery of mid-level discriminative patches. *Proceedings of the European Conference on Computer Vision* (2012).

[154] SONG, X., T. WU, JIA, Y., AND ZHU, S. Disriminatively trained and-or tree models for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013).

[155] SONG, Z., CHEN, Q., HUANG, Z., HUA, Y., AND YAN, S. Contextualizing object detection and classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2011).

[156] STEINWART, I. Sparseness of support vector machines – some asymptotically sharp bounds. In *Proceedings of the Conference on Advances in Neural Information Processing Systems* (2004).

[157] SZELISKI, R. *Computer Vision: Algorithms and Applications*. Springer, 2010.

[158] THAYANANTHAN, A., STENGER, B., TORR, P., AND CIPOLLA, R. Shape context and chamfer matching in cluttered scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2003).

[159] TIBSHIRANI, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. 58*, 1 (1996), 267–288.

[160] TORRALBA, A. Contextual priming for object detection. *International Journal of Computer Vision 53*, 2 (2003), 169–191.

[161] TSOCHANTARIDIS, I., HOFMANN, T., JOACHIMS, T., AND ALTUN., Y. Support vector learning for interdependent and structured output spaces. *International Conference on Machine Learning* (2004).

[162] TU, Z., CHEN, X., YUILLE, A., AND ZHU, S. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision 63* (2005), 113–140.

[163] TURING, A. Computing machinery and intelligence. *Mind 59* (1950), 433–460.

[164] UIJLINGS, J. R. R., VAN DE SANDE, K. E. A., GEVERS, T., AND SMEULDERS, A. W. M. Selective search for object recognition. *International Journal of Computer Vision* (2013).

[165] VAN DE SANDE, K. E. A., GEVERS, T., AND SNOEK, C. G. M. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 9 (2010), 1582–1596.

[166] VAN DE SANDE, K. E. A., SNOEK, C. G. M., AND SMEULDERS, A. W. M. Fisher and vlad with flair. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014).

[167] VAPNIK, V. N. *The Nature of Statistical Learning Theory.* Springer Verlag, 1995.

[168] VEDALDI, A., GULSHAN, V., VARMA, M., AND ZISSERMAN., A. Multiple kernels for object detection. *Proceedings of the IEEE International Conference on Computer Vision* (2009).

[169] VIOLA, P., AND JONES, M. Robust real-time face detection. *International Journal of Computer Vision 57* (2004), 137–154.

[170] VIOLA, P., JONES, M. J., AND SNOW, D. Detecting pedestrians using patterns of motion and appearance. *Proceedings of the IEEE International Conference on Computer Vision* (2003).

[171] VIOLA, P. A., AND JONES, M. J. Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2001).

[172] VON HELMHOLTZ, H. *Treatise on Physiological Optics*. Leopold Voss, 1867.

[173] WALK, S., MAJER, N., SCHINDLER, K., AND SCHIELE, B. New features and insights for pedestiran detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2010).

[174] WANG, H., GOULD, S., AND KOLLER, D. Discriminative learning with latent variables for cluttered indoor scene understanding. *Proceedings of the European Conference on Computer Vision* (2010).

[175] WANG, L., ZHU, J., AND ZOU, H. The doubly regularized support vector machine. *Statistica Sinica 16* (2006), 589–616.

[176] WANG, X., HAND, T. X., AND YAN, S. An HOG-LBP human detector with partial occlusion handling. *Proceedings of the IEEE International Conference on Computer Vision* (2009).

[177] WATANABE, T., ITO, S., AND YOKOI, K. Co-occurrence histograms of oriented gradients for pedestrian detection. *Pacific Rim Symposium on Advances in Image and Video Technology* (2009).

[178] WERTHEIMER, M. Untersuchungen zur lehre von der gestalt I. prinzipielle bemerkungen. *Psychologische Forschung 1* (1922), 47–58.

[179] WERTHEIMER, M. Untersuchungen zur lehre von der gestalt II. prinzipielle bemerkungen. *Psychologische Forschung 4* (1923), 301–350.

[180] WESTON, J., MUKHERJEE, S., CHAPELLE, O., PONTIL, M., POGGIO, T., AND VAPNIK, V. Feature selection for svms. *Proceedings of the Conference on Advances in Neural Information Processing Systems* (2000).

[181] WILLIAMS, C. K. I. Prediction with gaussian processes: From linear regression to linear prediction and beyond. In *Learning and Inference in Graphical Models* (1997), Kluwer, pp. 599–621.

[182] WITKING, A. P., AND TENENBAUM, J. M. One the role of structure in vision. In *Human and Machine Vision.*, J. Beck, B. Hope, and A. Rosenfeld, Eds. Academic Press, 1983, pp. 481–543.

[183] WOLF, L., HASSNER, T., AND TAIGMAN, Y. Descriptor based methods in the wild. *Proceedings of the European Conference on Computer Vision* (2008).

[184] YARLAGADDA, P., AND OMMER, B. From meaningful contours to discriminative object shape. *Proceedings of the European Conference on Computer Vision* (2012).

[185] YUAN, J., WU, Y., AND YANG, M. Discovery of collocation patterns: from visual words to visual phrases. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2007).

[186] YUAN, J., YANG, M., AND WU, Y. Mining discriminative co-occurrence patterns for visual recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2011).

[187] ZHU, L., CHEN, Y., AND YUILLE, A. Learning a hierarchical deformable template for rapid deformable object parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence 99* (2009).

[188] ZHU, L., CHEN, Y., YUILLE, A., AND FREEMAN, W. Latent hierarchical structural learning for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2010), 1062 –1069.

[189] ZOU, H., AND HASTIE, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society* (2005), 301–320.