Inaugural-Dissertation

zur Erlangung der Doktorwürde der

Naturwissenschaftlich-Mathematischen
Gesamtfakultät

der

Ruprecht-Karls-Universität
Heidelberg

vorgelegt von
Artsiom Sanakoyeu

aus Horki, Belarus

Tag der mündlichen Prüfung:

Visual Representation Learning with Limited Supervision

by

Artsiom Sanakoyeu

Advisor: Prof. Dr. Björn Ommer

Abstract

The quality of a Computer Vision system is proportional to the rigor of data representation it is built upon. Learning expressive representations of images is therefore the centerpiece to almost every computer vision application, including image search, object detection and classification, human re-identification, object tracking, pose understanding, image-to-image translation, and embodied agent navigation to name a few. Deep Neural Networks are most often seen among the modern methods of representation learning. The limitation is, however, that deep representation learning methods require extremely large amounts of manually labeled data for training. Clearly, annotating vast amounts of images for various environments is infeasible due to cost and time constraints. This requirement of obtaining labeled data is a prime restriction regarding pace of the development of visual recognition systems.

In order to cope with the exponentially growing amounts of visual data generated daily, machine learning algorithms have to at least strive to scale at a similar rate. The second challenge consists in the learned representations having to generalize to novel objects, classes, environments and tasks in order to accommodate to the diversity of the visual world. Despite the evergrowing number of recent publications tangentially addressing the topic of learning generalizable representations, efficient generalization is yet to be achieved. This dissertation attempts to tackle the problem of learning visual representations that can generalize to novel settings while requiring few labeled examples.

In this research, we study the limitations of the existing supervised representation learning approaches and propose a framework that improves the generalization of learned features by exploiting visual similarities between images which are not captured by provided manual annotations. Furthermore, to mitigate the common requirement of large scale manually annotated datasets, we propose several approaches that can learn expressive representations without human-attributed labels, in a self-supervised fashion, by grouping highly-similar samples into surrogate classes based on progressively learned representations.

The development of computer vision as science is preconditioned upon the seamless ability of a machine to record and disentangle pictures' attributes that were expected to only be conceived by humans. As such, particular interest was dedicated to the ability to analyze the means of artistic expression and style which depicts a more complex task than merely breaking an image down to colors and pixels. The ultimate test for this ability is the task of style transfer which involves altering the style of an image while keeping its content. An effective solution of style transfer requires learning such image representation which would allow disentangling image style and its content. Moreover, particular artistic styles come with idiosyncrasies that affect which content details should be preserved and which discarded. Another pitfall here is that it is impossible to get pixel-wise annotations of style and how the style should be altered. We address this problem by proposing an unsupervised approach that enables encoding the image content in such a way

that is required by a particular style. The proposed approach exchanges the style of an input image by first extracting the content representation in a style-aware way and then rendering it in a new style using a style-specific decoder network, achieving compelling results in image and video stylization.

Finally, we combine supervised and self-supervised representation learning techniques for the task of human and animals pose understanding. The proposed method enables transfer of the representation learned for recognition of human poses to proximal mammal species without using labeled animal images. This approach is not limited to dense pose estimation and could potentially enable autonomous agents from robots to self-driving cars to retrain themselves and adapt to novel environments based on learning from previous experiences.

Zusamenfassung

Die Qualität eines Computer Vision Systems ist proportional zur Genauigkeit der Daten-Repräsentationen, auf welchen das System beruht. Das Erlernen von aussagekräftigen Repräsentationen von Bildern ist folglich das Herzstück fast jeder Computer Vision Applikation, einschließlich Bildsuche, Objekt-Detektion und -Klassifikation, Re-Identifikation von Menschen, Objekt-Tracking, Verstehen von Körperhaltung, Bild-zu-Bild Überführung oder Navigation von intelligenten virtuellen Agenten, um hierbei lediglich einige zu nennen. Tiefe Neuronale Netze sind am häufigsten vertreten unter den modernen Methoden des Representation Learning. Die Limitierung dabei ist jedoch, dass Methoden des Deep Representation Learning ein sehr großes Volumen an Trainingsdaten benötigen, welche vorab manuell gekennzeichnet werden müssen und somit ein Label erhalten. Offensichtlich ist die Annotation einer derart großen Datenmenge an Bildern unzumutbar aufgrund von Zeit- und Kosten-Restriktionen. Die Anforderung zur Bereitstellung annotierter Daten ist hierbei eine wesentliche Einschränkung bei der Entwicklung von Systemen zur visuellen Erkennung.

Um die exponentiell wachsende Menge an visuellen Daten bewältigen zu können, welche täglich neu entstehen, müssen Algorithmen für Maschinelles Lernen eine Skalierung mit ähnlicher Rate zumindest anstreben. Die zweite Herausforderung besteht daraus, dass erlernte Repräsentationen in der Lage sein müssen zu generalisieren angesichts neuer Objekte, Klassen, Umgebungen sowie Aufgaben, um der Diversität der visuellen Welt gerecht werden zu können. Trotz einer steigenden Anzahl an neuesten wissenschaftlichen Veröffentlichungen, welche sich tangential mit der Thematik des Erlernens von generalisierbaren Repräsentationen beschäftigen, besteht weiterhin die Problemstellung der effizienten Generalisierung.

Diese Dissertation ist bestrebt diese Problematik des Erlernens visueller Repräsentationen anzugehen, um eine effiziente Generalisierung zu ermöglichen, sodass neue Umgebungen verarbeitet werden können ohne die Notwendigkeit annotierter Daten. In dieser Arbeit betrachten wir die Grenzen existierender Ansätze für Supervised Representation Learning und schlagen ein Framework vor, welches die Fähigkeit zur Generalisierung erlernter Merkmale verbessert. Dies wird erreicht durch die Nutzung von Ähnlichkeiten zwischen Bildern, welche sich nicht durch die manuelle Annotation visueller Daten erfassen lassen. Darüber hinaus schlagen wir mehrere Ansätze zum Erlernen aussagekräftiger Repräsentationen vor, welche ohne Annotation auskommen und in selbst-überwachter Methodik konzipiert sind, indem hochgradig-ähnliche Stichproben gemeinsam gruppiert werden zu stellvertretenden Klassen basierend auf stufenweise-erlernten Repräsentationen. Somit wird die Restriktion der Notwendigkeit großer Datenmengen mit Annotation relaxiert.

Die Entwicklung von Computer Vision als Wissenschaft setzt nahtloses Vermögen einer Maschine voraus die Attribute eines Bildes zu registrieren sowie zu entflechten, was zuvor lediglich dem Vermögen von Menschen zugesprochen wurde. Infolgedessen wurde spezielles Interesse den bildenden Künsten zuteil sowie der

Analyse verschiedener Stilrichtungen, was weitaus komplexer ist als lediglich die Unterteilung von Bildern in Farben und Pixel. Ein maßgebender Test für dieses Vermögen stellt der Transfers von Stilrichtungen dar, was das Verändern des Stiles eines Bildes beinhaltet, ohne dabei den visuellen Inhalt zu verändern. Eine effektive Lösung für Stil-Transfer erfordert das Erlernen entsprechender Repräsentationen, welche es ermöglichen den Stil eines Bildes sowie den Inhalt zu entflechten. Individuelle künstlerische Stile beinhalten Eigenheiten, die Auskunft darüber geben welche Details für den Transfer beibehalten werden sollen und welche nicht. Eine weitere Herausforderung ist dabei, dass es quasi unmöglich ist pixelweise Annotationen zu erhalten hinsichtlich des Stils und wie dieser angepasst werden sollte. In dieser Arbeit adressieren wir diese Problematik und schlagen einen unüberwachten Ansatz vor, welcher die Enkodierung von Bildinhalten ermöglicht, die notwendig sind für einen bestimmten Stil. Der vorgeschlagene Ansatz tauscht den Stil eines Eingabe-Bildes indem zunächst die Repräsentation des Inhalts mit einem stil-bewussten Verfahren extrahiert wird. Anschließend wird mittels des neuen Stils gerendert unter der Verwendung eines stil-spezifischen Decoder Netzwerks, sodass überzeugende Resultate bei der Bild- sowie Video-Stilisierung erreicht werden.

Abschließend kombinieren wir Techniken für überwachtes und selbst-überwachtes Erlernen von Repräsentationen für die Aufgabenstellung zum Verstehen von Körperhaltungen von Menschen sowie Tieren. Die vorgeschlagenen Methoden ermöglichen den Transfer erlernter Repräsentationen zur Erkennung menschlicher Körperhaltungen zu denen von ähnlichen Säugetieren ohne hierbei annotierte Daten verwenden zu müssen. Dieser Ansatz ist hierbei nicht beschränkt auf Dense Pose Schätzung und könnte potenziell eine Reihe an autonomen Agenten befähigen, von Robotern bis hin zu autonomen Fahrzeugen, sodass diese in der Lage sind selbst weiter zu lernen und folglich sich an neue Umgebungen basierend auf vorher Erlerntem anpassen können.

Acknowledgements

First and foremost, I want to express my most generous gratitude to my advisor Prof. Dr. Björn Ommer, for giving me such a unique opportunity to pursue a Ph.D. in tremendously exciting topic, which became my passion. You have been an inexhaustive source of insights and motivation for me during all these years. I have learned a lot about scientific work from you, and I'm very thankful for the countless hours you spent discussing and shaping research ideas with me. I also want to thank Prof. Dr. Christoph Schnörr for agreeing to review my thesis.

I want to thank Prof. Dr. Vladimir Kotov for giving me the opportunity to work on exciting summer research projects during my undergraduate studies and making it possible for me to collaborate with Dr. Aleh Kryvanos, which led me to life-changing choices. I want to express immense gratitude to Dr. Aleh Kryvanos for inspiring me to pursue a scientific career and engage in working towards a Ph.D. degree.

I am very grateful to my mentors at Facebook AI Research — Prof. Dr. Andrea Vedaldi and Dr. Natalia Neverova. I'm very impressed by your experience and research vision and very happy that I had such a fantastic chance to collaborate with you.

I have also been fortunate to be surrounded by amazing people during my work at Heidelberg University. Many thanks to all my colleagues Dima, Sabine, Uta, Biagio, Timo, Boris, Miguel, Niko, Masato, Patrick for remarkably productive work-related discussions, countless funny moments during coffee breaks, and for our always enjoyable social events outside work. I am also very grateful to Pamela Schmidts and Barbara Werner for all their support and for shielding us from the administrative processes, allowing us to focus purely on research work.

Thanks to all my collaborators, who contributed to the work described in this dissertation. It was such a joy to strive to finish our papers on time during conference deadlines with Miguel Bautista, Dima Kotovenko, Vadim Tschernezkim, and Pingchuan Ma. I will always remember our deadline working sessions that could last for more than 24 hours straight and beer in BräuStadel afterward in the morning. Special thanks to Dr. Miguel Bautista for starting this tradition of night shifts and to Dima Kotovenko for picking it up from me.

Thanks to HGS MathComp for supporting me with a scholarship and especially to its administrative director Dr. Michael Winckler for always being open to conversation and for his invaluable pieces of advice.

I also want to thank all my friends who were with me during challenging moments and celebrated my accomplishments with me. Thanks to Anastasia Evsyukova, Dima Kotovento, Ivan Titov, Karen Khachikyan, Artsiom Masiura,

Eugene Klyshko, Pavel Orsich, Aliaksei Novikau, Olga Chernoskutova, Eldar Insafutdinov. Thanks to my SPbU friends - Dimon Maksimov and Jeka Ikonnikov, for always memorable trips to Berlin. Thanks to Daniyar Chumbalov for bringing fun to the FB office in London and becoming my good friend. Special thanks to Anastasia, Eldar, and Dima for proofreading parts of this thesis. Thanks to our Happy Hour group – Diego and Priscila Costa, Mangayarkarasi Rajakrishnan, Lutz Büch, Asha Roberts, Fereydoon Taheri, Victoria Ponce for our memorable gatherings. Also, I want to send thanks to Punjab Curry at Bergheimer Straße for providing up with calories when I did not have time to cook myself during periods of intensive work.

Finally, I would like to thank and express my deepest gratitude to my parents Sviatlana and Aleh. This work will not be possible without their unconditional love, encouragement, and advice during every challenging moment of my life as a Ph.D. student. This thesis is dedicated to them.

Contents

1 Introduction

> Labels are the opium of a machine learning researcher.
>
> *Prof. Dr. Jitendra Malik, 2019.*

An ambitious goal of Computer Vision is to create an Artificial Intelligence (AI) system which is capable of an automatic understanding of the visual world around us, mimicking the human visual system. Such an AI system should also be able to plan and make decisions based on perceived visual observations. Human vision is capable of extracting conceptual information from the images even when they are present for a very short amount of time and test questions about them are not accessible in advance [218]. In other words, visual input is converted into some abstract representation that is general enough to be easily adapted for various downstream tasks. Similar behavior is desired for AI systems. To achieve this goal, AI needs to identify the underlying explanatory factors hidden in the observed low-level sensory data and build abstract representations of the input [15].

The process of manually searching for an appropriate input data representation by designing preprocessing and data transformations is called *feature engineering*. Data representations (features) extracted in this way are subsequently used for efficient training of the predictor that is supposed to output the final solution for a problem (e.g., classification or regression). Prominent works on handcrafted features include bag-of-features (BOF) [250], scale-invariant feature transform (SIFT) [178], histogram of oriented gradients (HOG) [47], and vector of locally aggregated descriptors (VLAD) [127]. Feature engineering is a very challenging process that heavily depends on a dataset and target task and often requires domain knowledge. Moreover, manual feature engineering is very laborious because ad-hoc solutions suitable for one task do not necessarily generalize to another; hence, the features have to be re-designed for every novel task. To circumvent the need for such a manual design of visual recognition pipelines, it is natural to seek an approach that would be able to extract useful representations from the data automatically.

In the past twenty years, we have witnessed a rapid increase in routinely available computational power as well as an emergence of effective methods for training deep neural network architectures [86, 149, 153, 154, 157] that can extract hierarchical features directly from raw input data. Therefore, recent methods do not require feature engineering, but rather *learn* features from large-scale data. Hence, end-to-end learnable systems substitute carefully handcrafted pipelines.

Deep learning systems, however, require enormous amounts of manually annotated training data. Besides the fact that manual annotations are very tedious and expensive to collect, it is infeasible to label every potential object of interest. Furthermore, it could even be impossible to collect a complete set of annotations required for learning some complex tasks like autonomous driving or image-to-image translation. Therefore, the traditional supervised learning paradigm suffers from such severe limitations and cannot scale. In recent years, we noticed the emergence of large amounts of easily accessible unlabeled and weakly labeled (using hashtags or meta-information) visual data on the internet. This in turn has spurred interest in the methods, which can learn with limited [28, 233, 304] or no human supervision at all [11, 51, 55, 101, 206, 285].

So, the question remains: How can AI learn without supervision? Biological systems are an inexhaustible source of inspiration in the field of AI research [251]. The core idea of learning without labels is learning to represent the environment before addressing any specific task. For example, human infants gain a basic understanding of the world before the age of 12-15 months purely in a *self-supervised* regime – by observing their surroundings and repeated interactions with various objects in their environment. In such a way, by the age of 7-9 months, they acquire basic object manipulation skills, and by the age of 12 months they master crawling and learn about elementary physical phenomena including gravity, inertia, object collision, and occlusion [10, 89, 232]. The basic learning mechanism at this stage is the following: First, build a simple hypothesis about a certain phenomenon (e.g., whether a toy car will roll down the ramp after a hit by a cylinder, or not) and refine the hypothesis during further experience by discovering and incorporating extra influencing factors (e.g., the size of the object hitting the car can impact the length of the car's trajectory). Starting from such easy concepts, toddlers can develop more complex skills as they gain more understanding of how the world works. When a child has a solid background knowledge about its environment, it can learn new tasks faster. For example, an adult need only show a cat to a 2-year-old child once for the child to learn to recognize and categorize any other cat later. There is no need to show thousands of cats for this learning to take place. This type of learning is dubbed *few-shot learning* in the machine learning field, and it can be used to measure how well the learned representation of the world generalizes to novel objects and tasks [70, 75, 150].

Having said that, our desired AI system should be able to learn like the human child by continuously observing and revisiting its hypotheses about the world with very limited supervision. More specifically, it should have the following traits:

1. Learning with minimal supervision or without supervision at all.

2. Learned representations should be compact and encode explanatory factors of the observation. These factors should be ideally disentangled.

3. Ability to generalize from previous experience. Learned representations should be general and serve as a basis for faster learning of novel tasks and attaining new skills, with only a few labeled examples provided.

Methods that do not require manual annotations for learning are called *unsupervised* or *self-supervised*. One of the main goals for these methods is to learn a compact representation of the world before learning downstream tasks. Such methods employ intrinsic supervisory signals contained in the input data. For example, learning can be achieved by predicting the outcome of an event and refining the hypothesis after every observation or by solving some surrogate tasks for which the labels can be easily obtained from the data automatically (e.g., erasing parts of the image and predicting the missing parts [215]). An important advantage of such methods is that they are free from human bias, which is introduced by manually curating and labeling the data. However, the performance of existing self-supervised methods still lags behind the fully supervised approaches for many computer vision tasks and there is huge room for improvement.

In this thesis, we explore the limitations of existing supervised and self-supervised representation learning techniques. We propose novel learning approaches to improve the generalization performance in scenarios when manual annotations are limited (*metric learning* in particular) or when no annotations are available at all (*self-supervised learning*). Then we present a novel method that allows transferring knowledge attained by learning on a labeled source domain to an unlabeled target domain in *self-training* fashion.

Finally, we tackle learning disentangled representations of image attributes in an unsupervised regime. In particular, we consider a *style transfer* task [79] — exchanging the artistic style of an image while keeping its content, for which the ability to decompose and separately represent the image content and style is essential. This task is especially challenging because the direct supervision is impossible to attain due to the infeasibility to manually segment out the artistic style from the content of the artworks. Moreover, particular artistic styles come with idiosyncrasies that affect which content details are preserved from the real scene and which discarded. Hence, there is a demand for a flexible approach respecting such peculiarities. This dissertation presents a novel approach for learning such a content representation that respects idiosyncrasies of a target style and enables compelling stylization quality without any supervision. Furthermore, the suggested representation enables real-time high-resolution image and video stylization.

1.1 Representation learning

This thesis addresses two major problems: learning generalizable representations and reducing the number of annotations required for learning. In particular, we first tackle a supervised scenario where the representations are learned using metric learning methods. Then we approach a self-supervised scenario where the task is to learn merely from raw data by discovering intrinsic structures and

regularities. Next, we combine the latter two scenarios and propose a self-training approach to adapt the representation attained by supervised pretraining to solve a task on the target domain which does not have any ground truth annotations. After that, we study how to learn disentangled representations for the style transfer task by decoupling an image into the content and style representations. The proposed unsupervised solution circumvents the impossibility of collecting manual annotations for such an image-to-image translation task.

In the following subsections, we give an introduction to the aforementioned problems and establish the context of the work performed in this thesis.

1.1.1 Metric Learning[1]

The exponential increase of easily accessible digital images and the importance of image retrieval and related tasks for numerous applications call for fast and scalable representation learning approaches. Deep Metric Learning (DML) is a class of representation learning approaches which aims to learn such a *representation space* (often called *embedding space*[2]) so that a predefined distance measure in this space can be used to describe the similarity between an arbitrary pair of data points. DML has for long been of major interest for the vision community, due to its broad applications including image search and object retrieval [14, 198, 210, 274, 281, 295], zero-shot and single-shot learning [210, 274], keypoint descriptor learning [248], face verification [41, 241, 282, 292], vehicle identification [42, 171, 311], near duplicate detection [320], visualization of high-dimensional data [95, 182], and clustering [106]. The typical scenario for DML is to train on one set of categories and evaluate on a completely different set of test categories. Therefore, the main goal of DML is to learn such a representation space that is able to generalize to previously unseen images and categories.

Formaly, given an image dataset $X = \{x_1, \ldots, x_n\} \subset \mathcal{X}$ with the corresponding class labels $Y = \{y_1, \ldots, y_n\}$, where $\mathcal{X}$ is the original RGB space, the task of DML is to learn a similarity measure between an arbitrary pair of images x_i and x_j. During training, the target similarity is provided by a user in the form of discrete ground-truth classes, since it is infeasible to obtain continuous ground-truth similarity scores. We then need to learn a mapping f from images to a representation space so that a predefined distance measure in this space captures the desired similarity between the images, i.e., the distances between images from the same class should be small, and the distances between images from different classes should be large. The mapping f is learned directly from RGB values using a Convolutional Neural Network (CNN) which maps an RGB image x_i onto a point $f(x_i)$ in the d-dimensional embedding space $\mathbb{R}^d$. The distance between two points

[1] The content of this subsection is partially based on the author's paper which is currently under review, "Improving Deep Metric Learning by Divide and Conquer" [239].

[2] We will use the terms *representation* and *embedding* interchangeably in the context of Deep Metric Learning throughout this thesis.

i and j in the embedding space is then computed using a predefined distance measure, e.g. Euclidean distance $d_{i,j} = ||f(x_i) - f(x_j)||_2$.

Loss functions. The Deep Metric Learning task implies specific loss functions which can be used to learn the mapping f satisfying our needs. And there is a large body of work focusing on designing new loss functions for Deep Metric Learning. One of the most renowned loss functions is triplet loss [241, 291]. Let us consider a tuple of images (x_a, x_p, x_n), where x_a is an anchor image, x_p is a positive image from the same class as the anchor image and x_n is a negative image from any other class. The triplet loss pushes the anchor image x_a closer to the positive image x_p and further from the negative image x_n and is defined as

$$l_{\text{triplet}}(a, p, n) = \left[d^2_{a,p} - d^2_{a,n} + \alpha \right]_+, \tag{1.1}$$

where $[\cdot]_+$ denotes the positive part and α is the margin hyperparameter.

Beyond the triplet loss, there is a pletora of diferent loss functions [198, 252, 253, 274, 280, 288]. For example, Facility Location loss [253] optimizes a clustering quality metric, Histogram loss [274] minimizes the overlap between the distribution of positive and negative distances, and Angular loss [280] imposes extra geometrical constraints in the embedding space. LiftedStruct [210] and N-pairs [252] losses introduce a soft formulation of the triplet loss [291] replacing hinge function with Neighbourhood Component Analysis (NCA) [230] formulation which does not require tuning of a margin parameter. Recently, Yu et al. [310] modified N-pairs loss by introducing a margin and a temperature scaling in the objective and an extra loss term which penalizes high intra- and inter-class pairwise distance variance. Proxy-based losses [8, 143, 198] further extend the NCA paradigm by computing proxies (prototypes) for the training classes in the dataset and optimizing the distances to these proxies using the NCA objective[230]. Proxy-based losses are closely related to the classification-based Deep Metric Learning methods [173, 279, 315]. In this case, the training images are classified using the softmax function, where the columns of the weight matrix of the classification layer represent the prototypes for the classes. Magnet loss [225] is similar to proxy-based losses, but it does not learn the class prototypes. Instead, it splits every ground truth class into sub-classes by clustering and pushes training samples closer to the precomputed centroids of the corresponding sub-classes. Wen et al. [292] proposed to use a center loss jointly with the softmax classification loss to enforce more clustered representations by minimizing the Euclidean distance between the image embeddings and the learnable centroids of the corresponding classes. MIC [227] models visual characteristics shared across classes by utilizing an extra surrogate loss discriminating between data clusters. SoftTriple Loss [219] upgrades the softmax classification loss by learning multiple prototypes for each class, allowing to capture several modes for the classes with high intra-class variance. The major drawback of proxy- and classification-based losses is the limited scalability with respect to the number of classes. Another type of loss is FastAP [179] which aims at

ranking the images by optimizing the non-differentiable Average Precision (AP) measure. The authors use a probabilistic interpretation of the AP and approximate it by distance quantization and histogram binning.

Our work, which we present in Chapter 2, is orthogonal to all aforementioned approaches, as it provides a framework for learning a distance metric that is independent of the choice of a particular loss function.

Informative sample mining. It is common for metric learning methods to use pairs [20, 313] or triplets [112, 241, 281, 291] of samples. Some even use quadruplets [274] or impose constraints on tuples of larger sizes [105, 210, 252]. Using tuples of images as training samples yields a huge amount of training data. However, only a small portion of the samples among all N^p possible tuples of size p is informative because most of the tuples produce very low loss and insufficient gradient for learning. A lot of works attempted to explore how to select the most informative tuples for training, and such approaches can be categorized into two groups: (a) methods which focus on meaningful sample mining within a randomly drawn mini-batch, and (b) the methods which mine the entire dataset but require computationally expensive preprocessing steps.

The first group of methods strives to find informative samples within a randomly drawn mini-batch (*local mining*). Some of the methods from this category utilize all pairwise relationships within a mini-batch [49, 210], others mine hard negative [105, 241] or easy positive pairs [302]. Wang et al. [287] consider all possible pairs within the batch but set the weights for negative pairs as the exponent of the margin violation magnitude. MS-loss [288] generalizes tuple-based losses and reformulates them as different weighting strategies of positive and negative pairs within a mini-batch. Wu et al. [296] sample negative examples uniformly according to their relative distance to the anchor, and, recently, Roth et al. [228] proposed to learn the distribution for sampling negative examples instead of using a predefined one. Deep Adversarial Metric Learning (DAML) [57] generates synthetic hard negatives for the current mini-batch using adversarial training. Hard Triplet Generation (HTG) [319] also employs adversarial training to alter a given triplet by pushing embedding vectors of the images from the same class apart while pulling embedding vectors of the images from different classes closer. Deep Variational Metric Learning (DVML) [169] assumes that the distribution of intra-class variance is independent on the class label and makes a negative example harder by adding a variance component sampled from the learned distribution. Another approach that synthesizes hard negatives is Hardness-Aware Deep Metric Learning (HDML) [321], it uses linear interpolation to move the negative example closer to the positive one in the embedding space, thus increasing the triplet hardness. The drawback of the *local mining* methods is the lack of global information while having only a local view of the data based on a single randomly-drawn mini-batch of images. As a consequence, the performance of such approaches strongly depends on the mini-batch size, which is limited by the GPU memory size.

The mining approaches in the second group have a global view of the data and utilize the entire dataset for finding samples that provide the largest training signal (*global mining*). Wang et al. [281] relies on a highly optimized handcrafted "golden feature" to compute the pairwise and unary relevance scores for all images in the dataset. The closer the image to the class centroid, the higher the unary relevance it gets. The anchor, positive and negative examples are then sampled according to the relevance scores. The shortcoming of this approach is that the "golden feature" is expensive to compute, difficult to develop, and is crafted for a specific dataset. Later works [82, 99, 122, 229] do not rely on handcrafted features and mine hard negative examples in the learned embedding space. However, these sampling techniques require either running an expensive preprocessing step (quadratic in the number of data points) for the entire dataset and for every epoch [82, 99, 122, 229] or utilize additional meta-class labels [24] for hard sample mining. Suh et al. [257] sample negative pairs of images from the most confused classes. But this approach requires a joint training of the classification head using softmax loss and the embedding layer using a triplet loss. The approach proposed in this thesis (in Chapter 2) can efficiently alleviate the problem of abundance of easy samples by jointly dividing the data and the representation space into subparts during training and does not require an extra classification head. Also, in contrast to [24], our method is designed with the idea of using less manual annotations in mind and does not rely on meta-class labels. Moreover, our approach is complementary to the *local mining* methods and can be used jointly with them.

Ensembling for deep metric learning. Another line of work in Deep Metric Learning is ensemble learning [76, 94, 212, 311]. Previous works [212, 311] employ a sequence of "learners" with increasing complexity and mine samples of different complexity levels for the next "learners" using the outputs of the previous learners. Hard-aware Deeply Cascaded Embedding (HDC) [311] uses a cascade of multiple models of a specific architecture and trains earlier layers of the cascade with easier examples while harder examples are harnessed in later layers. Boosting Independent Embeddings Robustly (A-BIER) [212] applies a gradient boosting learning algorithm to train several learners inside a single network in combination with an adversarial loss [78, 87]. Attention-based Ensemble (ABE) [144] introduces soft attention in the intermediate convolutional feature maps to focus on different parts of the objects combined with extra divergence loss to diversify the attention. Deep Randomized Ensembles for Metric Learning (DREML) [301] and Ensemble Deep Manifold Similarity (EDMS) [8] train multiple networks on random splits of the data using variants of the ProxyNCA [198] loss. The downside of these approaches is the drastic increase in the number of parameters and the computational cost. The key difference of the aforementioned approaches to the one proposed in this thesis (see Chapter 2) is that we do not have an ensemble of the networks, but rather train multiple "learners" within a *single* network by splitting the representation space and clustering the data so that each "learner" is assigned to the specific subspace and the corresponding portion of the data. The "learners" are jointly trained on

non-overlapping chunks of the data, which reduces the training complexity of each individual "learner", facilitates the learning of decorrelated representations, and can be easily parallelized. Moreover, our approach does not introduce extra parameters during training since we do not alter the architecture and utilize only a single network. It does not require any elaborate loss functions but can be applied to arbitrary losses and any existing network architectures.

1.1.2 Self-supervised Learning

Nowadays, with an estimated 83% of the worldwide population having mobile broadband internet subscription [256] and a wide spread of portable digital cameras, users are uploading millions of hours of video [255] and billions of images to the internet daily [120]. The amount of user-generated data hereby increases exponentially. However, internet data rarely contains precise labels, which are so crucial for supervised learning. Thus, such an abundance of unlabeled visual data sparked a strong interest in scalable unsupervised representation learning approaches.

The key to learning from raw data is trying to understand it. In order to profoundly understand and explain the observed data, we need to discover the hidden causes and explanatory factors which led to these observations in the first place. In other words, unsupervised learning approaches strive to reveal unknown properties of the source probability distribution that generated the data. In this way, the observed data can be represented as a function of a smaller number of explanatory factors, which means that a machine learning system can conceptualize the data and build compact representations, which in turn can be used for various downstream tasks.

Unsupervised representation learning approaches can be divided into *shallow (classical) methods* and *deep methods* which are based on neural networks. While in this thesis we focus on the latter methods, we will also briefly discuss the former. Classical unsupervised learning approaches include cluster analysis [98], various data decomposition methods which discover the most meaningful components (principal component analysis (PCA) [189], independent component analysis (ICA) [44], non-negative matrix factorization (NMF) [155], etc.) and dimensionality reduction methods (multidimensional scaling (MDS) [189], Isometric feature mapping (Isomap) [259], t-SNE [182], Uniform Manifold Approximation and Projection (UMAP) [192], and more). While these methods can perform reasonably well for low-dimensional inputs, they usually suffer from the curse of dimensionality in high-dimensional data, e.g., images which could have millions of dimensions[3]. Moreover, these methods often require a reliable measure of similarity between pairs of inputs, which is actually the task for the representation learning approaches.

Deep unsupervised representation methods are also dubbed as *self-supervised* because the supervision is usually obtained from the data automatically. The

[3] A one-megapixel RGB image has 3 million dimensions.

main idea behind self-supervised methods is to construct a *pretext/surrogate task* for which the supervisory signal can be algorithmically produced from the raw input data instead of labeling the data manually.

Pretext tasks are designed in a way that in order to solve them, the model is required to learn an efficient visual representation. One of the approaches to construct a pretext task is to drop or distort parts of the input image and ask the model to reconstruct it. For example, Larsson et al. [151] and Zhang et al. [316] proposed to use image colorization as a pretext task. The input image is converted to grayscale and the model is tasked to predict the original colors for every pixel. Clearly, to solve this task, the model has to understand the contents of the image. Gidaris et al. [84] tasks the network to predict 2d rotations that are applied to the input image. This formulation, despite its simplicity, forces the network to learn surprisingly good semantic features of the images. Learning the spatial context of the image can be employed as a useful pretext task too. For instance, an image inpainting pretext task was proposed in [215], where an arbitrary region in the image is dropped and the model learns to generate the contents of this region conditioned on its surroundings. Doersh et al. [51] and Norooze et al. [206] split the image in a grid of patches and use the relative location of the image patches as a learning cue. Another source of free supervision is the temporal context in videos. Misra et al. [196] and Brattoli et al. [18] learn representations by identifying whether the sequence of frames from a video is in correct temporal order or if it is permuted. Wang et al. [285] go further and use spatiotemporal signals in videos for learning. They propose to track moving objects in the videos and learn robust features by enforcing the representation of the cropped-out objects to stay unchanged across different frames. Büchler et al. [22] combine ideas from [206] and [18] and learn by reconstructing the correct ordering of the visual data in both temporal and spatial domain while using reinforcement learning to sample the permutations of appropriate difficulty during training. Temporal coherence between consecutive frames in the ego-centric monocular video allows to obtain a free learning signal by estimating the ego-motion and comparing the consecutive frames reprojected on one another [186, 324]. Another prominent pretext task is counting objects within the image [207] relying on the equivariant transformations of the image, e.g., the total sum of the object counts in the 4×4 grid of image tiles should equal to the count in the entire image.

Differently from the aforementioned works, in Chapter 3, 4 we propose a novel method that leverages automatically discovered compact groups of semantically related images (i.e., cliques) as surrogate classes for self-supervised learning [11, 236]. Cliques are built based on the available image features, for instance, off-the-shelf handcrafted features or deep representations from a randomly initialized CNN. Classification, whether the sample belongs to a clique (surrogate class), serves as a pretext task. Concurrent works [160, 307] also group semantically similar samples in the feature space to create surrogate labels. However, Li et al.[160] formulate a pretext task as a binary classification to decide if the two images are similar and, therefore, mine only pairs of related images, which hinders the

effective modeling of the intra-class variance where more positive samples are needed. Yang et al. [307] partition the entire dataset using clustering hence risking to group samples which are not semantically related as the distances in the original feature space of randomly initialized CNN are not always reliable [11]. Later, Caron et al. [27] extend our work by proposing an optimized pipeline for training on larger datasets using deeper network architectures and, instead of building compact cliques of images, clustering the data with K-means [183] similar to [307]. Asano et al. [7] further improve on [27] by substituting the K-means clustering with approximately solving an optimal transport problem to produce surrogate classes of samples.

Another group of self-supervised methods aims to learn representations invariant to various image perturbations by regarding every image and its random augmentations as a separate surrogate class. Dosovitskiy et al. [55] learn a classifier that distinguishes between these surrogate classes. But such an approach does not scale well because the number of the classifier parameters grows linearly with the number of instances in the training set. Recently, contrastive self-supervised learning approaches [36, 37, 38, 91, 101, 104, 297] gained popularity and showed very compelling results bridging the gap between supervised and unsupervised pretraining for various Computer Vision tasks. Inspired by Deep Metric Learning approaches, they use distance-based losses, which enforce a sample to be close to its augmented view in the representation space enabling efficient scaling to very large unlabeled datasets. Misra et al. [195] outperform for the first time supervised pre-training for object detection task evaluated on Pascal VOC07 dataset [66], while Grill et al. [91] outperform the supervised pretraining on ILSVRC-2012 [48] dataset in the setting when only a subset of less than 10% of original ILSVRC-2012 training labels are available. However, the driving force of these approaches is carefully designed augmentations, the optimal configuration of which may be different for different datasets. Moreover, an intra-class variance cannot be successfully captured by pulling together only a single image and its perturbations, which our approach in Chapters 3, 4 alleviates by finding compact cliques of samples.

Generative models. Generative models can also be used to learn representations without supervision. For example, Autoencoders [111] learn low-dimensional features, which would allow to precisely reconstruct the input. To make the learned representations more robust, the input images may be corrupted, and Autoencoder is required to restore the original input [276] (Denoising Autoencoder). The inpainting approach of Pathak et al. [215] is similar in principle, but utilizes larger image corruptions, dropping entire patches from images.

Generative Adversarial Networks (GANs) [88] is another powerful unsupervised learning approach. GANs learn the distribution of natural images by generating them in an adversarial mini-max game [88]. A GAN model consists of two separate networks – a generator and a discriminator which compete with each other. The goal of the generator is to generate fake images mimicking real images as close as possible, and the discriminator strives to discern fake images from real ones.

The optimal solution lies in the equilibrium point [88], but in practice, it is never reached, and the discriminator usually wins as the task of distinguishing between real and fake images is much easier than actually producing realistic fakes. There are several attempts to suit GANs for learning representations [19, 53, 220], however, they are usually inferior to the approaches based on the discriminative pretext tasks discussed above because they have to focus on the generation aspect, often requiring to capture low-level information about the images which could be excessive for tasks like classification or detection and impair generalization.

Evaluation of self-supervised learning methods. In contrast to the supervised approaches where the performance is measured by the accuracy with which the model can predict a target value given an observation, the measure of success for self-supervised approaches is how good the learned representations perform in novel settings rendering the ability to generalization to new tasks and previously unseen data. For example, learned representations can be evaluated by computing image retrieval performance on novel images using the learned representations or computing the performance on downstream tasks such as classification, segmentation, detection, or pose recognition, after finetuning the self-supervised pretrained network on these tasks.

1.1.3 Self-training

Getting high-quality data annotations is a laborious, error-prone, and expensive process, while the internet is abundant in unlabeled data. Techniques that involve training on the labeled data and at the same time make use of a large corpus of unlabeled data are called *semi-supervised* learning methods [28, 327].

Self-training [161, 205, 224, 267, 309] is a class of semi-supervised learning methods which is often called *teacher-student* learning. In self-training, the amount of available training instances is automatically extended by imputing labels for all unlabeled data. First, a *teacher* network is trained on the provided labeled dataset. Then it predicts pseudo-labels for unlabeled instances. After that, initially provided labeled data and pseudo-labeled data is jointly used to train a *student* model. To squeeze the last bits of performance, this teacher-student training process can be repeated several times by using the current student model as a new teacher. The main difference between self-training and other semi-supervised approaches like those based on entropy minimization [90, 156] and consistency regularization [16, 197, 233, 258] is that in self-training pseudo-labels are produced by a more accurate converged teacher model, while other approaches use the same model that is being trained to generate pseudo-labels for unlabeled data in each mini-batch.

Recently, Xie et.al [299] demonstrated the effectiveness of the self-training and achieved state-of-the-art accuracy on ImageNet [48] outperforming regular supervised training. Such remarkable performance of [299] was achieved by using an enormous unlabeled dataset with 300M images (JFT [40, 110]), making a student model larger than a teacher and improving the efficiency of student training. To

make the student model generalize better than the teacher, authors applied extra regularization techniques like dropout [254], stochastic depth [116] and aggressive image augmentations [46] during the training of the student. Another prominent work [304] experimented with an even larger unlabeled dataset of one billion images from Instagram. A slightly different approach to produce pseudo-labels for self-training is to propagate labels from annotated images to those without annotations [121]. The label-propagation is done on the nearest neighbor graph computed using the image embeddings extracted from the teacher network.

Predictions of a teacher model on an unlabeled dataset are not always correct and can contain errors. Several approaches have been proposed to prevent the injection of very noisy labels into student training. The first is to filter the pseudo-labels based on classifier score or confidence of a model, which we will describe in more detail in Chapter 5. The second approach is to average the predictions of an ensemble of several teacher models [9, 323] increasing the robustness of the produced pseudo-labels.

The self-training approaches discussed above aim at improving the generalization of the model on the same task for which the annotated data is available by leveraging additional unlabeled data. In this dissertation, we take it further and propose a method that allows the model that was initially trained to solve a source task, to adapt to a novel task for which no annotations are available (see Chapter 5). Specifically, we consider the problem of learning to recognize the pose of animals with as little supervision as possible. We study several strategies to transfer an existing dense pose [92] extractor for humans to chimpanzees for which we have zero ground truth annotations provided.

1.1.4 Style Transfer: Disentangling Content and Style

To understand why extracting and untangling different visual attributes is essential for various Computer Vision applications, especially for image editing, we first discuss general purpose image-to-image translation [34, 60, 107, 123] approaches and then focus on a particular problem of style transfer.

Image editing. An image exhibits different visual attributes like appearance, layout, shape, and style. Learning and disentangling these attributes is crucial for correct perception of objects by a visual recognition system [74]. It is even more critical to decouple visual attributes for image editing applications when only specific objects or object properties must be altered. Suppose a model could learn such representation of a human face, where different dimensions would be independent and correspond to variation of different face attributes such as the shape of the nose, ears, eyes, color of skin and hair, type of haircut. In that case, photorealistic image editing could be performed as efficiently as dragging a set of sliders.

A typical framework for image-to-image translation [33, 123, 283] is usually based on the Autoencoder architecture [111] with encoder and decoder networks. However, instead of learning to reconstruct the input from the representation produced by

the encoder, image-to-image translation methods aim to produce a transformed image that satisfies certain constraints enforced by carefully devised loss functions. Nevertheless, a lot of models are designed for generation of random images [139, 140] or images conditioned on class labels [19] and not for explicit editing of user-provided images [172, 214]. Such models do not have an encoder network, but only the decoder (also called generator), and generate images from a latent representation randomly sampled from a Gaussian distribution. For such models to edit an existing image, first, the image has to be projected into the representation space; after that, one can do some manipulations with the representation vector and decode it back into the RGB domain. Since the learned representation space is not easily interpretable, it is not clear how to edit images in this space. To understand how to change the representation vector to achieve desired image transformations, Upchurch et al. and Shen et al. [245, 272] proposed to discover linear directions in the representation space which are responsible for particular visual attributes. For example, for face editing, discovered directions in the representation space can correspond to head pose, smile, age, gender, and eyeglasses [245]. However, this approach requires a significant amount of annotations. Moreover, interpretable linear directions are not always possible to find, and some visual attributes can be highly correlated, e.g., older people are more likely to have glasses [245]. These shortcomings stem from the fact that the models mentioned above are not directly trained to produce representations with disentangled visual attributes and, hence, only limited control of the editing process can be achieved. Moreover, the quality of the results is not production-ready because the methods which are used to project an existing image to the representation space [2, 181, 325] of existing GAN models tend to produce non-realistic results and sometimes even lose the identity of an object in the input image (cf. [245]).

State-of-the-art approaches deliberately designed for *controllable* image editing rely on user-provided geometrical and color constraints such as hand-drawn sketches, or the semantic layout of the entire scene or its parts [213, 283], or collaging [217], i.e., when a user alters the source image by cutting and pasting some patches from another image. The model combines the source image and coarse user-provided input into a plausible output image. However, these methods do not exhibit disentanglement of the high-level semantic image attributes in the representation space, and the editing is possible only through tedious manual changes of the input image by a user.

Style Transfer. Style Transfer [79] is an image-to-image translation task where the goal is to change the style of an image while retaining its content. In other words, provided an input image x_c and a reference style s, the task is to extract the content information from the input and combine it with the reference style to generate a plausible stylization (the reference style can be provided as a single artwork image x_s or a collection of artworks $\{x_s^{(i)}\}$ of the same style [238]). Therefore, to enable effective style transfer, it is crucial to learn a disentangled representation that can decompose and separately represent the content and style of an image.

One of the earliest attempts to separate content from style was made by Tenenbaum et al. [260]. They used a factor analysis technique to model an observation as a bilinear combination of style and content vectors. However, the relationship between content and style is often more complex and cannot be captured by a simple bilinear model. Seminal work of Gatys et al. [79] proposed to represent the content of an image as activations extracted from intermediate convolutional layers of the 19-layer network VGG-19 [249] pretrained on ImageNet [48] and represent style as correlations of activations from the lower layers of the VGG-19. After extracting content representation $F(x_c)$ from the input image and a style representation $G(x_s)$ from the reference style image, the synthetic artwork is obtained by running iterative optimization procedure on the output image x_o by minimizing the difference between its content and style representations $F(x_o), G(x_o)$ with $F(x_c)$ and $G(x_s)$ correspondingly. However, such representations of image content and style are, obviously, not independent as they are derived from one another[4]. Moreover, since the same ImageNet-pretrained network is used to extract style representations from images exhibiting different artistic styles, there is no way to adapt to the peculiarities of particular styles. For instance, the degree of preservation of object details depends on the level of abstractness of style (see Chapter 6) and what is crucial for a realistic Baroque artwork of Vermeer may be completely unimportant for a Cubist artwork of Picasso. We address this issue, neglected by the existing approaches, in Chapter 6 by introducing an adaptive approach that untangles style and content conditioned on a particular style.

Though recent style transfer methods [59, 117, 130, 159, 164, 270] circumvent the prohibitively expensive optimization process of Gatys et al. [79] by approximating its solution with Autoencoder-based models. Yet, they rely on the loss functions exploiting the same style and content representations (as proposed in [79]), and, therefore, have similar drawbacks.

1.2 Contributions

This dissertation makes the following contributions:

- Novel easy-to-implement framework for supervised representation learning using divide and conquer paradigm, which significantly improves the state-of-the-art performance of existing representation learning methods based on DML. Our approach utilizes the representation (embedding) space more efficiently by jointly splitting the representation space and data into smaller sub-problems. The proposed framework increases the convergence speed and improves the generalization since the complexity of each sub-problem is reduced compared to the original one.

[4]Style representation uses the convolutional layers of the VGG-19 that precede the layers used to compute content representation.

- Novel self-supervised approach for visual representation learning based on a surrogate categorization task. Weak estimates of similarities between the samples, induced by the currently available representations, are used to define surrogate classes as a compact group of mutually related samples. A single optimization problem to extract batches of samples with mutually consistent relations is proposed to mitigate the effect of conflicting relations. A CNN is trained to consolidate the transitivity relations within and between surrogate classes. As a result, it learns a single representation for all samples without the need for manual annotations.
- We further boost the self-supervised representation learning by incorporating unreliable and mutually contradicting relationships between samples (which cannot be used to build surrogate classes). Proposed method leverages the local partial order of samples to surrogate classes. Self-supervised representation learning is then formulated as a partial ordering task with soft correspondences of all samples to surrogate classes. The representation learning and building the surrogate classes are integrated into a single model and are optimized jointly.
- Novel self-training approach for transferring the knowledge existing in dense pose recognition for humans, as well as in more general object detectors and segmenters, to the problem of dense pose recognition in other proximal animal classes, such as chimpanzees. We (1) establish a DensePose [92] model for the new animal which is also geometrically aligned to humans, (2) introduce a multi-head R-CNN [100] architecture that facilitates transfer of multiple recognition tasks between classes, (3) find which combination of known classes can be transferred most effectively to the new animal and (4) use self-calibrated uncertainty heads to generate pseudo-labels graded by quality for training a model for this class. We also introduce two benchmark datasets labeled in the manner of DensePose [92] for the class chimpanzee and use them to evaluate our approach, showing excellent transfer performance.
- Novel self-supervised approach for the real-time, high-resolution stylization of images and videos, which learns the representation of image content disentangled from the style representation. The proposed method can learn style representation from a collection of style images; it does not require tedious manual labels, reducing the bias introduced by the annotators. Furthermore, we propose a new quantitative measure for evaluating the quality of stylized images. The approach achieves state-of-the-art results in image and video stylization.

1.3 List of published research papers

The remaining chapters of this dissertation are based on the following publications of the author. The sign * indicates equal contribution of the first two co-authors.

1. **CliqueCNN: Deep Unsupervised Exemplar Learning**
   Miguel A. Bautista*, Artsiom Sanakoyeu*, Ekaterina Tikhoncheva, and Björn Ommer
   Advances in Neural Information Processing Systems (NeurIPS) 2016

2. **Deep Unsupervised Similarity Learning Using Partially Ordered Sets**
   Miguel A. Bautista*, Artsiom Sanakoyeu*, and Björn Ommer
   IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017

3. **Deep Unsupervised Learning of Visual Similarities**
   Artsiom Sanakoyeu, Miguel A. Bautista, and Björn Ommer
   Pattern Recognition 78, 2018

4. **A Style-aware Content Loss for Real-time HD Style Transfer**
   Artsiom Sanakoyeu*, Dmytro Kotovenko*, Sabine Lang, aand Björn Ommer
   European Conference on Computer Vision (ECCV) 2018

5. **Divide and Conquer the Embedding Space for Metric Learning**
   Artsiom Sanakoyeu*, Vadim Tschernezki*, Uta Büchler, and Björn Ommer
   IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019

6. **Transferring Dense Pose to Proximal Animal Classes**
   Artsiom Sanakoyeu, Vasil Khalidov, Maureen S. McCarthy, Andrea Vedaldi, and Natalia Neverova
   IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2020

The author has also contributed to the following relevant publications. However, they are not discussed in this thesis.

7. **A Content Transformation Block for Image Style Transfer**
   Dmytro Kotovenko, Artsiom Sanakoyeu, Pingchuan Ma, Sabine Lang, and Björn Ommer
   IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019

8. **Content and Style Disentanglement for Artistic Style Transfer**
   Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Björn Ommer
   IEEE International Conference on Computer Vision (ICCV) 2019

9. **Semi-supervised Segmentation of Salt Bodies in Seismic Images Using an Ensemble of Convolutional Neural Networks**
   Yauhen Babakhin, Artsiom Sanakoyeu, and Hirotoshi Kitamura
   German Conference on Pattern Recognition (GCPR) 2019

1.4 Thesis Organization

The remainder of this dissertation is organized as follows. In Chapter 2 we introduce our framework for supervised learning of representations with DML. We

demonstrate that it achieves state-of-the-art performance on five widely-used DML benchmarks. Chapter 3 presents a novel self-supervised representation learning approach based on a surrogate classification. The competitive performance of the approach is demonstrated on detailed posture analysis and object classification downstream tasks. In Chapter 4 we further extend this self-supervised approach by incorporating all available data samples into the learning procedure, including unreliable ones using their partial ordering. In Chapter 5 we introduce a self-training approach for transferring representation learned for dense pose recognition of humans to the problem of dense pose recognition of chimpanzees. We introduce two datasets with densely annotated poses of chimpanzees and achieve compelling performance on them. Next, in Chapter 6 we study learning an efficient content representation for Artistic Style Transfer and describe a novel approach for the real-time, high-resolution stylization of images and video. We discuss and conclude the work done in this thesis in Chapter 7.

2 Divide and Conquer the Embedding Space for Metric Learning[1]

In this chapter, we introduce a novel divide and conquer framework for learning generalizable image representations using Deep Metric Learning (DML) approaches.

Deep metric learning methods learn such representation which allows to measure similarities or distances between arbitrary groups of data points, which is a task of paramount importance for a number of computer vision applications. Deep metric learning has been successfully applied to image search [14, 114, 210, 281], person/vehicle re-identification [41, 171, 311], fine-grained retrieval [198], near duplicate detection [320], clustering [253] and zero-shot learning [210, 274].

The core idea of deep metric learning is to pull together samples with the same class label and to push apart samples coming from different classes in the learned embedding space. An embedding space with the desired properties is learned by optimizing loss functions based on pairs of images from the same or different class [14, 20, 95], triplets of images [112, 241, 281] or tuples of larger number of images [12, 115, 252, 274], which express positive or negative relationships in the dataset.

Existing Deep Metric Learning approaches strive to directly learn a single embedding space for all samples from the training data distribution. The ultimate goal for the learned embedding space is to resolve all conflicting relationships and pull similar images closer while pushing dissimilar images further away. However, visual data is, commonly, not uniformly distributed, but has a complex structure, where different regions of the data distribution have different densities [115]. Data points in different regions of the distribution are often related based on different types of similarity such as shape, color, identity or semantic meaning. While, theoretically, a deep neural network representation is powerful enough to approximate arbitrary continuous functions [113], in practice this often leads to poor local minima and overfitting. This is partially due to an inefficient usage of the embedding space [210, 212] and an attempt to capture all the aforementioned types of visual similarity directly by fitting a single embedding space to all the training data [165, 166, 231].

[1] This chapter is based on joint work [240] with Vadim Tschernezki, Uta Büchler, and Björn Ommer, originally presented at CVPR 2019. References to prior work (such as "existing approaches", "recent methods", or "state-of-the-art methods") should be read with this context in mind.

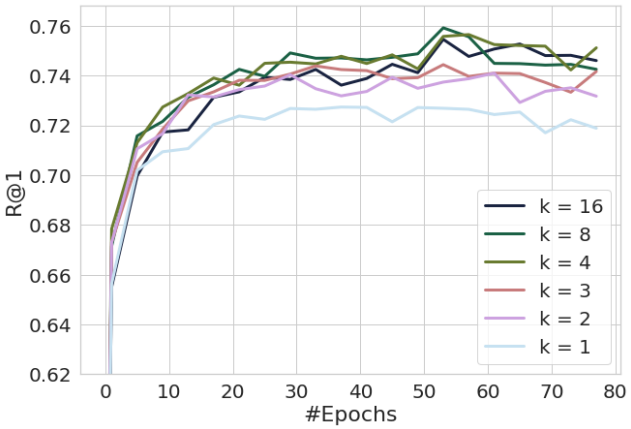


Figure 2.1: Evaluation of different numbers of learners. We train our model with $K = 1, 2, 3, 4, 8$ and 16 learners on the Stanford Online Products dataset [210] and report the change of the Recall@1 score during training. An increase in the number of learners leads to higher Recall@1. The best performance is achieved with $K = 8$.

The problems stated above motivate an approach which will use the embedding space in a more profound way by learning a separate embedding subspace for different regions of the data distribution. We propose a novel deep metric learning approach, inspired by the well-known divide and conquer algorithm. We explicitly split the embedding space and the data distribution into multiple parts given the network representation and learn a separate embedding subspace for each part of the data distribution. Different subspaces are learned on non-overlapping parts of the training data, but all of them share the same feature representation from the previous layer of the Convolutional Neural Network (CNN). The final embedding space is seamlessly composed by concatenating the solutions on each of the non-overlapping subspaces. See Fig. 2.2 for an illustration.

Our approach can be utilized as an efficient drop-in replacement for the final linear layer commonly used for learning embeddings in the existing deep metric learning approaches, regardless of the loss function used for training. We demonstrate a consistent performance boost when applying our approach to the widely-used triplet loss [241] and more complex state-of-the-art metric learning losses such as Proxy-NCA [198] and Margin loss [296]. By using the proposed approach, we achieve new state-of-the-art performance on five benchmark datasets for retrieval, clustering and re-identification: CUB200-2011 [277], CARS196 [148], Stanford Online Products [210], In-shop Clothes [174], and PKU VehicleID [171]. The source code is available on github[2].

[2] https://github.com/CompVis/metric-learning-divide-and-conquer

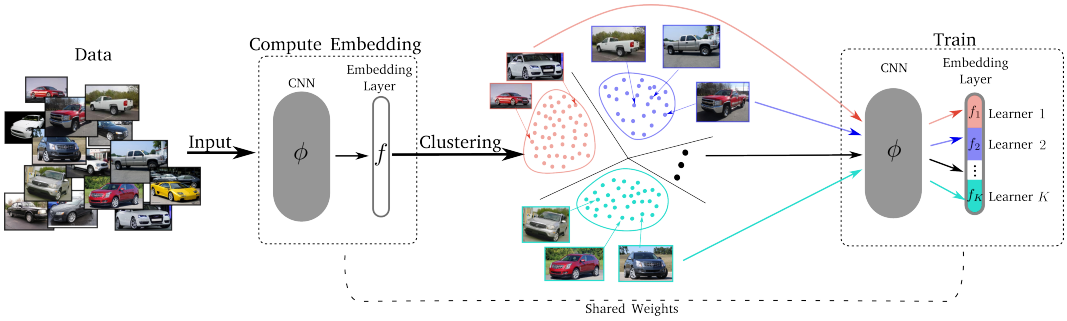


Figure 2.2: Pipeline of our approach. We first cluster the data in the embedding space in K groups and assign a separate subspace (learner) of the embedding layer to every cluster. During training, every learner only sees the samples assigned to the corresponding cluster.

2.1 Related work

Metric learning has been of major interest for the vision community since its early beginnings, due to its broad applications including object retrieval [210, 274, 295], zero-shot and single-shot learning [210, 274], keypoint descriptor learning [248], face verification [41] and clustering [106]. With the advent of CNNs, several approaches have been proposed for supervised Distance Metric Learning. Some methods use pairs [313] or triplets [241, 281] of images. Others use quadruplets [252, 274] or impose constraints on tuples of larger sizes like Lifted Structure [210], n-pairs [252] or poset loss [12].

Using a tuple of images as training samples yields a huge amount of training data. However, only a small portion of the samples among all N^p possible tuples of size p is meaningful and provides a learning signal. A number of recent works tackle the problem of hard and semi-hard negative mining which provides the largest training signal by designing sampling strategies [82, 99, 122, 296, 319]. Existing sampling techniques, however, require either running an expensive, quadratic on the number of data points preprocessing step for the entire dataset and for every epoch [82, 99], or lack global information while having a local view on the data based on a single randomly-drawn mini-batch of images [241, 252, 296]. On the contrary, our approach efficiently alleviates the problem of the abundance of easy samples, since it jointly splits the embedding space and clusters the data using the embedding space learned so far. Hence, samples inside one cluster will have smaller distances to one another than to samples from another cluster, which serves as a proxy to the mining of more meaningful relationships [99, 241]. For further details of our approach see Sec. 2.2.

Recently, a lot of research efforts have been devoted to designing new loss functions [198, 230, 252, 253, 274, 280]. For example, Facility Location [253] optimizes a cluster quality metric, Histogram loss [274] minimizes the overlap between the

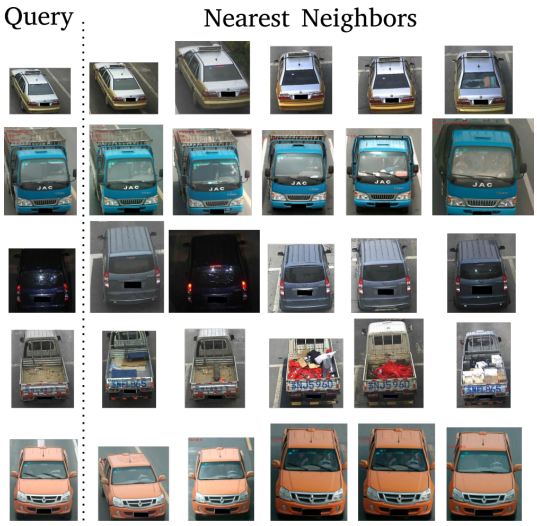


Figure 2.3: Qualitative image retrieval results on PKU VehicleID [171]. We show 5 nearest neighbors per randomly chosen query image given our trained features. The queries and retrieved images are taken from the test set of the dataset.

distribution of positive and negative distances. Kihyuk Sohn proposed in [252] the N-pairs loss which enforces a softmax cross-entropy loss among pairwise similarity values in the batch. The Proxy-NCA loss, presented in [198] computes proxies for the original points in the dataset and optimizes the distances to these proxies using NCA [230]. Our work is orthogonal to these approaches and provides a framework for Distance Metric Learning independent on the choice of a particular loss function.

Another line of work in deep metric learning which is more related to our approach is ensemble learning [76, 94, 212, 311]. Previous works [212, 311] employ a sequence of "learners" with increasing complexity and mine samples of different complexity levels for the next learners using the outputs of the previous learners. HDC [311] uses a cascade of multiple models of a specific architecture, A-BIER [212] applies a gradient boosting learning algorithm to train several learners inside a single network in combination with an adversarial loss [78, 87]. The key difference of the aforementioned approaches to our approach is that we split the embedding space and cluster the data jointly, so each "learner" will be assigned to the specific subspace and corresponding portion of the data. The "learners" are independently trained on non-overlapping chunks of the data which reduces the training complexity of each individual learner, facilitates the learning of decorrelated representations and can be easily parallelized. Moreover, our approach does not introduce extra parameters during training and works in a single model. It does not require any additional loss functions and can be applied to any existing network architecture.

Figure 2.4: Qualitative image retrieval results on the test set of Stanford Online Products [210]. We randomly choose 5 query images and show 5 nearest neighbors per query image given the features of our trained model.

2.2 Approach

The main intuition behind our approach is the following: Solving bigger problems is usually harder than solving a set of smaller ones. We propose an effective and easily adaptive divide and conquer algorithm for deep metric learning. We divide the data into multiple groups (sub-problems) to reduce the complexity and solve the metric learning problem on each sub-problem separately. Since we want the data partitioning to be coupled with the current state of the embedding space, we cluster the data in the embedding space learned so far. Then we split the embedding layer of the network into slices. Each slice of the embedding layer represents an individual learner. Each learner is assigned to one cluster and operates in a certain subspace of the original embedding space. At the conquering stage we merge the solutions of the sub-problems, obtained by the individual learners, to get the final solution. We describe each step of our approach in details in Sec. 2.2.2 and 2.2.3.

2.2.1 Preliminaries

We denote the training set as $X = \{x_1, \dots, x_n\} \subset \mathcal{X}$, where $\mathcal{X}$ is the original RGB space, and the corresponding class labels as $Y = \{y_1, \dots, y_n\}$. A Convolutional Neural Network (CNN) learns a non-linear transformation of the image into an m-dimensional deep feature space $\phi(\cdot;\theta_\phi) : \mathcal{X} \to \mathbb{R}^m$, where θ_ϕ is the set of the CNN parameters. For brevity, we will use the notations $\phi(x_i;\theta_\phi)$ and ϕ_i interchangeably.

To learn a mapping into the embedding space, a linear layer $f(\cdot;\theta_f) : \mathbb{R}^m \to \mathbb{R}^d$ with d neurons is typically appended to the CNN, where θ_f denotes the parameters

Figure 2.5: Qualitative image retrieval results on In-shop clothes [174]. We randomly choose 5 query images from the query set of the In-shop clothes dataset and show 5 nearest neighbors per query image given our trained features. The retrieved images are taken from the gallery set.

of this layer. $f(\cdot;\theta_f)$ is often normalized to have a unit length for training stability [241]. The goal of metric learning is to jointly learn ϕ and f in such a way that $(f \circ \phi)(x;\theta_\phi,\theta_f)$ maps similar images close to one another and dissimilar ones far apart in the embedding space. Formally, we define a distance between two data points in the embedding space as

$$d_f(x_i,x_j) = ||f(\phi_i) - f(\phi_j)||_2. \tag{2.1}$$

For DML, one can use any loss function with options such as [198, 210, 241, 252, 274, 296]. Our framework is independent of the choice of the loss function. In this chapter we experiment with three different losses: Triplet loss [241], Proxy-NCA loss [198] and Margin loss [296]. For simplicity, we will demonstrate our approach in this section on the example of triplet loss, which is defined as

$$l_{\text{triplet}}(a,p,n;\theta_\phi,\theta_f) = \left[d_f(a,p)^2 - d_f(a,n)^2 + \alpha\right]_+, \tag{2.2}$$

where $[\cdot]_+$ denotes the positive part and α is the margin. The triplet loss strives to keep the positive data point p closer to the anchor point a than any other negative point n. For brevity we omit the definitions of other losses, but we refer the interested reader to the original works [198, 296].

Figure 2.6: Qualitative image retrieval results on the test set of CARS196 dataset [148]. We randomly choose 5 query images and show 5 nearest neighbors per query retrieved using our trained features.

2.2.2 Division of the embedding space

We begin with the division stage of our approach. To reduce the complexity of the problem and to utilize the entire embedding space more efficiently we split the embedding dimensions and the data into multiple groups. Each learner will learn a separate subspace of the original embedding space using only a part of the data.

Splitting the data. Let K be the number of sub-problems. We group all data points $\{x_1, \ldots, x_n\}$ according to their pairwise distances in the embedding space into K clusters $\{C_k | 1 \leq k \leq K\}$ with K-means [183].

Splitting the embedding. Next, we define K individual learners within the embedding space by splitting the embedding layer of the network into K consecutive slices. Formally, we decompose the embedding function $f(\cdot;\theta_f)$ into K functions $\{\mathbf{f}_1, \ldots, \mathbf{f}_K\}$, where each $\mathbf{f}_k$ maps the input into the d/K-dimensional subspace of the original d-dimensional embedding space: $\mathbf{f}_k(\cdot;\theta_{\mathbf{f}_k}) : \mathbb{R}^m \rightarrow \mathbb{R}^{d/K}$. $\mathbf{f}_1$ will map into the first d/K dimensions of the original embedding space, $\mathbf{f}_2$ into the second d/K dimensions and so on. Please see Fig. 2.2 for an illustration. Note that the number of the model parameters stays constant after we perform the splitting of the embedding layer, since the learners share the underlying representation.

2.2.3 Conquering stage

In this section, we first describe the step of solving individual problems. Then, we outline the merging step, where the solutions of sub-problems are combined to form the final solution.

Training. After the division stage, every cluster C_k is assigned to a learner $\mathbf{f}_k$, $1 \leq k \leq K$. Since all learners reside within a single linear embedding layer and

share the underlying feature representation, we train them jointly in an alternating manner. In each training iteration only one of the learners is updated. We uniformly sample a cluster $C_k, 1 \leq k \leq K$ and draw a random mini-batch B from it. Then, a learner $\mathbf{f}_k$ minimizes its own loss defined as follows:

$$\mathcal{L}_k^{\theta_\phi, \theta_{\mathbf{f}_k}} = \sum_{(a,p,n)\sim B} \left[d_{\mathbf{f}_k}(a,p)^2 - d_{\mathbf{f}_k}(a,n)^2 + \alpha \right], \tag{2.3}$$

where triplet $(a, p, n) \in B \subset C_k$ denotes the triplets sampled from the current mini-batch, and $d_{\mathbf{f}_k}$ is the distance function defined in the subspace of the k-th learner. As described in Eq. (2.3) each backward pass will update only the parameters of the shared features θ_ϕ and the parameters of the current learner $\theta_{\mathbf{f}_k}$. Motivated by the fact that the learned embedding space is improving during the time, we update the data partitioning by re-clustering every T epochs using the full embedding space. The full embedding space is composed by simply concatenating the embeddings produced by the individual learners.

Merging the solutions. Finally, following the divide and conquer paradigm, after individual learners converge, we merge their solutions to get the full embedding space. Merging is done by joining the embedding layer slices, corresponding to the K learners, back together. After this, we fine-tune the embedding layer on the entire dataset to achieve the consistency between the embeddings of the individual learners. An overview of the full training process of our approach can be found in Algorithm 1.

2.3 Experiments

In this section, we first introduce the datasets we use for evaluating our approach and provide afterwards additional details regarding the training and testing of our framework. We then show qualitative and quantitative results which we compare with the state-of-the-art by measuring the image retrieval quality and clustering performance. The ablation study in subsection 2.3.4 provides then some inside into our metric learning approach.

2.3.1 Datasets

We evaluate the proposed approach by comparing it with the state-of-the-art on two small benchmark datasets (CARS196 [148], CUB200-2011 [277]), and on three large-scale datasets (Stanford Online Products [210], In-shop Clothes [174], and PKU VehicleID [171]). For assessing the clustering performance we utilize the normalized mutual information score [242] $\text{NMI}(\Omega, \mathbb{C}) = \frac{2 \cdot I(\Omega, \mathbb{C})}{H(\Omega) + H(\mathbb{C})}$, where Ω denotes the ground truth clustering and $\mathbb{C}$ the set of clusters obtained by K-means. Here I represents the mutual information and H the entropy. For the retrieval task we report the Recall@k metric [126].

Algorithm 1 Training a model with our approach

Input: X,f,θ_ϕ, θ_f,K,T ▷ data, linear layer, CNN weights, weights of f, # clusters, re-cluster freq.

▷ cluster affiliation $\forall x_i \in X$

$\{\mathbf{f}_1, \dots \mathbf{f}_K\} \leftarrow \text{SplitEmbedding}(f)$ ▷ set of learners

$epoch \leftarrow 0$

while Not Converged **do**

  if $epoch$ mod $T == 0$ **then**

    $f \leftarrow \text{ConcatEmbedding}(\{\mathbf{f}_1, \dots \mathbf{f}_K\})$

    $emb \leftarrow \text{ComputeEmbedding}(X, \theta_\phi, \theta_f)$

    $\{C_1, \dots, C_K\} \leftarrow \text{ClusterData}(emb, K)$

    $\{\mathbf{f}_1, \dots \mathbf{f}_K\} \leftarrow \text{SplitEmbedding}(f)$

  end if

  repeat

    $C_k \sim \{C_1, \dots, C_K\}$ ▷ sample cluster

    $b \leftarrow \text{GetBatch}(C_k)$ ▷ draw mini-batch

    $\mathcal{L}_k \leftarrow \text{FPass}(b, \theta_\phi, \theta_{\mathbf{f}_k})$ ▷ compute loss of learner $\mathbf{f}_k$ (Eq. (2.3))

    $\theta_\phi, \theta_{\mathbf{f}_k} \leftarrow \text{BPass}(L, \theta_\phi, \theta_{\mathbf{f}_k})$ ▷ update weights

  until Epoch completed

  $epoch \leftarrow epoch + 1$

end while

$f \leftarrow \text{ConcatEmbedding}(\{\mathbf{f}_1, \dots \mathbf{f}_K\})$

$\theta_\phi, \theta_f \leftarrow \text{Finetune}(X, \theta_\phi, \theta_f, f)$

Output: θ_ϕ, θ_f

Stanford Online Products [210] is one of the largest publicly available image collections for evaluating metric learning methods. It consists of 120,053 images divided into 22,634 classes of online products, where 11,318 classes (59,551 images) are used for training and 11,316 classes (60,502 images) for testing. We follow the same evaluation protocol as in [210]. We calculate Recall@k score for $k = 1, 10, 100, 1000$ for evaluating the image retrieval quality and the NMI metric for appraising the clustering performance, respectively.

CARS196 [148] contains 196 different types of cars distributed over 16,185 images. The first 98 classes (8,054 images) are used for training and the other 98 classes (8,131 images) for testing. We train and test on the entire images without using bounding box annotations.

CUB200-2011 [277] is an extended version of the CUB200 dataset which consolidates images of 200 different bird species with 11,788 images in total. The first 100 classes (5,864 images) are used for training and the second 100 classes (5,924 images)

R@k	1	10	100	1000	NMI
Histogram [274]	63.9	81.7	92.2	97.7	-
Bin. Deviance [274]	65.5	82.3	92.3	97.6	-
Triplet Semihard [253]	66.7	82.4	91.9	-	89.5
LiftedStruct [210]	63.0	80.5	91.7	97.5	87.4
FacilityLoc [253]	67.0	83.7	93.2	-	-
N-pairs [252]	67.7	83.7	93.0	97.8	88.1
Angular [280]	70.9	85.0	93.5	98.0	88.6
DAML (N-p) [57]	68.4	83.5	92.3	-	89.4
HDC [311]	69.5	84.4	92.8	97.7	-
DVML [169]	70.2	85.2	93.8	-	90.8
BIER [211]	72.7	86.5	94.0	98.0	-
ProxyNCA [198]	73.7	-	-	-	-
A-BIER [212]	74.2	86.9	94	97.8	-
HTL [82]	74.8	88.3	94.8	98.4	-
Margin baseline [296]	72.7	86.2	93.8	98.0	**90.7**
Ours (Margin)	**75.9**	**88.4**	**94.9**	**98.1**	90.2

Table 2.1: Recall@k for $k = 1, 10, 100, 100$ and NMI on Stanford Online Products [210]

for testing. We train and test on the entire images without using bounding box annotations.

In-shop Clothes Retrieval [174] contains 11,735 classes of clothing items with 54,642 images. We follow the evaluation protocol of [174] and use a subset of 7,986 classes with 52,712 images. 3,997 classes are used for training and 3,985 classes for testing. The test set is partitioned into query set and gallery set, containing 14,218 and 12,612 images, respectively.

PKU VehicleID [171] is a large-scale vehicle dataset that contains 221,736 images of 26,267 vehicles captured by surveillance cameras. The training set contains 110,178 images of 13,134 vehicles and the testing set comprises 111,585 images of 13,133 vehicles. We evaluate on 3 test sets of different sizes as defined in [171]. The small test set contains 7,332 images of 800 vehicles, the medium test set contains 12,995 images of 1600 vehicles, and the large test set contains 20,038 images of 2400 vehicles. This dataset has smaller intra-class variation, but it is more challenging than CARS196, because different identities of vehicles are considered as different classes, even if they share the same car model.

2.3.2 Implementation Details

We implement our approach by closely following the implementation of Wu et al. [296] based on ResNet-50 [102]. We use an embedding of size $d = 128$ and an

R@k	1	2	4	8	NMI
Triplet Semihard [253]	51.5	63.8	73.5	82.4	53.4
LiftedStruct [210]	48.3	61.1	71.8	81.1	55.1
FacilityLoc [253]	58.1	70.6	80.3	87.8	59.0
SmartMining [99]	64.7	76.2	84.2	90.2	-
N-pairs [252]	71.1	79.7	86.5	91.6	64.0
Angular [280]	71.4	81.4	87.5	92.1	63.2
ProxyNCA [198]	73.2	82.4	86.4	88.7	64.9
HDC [311]	73.7	83.2	89.5	93.8	-
DAML (N-pairs) [57]	75.1	83.8	89.7	93.5	66.0
HTG [319]	76.5	84.7	90.4	94	-
BIER [211]	78.0	85.8	91.1	95.1	-
HTL [82]	81.4	88.0	92.7	95.7	-
DVML [169]	82.0	88.4	93.3	96.3	67.6
A-BIER [212]	82.0	89.0	93.2	96.1	-
Margin baseline [296]	79.6	86.5	91.9	95.1	69.1
Ours (Margin)	**84.6**	**90.7**	**94.1**	**96.5**	**70.3**
DREML [301]	86.0	91.7	95.0	97.2	76.4

Table 2.2: Recall@k for $k = 1, 2, 4, 8$ and NMI on CARS196 [148]

input image size of 224×224 [102] for all our experiments. The embedding layer is randomly initialized. All models are trained using Adam [145] optimizer with the batch size of 80 for Stanford Online Products and In-shop Clothes datasets, and 128 for the other datasets. We resize the images to 256 and apply random crops and horizontal flips for data augmentation. For training our models we set the number of learners $K = 4$ for CUB200-2011 and CARS196 due to their small size, and $K = 8$ for all the other datasets.

We update the data partitioning by re-clustering every T epochs using the full embedding space, composed by concatenating the embeddings produced by the individual learners. We have noticed that our approach is not sensitive to the values of T in the range between 1 and 10. We set $T = 2$ for all experiment, since the value alteration did not lead to significant changes in the experimental results. To maintain consistency, each learner is associated to the cluster, which is most similar to the cluster assigned to this learner in the previous iteration (i.e. in epoch $t - T$). This amounts to solving a linear assignment problem where similarity between clusters is measured in terms of IoU of points belonging to the clusters.

Similar to [241, 296] we initialize Margin loss with $\beta = 1.2$ and Triplet loss with $\alpha = 0.2$. Mini-batches are sampled following the procedure defined in [241, 296] with $m = 4$ images per class per mini-batch for Margin loss [296] and Triplet loss [241], and uniformly for Proxy-NCA [198]. During the clustering (Sec. 2.2.2) and

R@k	1	2	4	8	NMI
LiftedStruct [210]	46.6	58.1	69.8	80.2	56.2
FacilityLoc [253]	48.2	61.4	71.8	81.9	59.2
SmartMining [99]	49.8	62.3	74.1	83.3	-
Bin. Deviance [274]	52.8	64.4	74.7	83.9	-
N-pairs [252]	51.0	63.3	74.3	83.2	60.4
DVML [169]	52.7	65.1	75.5	84.3	61.4
DAML (N-pairs) [57]	52.7	65.4	75.5	84.3	61.3
Histogram [274]	50.3	61.9	72.6	82.4	-
Angular [280]	54.7	66.3	76.0	83.9	61.1
HDC [311]	53.6	65.7	77.0	85.6	-
BIER [211]	55.3	67.2	76.9	85.1	-
HTL [82]	57.1	68.8	78.7	86.5	-
A-BIER [212]	57.5	68.7	78.3	86.2	-
HTG [319]	59.5	71.8	81.3	88.2	-
Triplet Semihard [253]	42.6	55.0	66.4	77.2	55.4
Triplet Semihard baseline*	53.1	65.9	76.8	85.3	60.3
Ours (Triplet Semihard)	55.4	66.9	77.5	86.5	61.9
ProxyNCA [198]	49.2	61.9	67.9	72.4	64.9
ProxyNCA baseline*	58.7	70.0	79.1	87.0	62.5
Ours (ProxyNCA)	61.8	73.1	81.8	88.2	65.7
Margin baseline [296]	63.6	74.4	83.1	90.0	69.0
Ours (Margin)	**65.9**	**76.6**	**84.4**	**90.6**	**69.6**
DREML [301]	63.9	75.0	83.1	89.7	67.8

Table 2.3: Recall@k for $k = 1, 2, 4, 6, 8$ and NMI on CUB200-2011 [277]. * denotes our own implementation based on ResNet-50 with $d = 128$.

test phase, an image embedding is composed by concatenating the embeddings of individual learners.

The source code is available at GitHub.

2.3.3 Results

We now compare our approach to the state-of-the-art. From Tables 2.1, 2.2, 2.3, 2.4 and 2.5 we can see that our method with Margin loss [296] outperforms existing state-of-the-art methods on all 5 datasets, proving its wide applicability. Note that we use a smaller embedding size of $d = 128$ instead of 512 employed by runner-up approaches HTL [82], A-BIER [212], BIER [211], DVML [169], DAML [57], and Angular loss [280]; HDC [311] uses a 384-dimensional embedding layer. Moreover, we compare our results to the deep ensembles approach DREML [301],

R@k	1	10	20	30	50	NMI
FashionNet [174]	53.0	73.0	76.0	77.0	80.0	-
HDC [311]	62.1	84.9	89.0	91.2	93.1	-
BIER [211]	76.9	92.8	95.2	96.2	97.1	-
HTG [319]	80.3	93.9	95.8	96.6	97.1	-
HTL [82]	80.9	94.3	95.8	97.2	97.8	-
A-BIER [212]	83.1	95.1	96.9	97.5	98.0	-
Margin baseline* [296]	82.6	94.8	96.2	97.0	97.7	87.8
Ours (margin)	**85.7**	**95.5**	**96.9**	**97.5**	**98.0**	**88.6**
DREML [301]	78.4	93.7	95.8	96.7	-	-

Table 2.4: Recall@k for $k = 1, 10, 20, 30, 50$ and NMI on In-shop Clothes [174]. * denotes our own implementation based on ResNet-50 with $d = 128$.

which trains an ensemble of 48 ResNet-18 [102] networks with a total number of 537M trainable parameters. Our model has only 25.5M trainable parameters and still outperforms DREML [301] on CUB200-2011 and In-shop Clothes datasets by a large margin.

We demonstrate the results of our approach with three different losses on CUB200-2011: Triplet [241], Proxy-NCA [198] and Margin loss [296]. Our approach improves the Recall@1 performance by at least 2.1% in each of the experiments (see Tab. 2.3). This confirms that our approach is universal and can be applied to a variety of metric learning loss functions. We noticed that it shows especially large improvements on large-scale datasets such as on PKU VehicleID, where we improve by 3.6% over the baseline with Margin loss [296] and surpass the state-of-the-art by 1% in terms of Recall@1 score on the large test set. We attribute this success on such a challenging dataset to the more efficient exploitation of large amounts of data due to dividing it between different learners which operate on non-overlapping subspaces of the entire embedding space.

In addition to the quantitative results, we show in Figure 2.3,2.4,2.5 and 2.6 qualitative image retrieval results on CUB200-2011, Stanford Online Products, In-shop clothes, and Cars196. Note that our model is invariant to viewpoint and daylight changes.

2.3.4 Ablation Study

We perform several ablation experiments to demonstrate the effectiveness of the proposed method and evaluate the different components of our contribution. We use the Stanford Online Products dataset and train all models with Margin loss [296] for 80 epochs.

First, we analyze the choice of the number of learners K. As can be seen in Fig. 2.1, Recall@1 significantly increased already with $K = 2$. The best result is

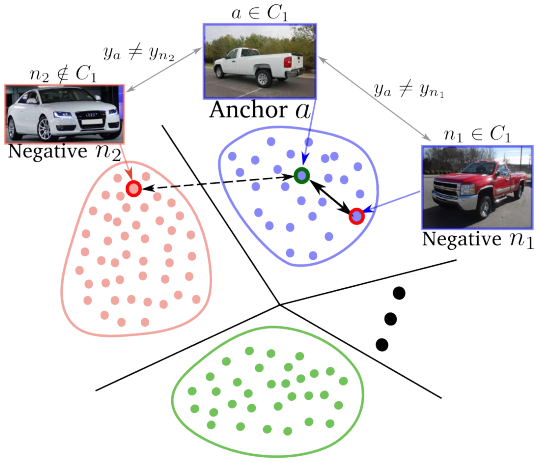


Figure 2.7: Natural hard negative mining. During training, we only sample tuples (e.g., pairs or triplets) from the same cluster. The expected value of the distance between a negative sample and an anchor within a cluster is lower than the expected value when the data points belong to different clusters. Our approach naturally finds hard negative samples without explicitly performing a hard negative mining procedure.

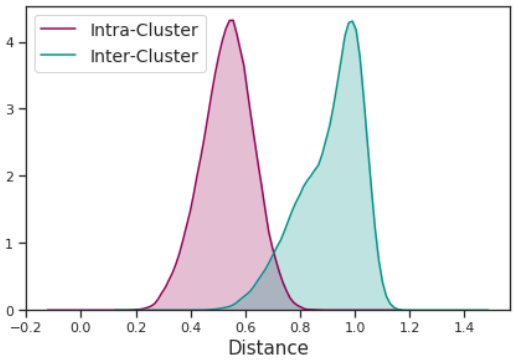


Figure 2.8: Intra-cluster and inter-cluster distributions of distances for negative pairs. Red histogram shows the distribution of the pairwise distances of samples having different class labels but from the same cluster (intra-cluster); green histogram shows the distribution of the pairwise distances of samples having different class labels and drawn from different clusters (inter-cluster). Negative pairs within one cluster have lower distances and are harder on average.

achieved with $K = 8$, where each learner operates in a 16-dimensional embedding subspace. Increasing the number of learners from $K > 1$ on, results in faster convergence and better local optima.

Split Size →	Small		Medium		Large	
R@k	1	5	1	5	1	5
Mixed Diff+CCL [171]	49.0	73.5	42.8	66.8	38.2	61.6
GS-TRS loss [63]	75.0	83.0	74.1	82.6	73.2	81.9
BIER [211]	82.6	90.6	79.3	88.3	76.0	86.4
A-BIER [212]	86.3	92.7	83.3	88.7	81.9	88.7
Margin baseline* [296]	85.1	91.4	82.9	88.9	79.2	88.4
Ours (margin)	**87.7**	**92.9**	**85.7**	**90.4**	**82.9**	**90.2**
DREML [301]	88.5	94.8	87.2	94.2	83.1	92.4

Table 2.5: Recall@k for $k = 1, 5$ on the small, medium and large PKU VehicleID [171] dataset. * denotes our own implementation based on ResNet-50 with $d = 128$.

Next, we study the effect of clustering the data. In Tab. 2.6 we see that substituting K-means clustering in the embedding space with random data partitioning significantly degrades the performance. On the other hand, what happens if we use K-means clustering in the embedding space, but do not split the embedding f into K subspaces $\mathbf{f}_1, \ldots, \mathbf{f}_K$ during training? I.e., we perform regular training but with sampling from clusters. From Tab. 2.6 we see that it leads to a performance drop compared to the proposed approach, however it is already better than the baseline. This is due to the fact that drawing mini-batches from the clusters yields harder training samples compared to drawing mini-batches from the entire dataset. The expectation of the distance between a negative pair within the cluster is lower than the expectation of the distance between a negative pair randomly sampled from the entire dataset, as visually depicted on Fig. 2.7 and Fig. 2.8. This shows that: a) sampling from clusters provides a stronger learning signal than regular sampling from the entire dataset, b) to be able to efficiently learn from harder samples we need an individual learner for each cluster, which significantly reduces the complexity of the metric learning task. We also substitute K-means clustering with the fixed data partitioning, based on the ground truth labels, which are manually grouped according to semantic similarity (see "GT labels grouping" in Tab. 2.6). We recognize that the use of a flexible clustering scheme, which depends on the data distribution in the embedding space, leads to better performance than using class labels.

Runtime complexity. Splitting the embedding space into subspaces and training K independent learners reduces the time required for a single forward and backward pass, since we only use a d/K-dimensional embedding instead of the full embedding. We perform K-means clustering every T epochs. We use the K-means implementation from the Faiss library [129] which has an average complexity of $O(Kni)$, where n is the number of samples, and i is the number of iterations. This adds a neglectable overhead compared to the time required for a full forward and

R@k	1	10	100	1000
Baseline [296]	72.7	86.2	93.8	98.0
K-means in the embedding space, no embedding splitting	75.0	87.6	94.2	97.8
Random data partition	73.2	85.8	93.4	97.6
GT labels grouping	74.5	87.1	93.8	97.6
K-means in the embedding space	**75.9**	**88.4**	**94.9**	**98.1**

Table 2.6: Evaluation of different data grouping methods on Stanford Online Products [210] with $K = 8$ and Margin loss [296].

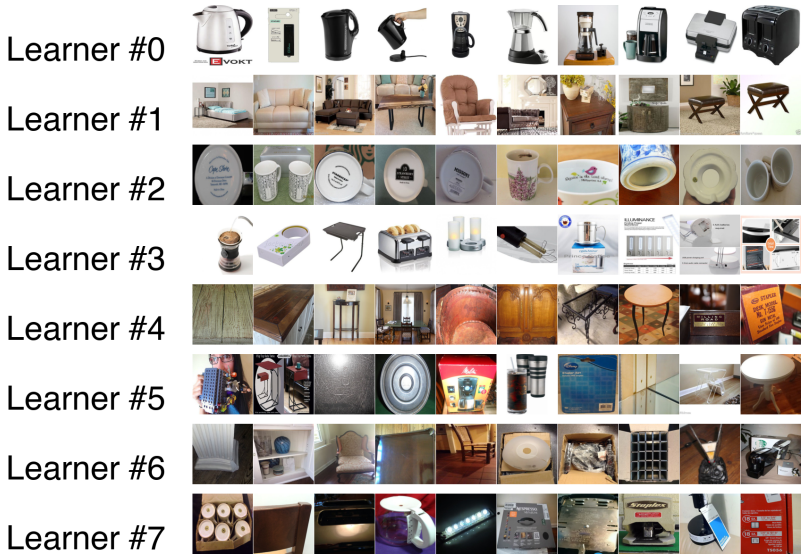


Figure 2.9: Representative images for the individual learners and their corresponding subspaces. The model was trained on the Stanford Online Products dataset with $K = 8$. Best viewed zoomed in.

backward pass of all images in the dataset. For example, in case of $T = 2$ the clustering will add $\approx 25\%$ overhead and in case of $T = 8$ only $\approx 6.25\%$.

Ablation study of individual learners

Our approach facilitates the learning of decorrelated representations of individual learners. To show this, we conduct an additional ablation study where we evaluate the performance of individual learners and compute the correlation between their embeddings. Here we use the Stanford Online Products dataset [210] and train our model with Margin loss [296], $K = 8$ and embedding size $d = 128$.

	Baseline	Ours	Embedding dimensions
Learner 1	37.0	29.6	1..16
Learner 2	37.0	29.7	17..32
Learner 3	36.5	29.5	33..48
Learner 4	36.5	29.4	49..64
Learner 5	36.3	29.1	65..80
Learner 6	37.4	29.7	81..96
Learner 7	36.7	29.4	97..112
Learner 8	37.1	29.9	113..128
Concatenation of all (↑)	72.7	**75.9**	1..128
Correlation coeff. (↓)	0.0602	**0.0498**	-

Table 2.7: Evaluation of the individual learners. We calculated Recall@1 for every individual learner on the entire test set of Stanford Online Products [210]. The last column shows the indices of the corresponding dimensions of the embedding space assigned to the learners. The individual learners of our model yield significantly higher Recall@1 than the baseline model when they are concatenated and evaluated all together ("Concatenation of all"), since they learn less correlated representations.

We computed Recall@1 on the entire test set for every individual learner, each of which operates in a 16-dimensional embedding subspace. However, the baseline model was trained with only *one* learner operating in the embedding space with 128 dimensions. Hence, for comparison with the learners of our model, we split the embedding of the baseline model on 8 non-overlapping slices of 16 dimensions each and evaluate them separately. In Tab. 2.7 we can see that each individual learner trained using our approach is weaker in average than slices of the baseline model embedding. However, when we concatenate the embeddings of all individual learners together they yield significantly higher Recall@1 than the baseline model (3.2% higher in absolute values). In Fig. 2.10 we also show how the performance changes when we use together only $1, 2, \ldots 7$ or all 8 learners for evaluation: one learner corresponds to 16 out of 128 dimensions, two learners to 32 out of 128 dimensions and so on; 8 learners correspond to all 128 dimensions. We observe a larger gain compared to the baseline when more learners are used together for evaluation. This shows that the learners trained by our approach learn complementary features.

Moreover, in Tab. 2.7 we directly computed the correlation coefficient between the embedding produced by different learners. The correlation coefficient between the learners in our model is lower than between the slices of the baseline model embedding. This evidence supports our claim that the learners proposed by our approach learn less correlated features and, hence, utilize the embedding space in a more efficient way.

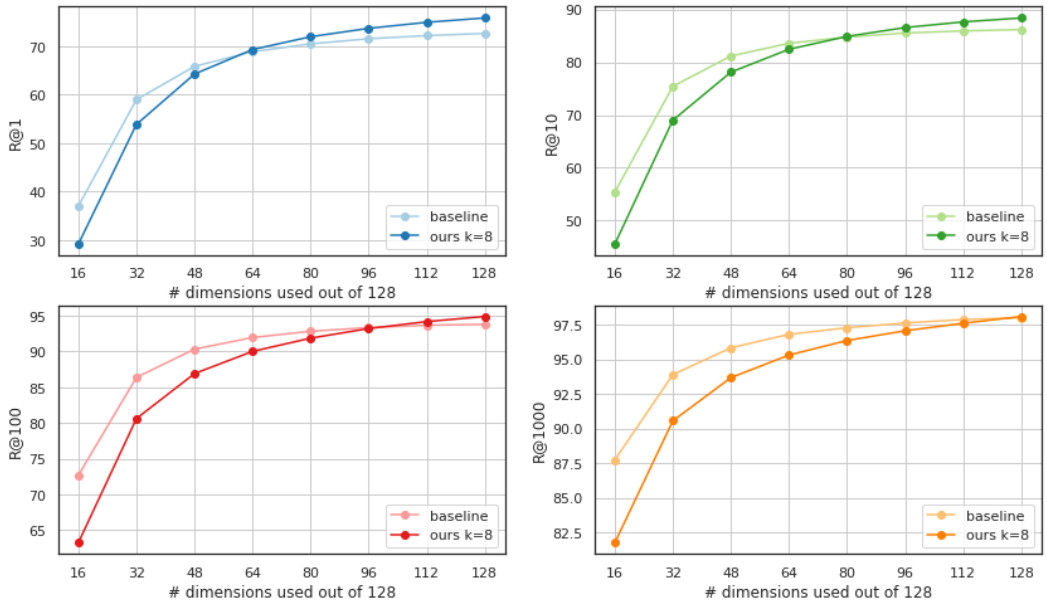


Figure 2.10: Evaluation of the individual learners. We trained our model with $K = 8$ learners and embedding size $d = 128$ on the Stanford Online Products dataset [210]. The plots show the the Recall@k score when we use only the first m out of 128 dimensions of the embedding layer ($m = \{16, 32, \ldots, 128\}$) for evaluation. Adding another 16 dimensions corresponds to using one more learner $\mathbf{f}_{m/16}$ during the evaluation of our model. In case of the baseline model we do not have any learners, but for a fair comparison we also use only the first m dimensions of the embedding layer. We see a higher performance of our approach compared to the baseline when more dimensions are used together, which shows that the individual learners in our model produce less correlated embeddings.

To visualize what is captured in each embedding subspace, in Fig. 2.9 we show representative images for different learners. Every row shows 10 query images, which are the easiest in terms of recall for one learner (R@1 = 1) but extremely difficult (R@30 = 0) for any other learner. We can see that every subspace has its own abstract "specialization". The 1st focuses on the electrical appliances, the 2nd – on furniture, the 3rd – on plates and mugs, etc.

2.4 Conclusion

In this chapter, we introduced a simple and efficient divide and conquer approach for Deep Metric Learning, which divides the data in K clusters and assigns them to individual learners, constructed by splitting the network embedding layer into K non-overlapping slices. We described the procedure for joint training of multiple

learners within one neural network and for combining partial solutions into the final representation. The proposed approach is easy to implement and can be used as an efficient drop-in replacement for a linear embedding layer commonly used in the existing Deep Metric Learning approaches independent on the choice of the loss function. The experimental results on CUB200-2011 [277], CARS196 [148], Stanford Online Products [210], In-shop Clothes [174], and PKU VehicleID [171] show that our approach significantly outperforms the state-of-the-art on all the datasets.

3 Unsupervised Representation Learning using Surrogate Classification[1]

As was shown in Chapter 2, one of the properties of learned visual representations is that they induce similarity (or distance) metric in the visual domain, enabling comparisons between images. Such similarities in the visual domain play a central role for numerous computer vision tasks which range across different levels of abstraction, from low-level image processing to high-level object recognition or human pose estimation. In this chapter, we go beyond supervised learning of representations studied in Chapter 2 and present an unsupervised approach. To design such an approach, we will primarily rely on (dis-)similarities between images. Since the visual representations induce a similarity measure between images, we will use terms *similarity learning* and *representaion learning* interchangeably throughout this thesis.

Similarities have been usually obtained as a result of representations learned for category-level recognition tasks, where the samples are attributed to discrete categories. However, the large intra-class variability of visual categories has recently spurred exemplar methods [103, 187], which split the category-level model into simpler sub-tasks for each sample. Therefore, separate exemplar classifiers are trained by learning the similarities of individual exemplars against a large set of negatives. This paradigm of exemplar learning has been applied with successful results in problems like object recognition [61, 187], instance retrieval [50, 231], and grouping [97]. Learning visual similarities has been also of particular importance for posture analysis [73] and video parsing [216], where exploiting both the appearance [51] and the temporal domain [285] has proven useful.

Throughout the numerous methods for learning visual similarities, supervised techniques have been of particular interest in the computer vision field. These supervised techniques have therefore followed different formulations either as ranking [298], regression [62], and classification [216] problems. Furthermore, with the recent advent of Convolutional Neural Networks (CNNs), two stream architectures [313] and ranking losses [281] have shown great improvements over similarities based on hand-crafted features. Nevertheless, these performance improvements

[1] This chapter is based on joint work [236] with Miguel A. Bautista and Björn Ommer, originally published in Pattern Recognition 78 (2018), which is an extension of our NeurIPS 2016 paper [11]. References to prior work (such as "existing approaches", "recent methods", or "state-of-the-art methods") should be read with this context in mind.

obtained by CNNs come at the cost of requiring millions of samples of supervised training data or at least the fine-tuning [51] on large labeled datasets such as Pascal VOC [67]. Even though the amount of accessible image data is growing at an ever increasing rate, supervised labeling of image similarities is extremely costly. In addition to the difficulty of labeling a similarity metric, not only similarities between images are important, but also between objects and their parts. Annotating the fine-grained similarities between all these entities is hopelessly complex, in particular for the large datasets typically used for training CNNs.

Unsupervised deep learning of similarities that does not require any labels for pre-training or fine-tuning is, therefore, of great interest to the vision community. This way we can utilize large image datasets without being limited by the need for costly manual annotations. However, CNNs for exemplar-based learning have been rare [55] due to limitations resulting from the widely used cross-entropy loss. The learning task of Dosovitskiy et al. [55] suffers from only a single positive instance, it is highly unbalanced with many more negatives, and the relationships between samples are unknown, cf. Sec. 3.2. Consequentially, Stochastic gradient descent (SGD) gets corrupted and has a bias towards negatives, thus forfeiting the benefits of deep learning.

Our approach overcomes these limitations by formulating similarity learning as an exemplar grouping and a surrogate classification problem using CNNs. Typically, at the beginning, only a few local estimates of (dis-)similarities are easily available (i.e. pairs of samples that are highly similar (near duplicates) or that are very distant). Most of the initial similarities are, however, unknown or non-transitive, i.e. mutually contradicting. To nevertheless define balanced classification tasks suited for CNN training, we formulate an optimization problem that builds training batches for the CNN by selecting groups of compact cliques so that all cliques in a batch are mutually distant. Thus for all samples of a batch (dis-)similarity is defined—they either belong to the same compact clique or are far away and belong to different cliques. However, pairs of samples with no reliable similarities end up in different batches so they do not yield false training signal for SGD. Classifying if a sample belongs to a clique serves as a pretext task for learning exemplar similarity. Training the network then implicitly reconciles the transitivity relations between samples in different batches. Thus, the learned CNN representations impute similarities that were initially unavailable and generalize them to unseen data. Furthermore, to incorporate temporal context in our model, we introduce a Local Temporal Pooling (LTP) strategy that models how similarities between exemplars change over short periods of time.

In the experimental evaluation, the proposed approach significantly improves over state-of-the-art approaches for posture analysis and retrieval by learning a general feature representation for a human pose that can be transferred across datasets.

3.1 Related Work

The Exemplar Support Vector Machine (Exemplar-SVM) has been one of the driving methods for exemplar-based learning [187]. Each Exemplar-SVM classifier is defined by a single positive instance and a large set of negatives. To improve performance, Exemplar-SVMs require several rounds of hard negative mining, increasing greatly the computational cost of this approach. To circumvent this high computational cost, Hariharan et al. [97] proposes to train Linear Discriminant Analysis (LDA) over Histogram of Oriented Gradients (HOG) features [97]. LDA whitened HOG features with the common covariance matrix estimated for all the exemplars removes correlations between the HOG features, which tend to amplify the background of the image.

Recently, several CNN approaches have been proposed for supervised similarity learning using either pairs [313], or triplets [281] of images. However, supervised formulations for learning similarities require that the supervisory information scales quadratically for pairs of images, or cubically for triplets. This results in very large training times.

The literature on exemplar-based learning in CNNs is very scarce. In [55] the authors of Exemplar-CNN tackle the problem of unsupervised feature learning. A patch-based categorization problem is designed by randomly extracting patch for each image in the training set and defining it as a surrogate class. Hence, since this approach does not take into account (dis-)similarities between exemplars, it fails to model their transitivity relationships, resulting in poor performances (see Sect. 3.3.1).

Furthermore, recent works [285], [51], [118] and [196] showed that temporal information in videos and spatial context information in images can be utilized as a convenient supervisory signal for learning feature representation with CNNs. However, the computational cost of the training algorithm is enormous since the approach in [51] needs to tackle all possible pair-wise image relationships requiring a training set that scales quadratically with the number of samples. On [118] authors leverage time-contrastive loss to learn representations leveraging the temporal structure of the data. However, this approach is limited to video sequences without repetitions since the method is based on the assumption of mutual independence of time segments. In contrast, our approach leverages the relationship information between compact cliques, framing similarity learning as a multi-class classification problem. As each training batch contains mutually distinct cliques the computational cost of the training algorithm is greatly decreased.

3.2 Methodology

In this section we show how a CNN can be employed for learning similarities between all pairs of a large number of exemplars. In particular, the idiosyncrasies of exemplar learning have made it difficult to unravel its full capabilities in CNNs.

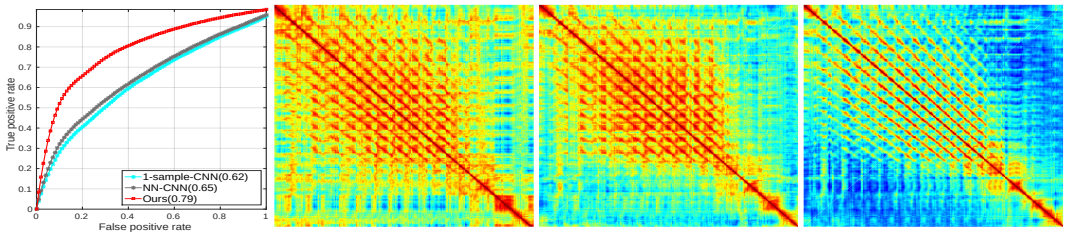


Figure 3.1: (a) Average ROC AUC for posture retrieval in the Olympic Sports dataset. Similarities learned by (b) 1-sample CNN, (c) using NN-CNN, and (d) for the proposed approach. The plots show a magnified crop of the full similarity matrix. Note the more detailed fine structure in (d).

First, deep learning is extremely data hungry, which conflicts with having a single positive exemplar for training, we now abbreviate this setup as 1-sample CNN. This 1-sample setup then faces several issues. *(i)* The within-class variance of an individual exemplar cannot be modeled. *(ii)* The ratio of one exemplar and many negatives is highly imbalanced so that the cross-entropy loss over SGD batches overfits against the negatives. *(iii)* An SGD batch for training a CNN on multiple exemplars can contain arbitrarily similar samples with different label (the different exemplars may be similar or dissimilar), resulting in label inconsistencies. *(iv)* Provided the single training sample, exemplar learning cannot exploit the temporal context of training data, if available.

The methodology proposed in this chapter overcomes this issues as follows. In Sect. 3.2.2 we discuss why simply appending an exemplar with its nearest neighbors and data augmentation (similar in spirit to the Clustered Exemplar-SVM [243], which we abbreviate as NN-CNN) is not sufficient to address *(i)*. Sect. 3.2.3 deals with *(ii)* and *(iii)* by generating batches of cliques that maximize the intra-clique similarity while minimizing inter-clique similarity. In addition, Sect. 3.2.5 shows how to exploit temporal information to further impose structure on the learned similarities by using a temporal average pooling.

To show the effectiveness of the proposed method we give empirical proof by training CNNs following both 1-sample CNN and NN-CNN training protocols. Fig. 3.1(a) shows the average Receiver Operating Characteristic (ROC) curve for posture retrieval in the Olympic Sports dataset [204] (refer to Sec. 3.3.1 for further details) for 1-sample CNN, NN-CNN and the proposed method, which clearly outperforms both exemplar based strategies. In addition, Fig. 3.1(b-d) show an excerpt of the similarity matrix learned for each method. It becomes evident that the proposed approach captures more detailed similarity structures, e.g., the diagonal structures correspond to repetitions of the same gait cycle within a long jump.

3.2.1 Initialization

In the previous section we have shown the shortcomings of exemplar-based training of CNNs. The key obstacle is the discrepancy between the single positive sample used in exemplar learning and the large amounts of data needed to train deep CNNs. Therefore, given a single exemplar x_i we attempt to find an initial number of related samples to enable the training of a CNN which further improves the similarities between exemplars. To obtain this initial group of related samples we employ LDA whitened HOG [97], which is a fundamental and computationally efficient approach to estimate similarities s_{ij} between large numbers of samples. Moreover, since they constitute a view-based approach, HOG features are viewpoint and rotation variant, which is therefore beneficial for pose estimation in 2D. We define $s_{ij} = s(\phi(x_i), \phi(x_j)) = \phi(x_i)^\top \phi(x_j)$, where $\phi(x_i)$ is the whitened HOG descriptor of the exemplar and $\mathbf{S} = (s_{ij}) \in \mathbb{R}^{N \times N}$ is the resulting kernel matrix. The nearest neighbor of the sample i is the sample j which maximizes s_{ij}.

As can be seen from Fig. 3.4(b) most of these similarities are evidently unreliable and, thus, the majority of samples cannot be properly ranked w.r.t. their similarity to an exemplar x_i. However, the most similar and most dissimilar samples can be reliably identified as they are sticking out from the similarity distribution. We can thus utilize these samples to find a small set of nearest neighbors to the exemplar and a set of samples that are dissimilar.

3.2.2 Compact Cliques

Given an exemplar x_i, assigning the same label to its nearest neighbors (positive group) and another label to its furthest neighbors (negative group) is not suitable for learning similarities. The exemplars in these groups may be close to x_i (or distant for the negative group) but not to another due to lacking transitivity. Furthermore, simple synthetic augmentation of either the positive or negative groups [55] does not add transitivity relations to other exemplars. As a result, to learn intra-class similarities we need to restrict the model to groups of samples which are compact and mutually similar to another (i.e. a clique), where all samples in the clique are worthy of having the same label assigned.

To build candidate cliques we apply complete-linkage clustering to merge a x_i with its local neighborhood so that all these samples are mutually similar. Therefore, we start at each x_i and merge the sample with its local neighborhood, so that all merged samples are mutually similar. Thus, cliques are compact, differ in size, and may be mutually overlapping. To reduce redundancy, highly overlapping cliques are subsequently merged by clustering cliques using farthest-neighbor clustering. This agglomerative grouping is terminated if the intra-clique similarity of a cluster is less than half that of its constituents.

Let K be the resulting number of compact cliques and N the number of samples x_i. Then $\mathbf{C} \in \{0,1\}^{K \times N}$ is the resulting assignment matrix of samples to cliques.

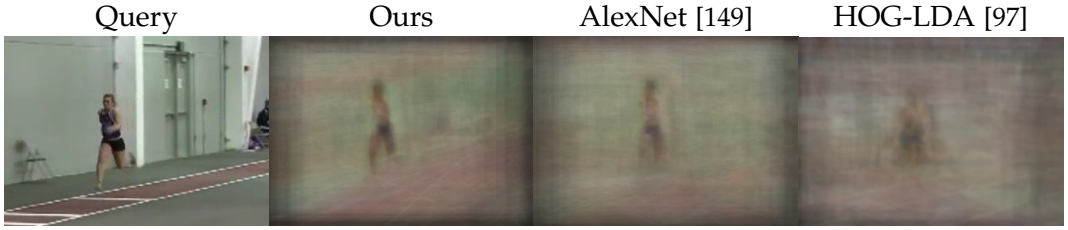


Figure 3.2: Averaging of the 50 nearest neighbors for a given query frame using similarities obtained by our approach, AlexNet [149] pretrained on ImageNet [48], and HOG-LDA [97].

3.2.3 Selecting Mutually Consistent Cliques

After generating a set of compact cliques we assign a unique surrogate (i.e. artificial) label to each clique. Which means that all the samples belonging to the same clique get the same surrogate label. However, since only the highest and lowest similarities are reliable, samples in different cliques are not necessarily dissimilar, even if they get assigned a different surrogate label (e.g. cliques can partially overlap). This issue implies that the surrogate labeling is not consistent since samples with different surrogate labels can be highly similar. Motivated by this observation and leveraging the fact that CNNs are trained on batches of samples, we strive to find batches of mutually distant cliques to compose our batches. Thus, all samples in a batch can be labeled consistently because they are either similar (same compact clique) or dissimilar (different, distant clique). Samples with unreliable similarity then end up in different batches and we train a CNN successively on these batches.

To find a set of different batches of mutually distant cliques we now design an optimization problem that produces a set of consistent batches of cliques. Let $\mathbf{Z} \in \{0,1\}^{B\times K}$ be an indicator matrix that assigns K cliques to B batches (row $\mathbf{z}_b$ of $\mathbf{Z}$ indicates the cliques in batch b) and $\mathbf{S}' \in \mathbb{R}^{K\times K}$ be the similarity between cliques (computed as the average pairwise sample similarity). We enforce cliques in the same batch to be dissimilar by minimizing $\mathrm{tr}\,(\mathbf{Z}\mathbf{S}'\mathbf{Z}^\top)$. Essentially, we seek a selection of cliques that minimize the sum of pairwise similarities between cliques for each batch b, integrated over all batches. To remove the penalty for selecting compact cliques (i.e. with high self-similarity) we subtract $\mathrm{tr}\,(\mathbf{Z}\,\mathrm{diag}\,(\mathbf{S}')\mathbf{Z}^\top)$, which defines the sum of similarities of cliques to themselves, integrated over all batches. Moreover, each batch should maximize sample coverage, i.e., the number of distinct samples in all cliques of a batch $\|\mathbf{z}_b\mathbf{C}\|_p^p$ should be maximal. Finally, the number of distinct points covered by all batches, $\|\mathbb{1}\mathbf{Z}\mathbf{C}\|_p^p$, should be maximal, so that the different (potentially overlapping) batches together comprise as many samples as

possible. We select $p = 1/16$ so that our penalty function roughly approximates the non-linear step function. The objective of the optimization problem then becomes

$$\min_{\mathbf{Z}\in\{0,1\}^{B\times K}} \operatorname{tr}(\mathbf{Z}\mathbf{S}'\mathbf{Z}^\top) - \operatorname{tr}(\mathbf{Z}\operatorname{diag}(\mathbf{S}')\mathbf{Z}^\top) - \lambda_1 \sum_{b=1}^{B} \|\mathbf{z}_b\mathbf{C}\|_p^p - \lambda_2\|\mathbb{1}\mathbf{Z}\mathbf{C}\|_p^p, \tag{3.1}$$

$$\text{s.t.} \qquad \mathbf{Z}\mathbb{1}_K^\top = r\mathbb{1}_B^\top, \tag{3.2}$$

where r is the desired number of cliques in one batch for CNN training. The number of batches, B, can be set arbitrarily high to allow for as many rounds of SGD training as desired. If it is too low, this can be easily spotted as only limited coverage of training data can be achieved in the last term of Eq. (3.1). Since $\mathbf{Z}$ is discrete, the optimization problem (3.1) is not easier than the Quadratic Assignment Problem which is known to be NP-hard [23]. To overcome this issue we relax the binary constraints and force instead the continuous solution to the boundaries of the feasible range by maximizing the additional term $\lambda_3\|\mathbf{Z} - 0.5\|_F^2$ using the Frobenius norm.

We condition $\mathbf{S}'$ to be positive semi-definite by thresholding its eigenvectors and projecting onto the resulting base. Since also $p < 1$ the previous objective function is a difference of convex functions $u(\mathbf{Z}) - v(\mathbf{Z})$, where

$$u(\mathbf{Z}) = \operatorname{tr}(\mathbf{Z}\mathbf{S}'\mathbf{Z}^\top) - \lambda_1 \sum_{b=1}^{B} \|\mathbf{z}_b\mathbf{C}\|_p^p - \lambda_2\|\mathbb{1}\mathbf{Z}\mathbf{C}\|_p^p \tag{3.3}$$

$$v(\mathbf{Z}) = \operatorname{tr}(\mathbf{Z}\operatorname{diag}(\mathbf{S}')\mathbf{Z}^\top) + \lambda_3\|\mathbf{Z} - 0.5\|_F^2 \tag{3.4}$$

It can be solved using the CCCP Algorithm [312]. In each iteration of CCCP, the following convex optimization problem is solved,

$$\operatorname*{arg\,min}_{\mathbf{Z}\in[0,1]^{B\times K}} u(\mathbf{Z}) - \operatorname{vec}(\mathbf{Z})^\top \operatorname{vec}(\nabla v(\mathbf{Z}^t)), \tag{3.5}$$

$$\text{s.t.} \qquad \mathbf{Z}\mathbb{1}_K^\top = r\mathbb{1}_B^\top \tag{3.6}$$

where $\nabla v(\mathbf{Z}^t) = 2\mathbf{Z} \odot (\mathbb{1}\operatorname{diag}(\mathbf{S}')) + 2\mathbf{Z} - \mathbb{1}$ and $\odot$ denotes the Hadamard product. We solve this constrained optimization problem by means of the interior-point method. Fig. 3.3 shows a visual example of a selected batch of cliques.

Let us now analyze the contribution of each term in Eq. (3.1). We observed a significant drop in performance (by more than 30%) when we omitted any of the terms in Eq. (3.1) because of the following reasons: *(i)* omitting term $\left(\operatorname{tr}(\mathbf{Z}\mathbf{S}'\mathbf{Z}^\top) - \operatorname{tr}(\mathbf{Z}\operatorname{diag}(\mathbf{S}')\mathbf{Z}^\top)\right)$ will allow batches to have arbitrarily similar cliques. Thus semantically very similar samples can occur in the same batch but with different labels; *(ii)* omitting term $\sum_{b=1}^{B} \|\mathbf{z}_b\mathbf{C}\|_p^p$ will allow a trivial solution – each batch will degenerate to a single clique containing only a single sample;

(iii) omitting term $\|\mathbb{1}\mathbf{ZC}\|_p^p$ will yield B identical batches, which contain the most dissimilar cliques.

3.2.4 CNN Training

We successively train a CNN on the different batches $\mathbf{z}_b$ obtained by solving the minimization problem in Eq. (3.1). In each batch, classifying samples according to the clique they are in then serves as a pretext task for learning sample similarities.

One of the key properties of CNNs is the training using SGD and backpropagation [152]. The backpropagated gradient is estimated only over a subset (batch) of training samples, so it depends only on the subset of cliques in $\mathbf{z}_b$. Following this observation, the clique categorization problem is effectively decoupled into a set of smaller sub-tasks (i.e. the individual batches of cliques). During training, we randomly pick a batch $\mathbf{z}_b$ in each iteration and compute the stochastic gradient, using the loss $L(\mathbf{W})$,

$$L(\mathbf{W}) \approx \frac{1}{M} \sum_{j \in x^b} f_{\mathbf{W}}(x_j) + \lambda r(\mathbf{W}) \tag{3.7}$$

$$\mathbf{V}_{t+1} = \mu \mathbf{V}_t - \alpha \nabla L(\mathbf{W}_t), \qquad \mathbf{W}_{t+1} = \mathbf{W}_t + \mathbf{V}_{t+1}\ , \tag{3.8}$$

where M is the SGD batch size, $\mathbf{W}_t$ denotes the CNN weights at iteration t, and $\mathbf{V}_t$ denotes the weight update of the previous iteration. Parameters α and μ denote the learning rate and momentum, respectively.

We then compute similarities between exemplars by simply measuring correlation on the learned feature representation extracted from the CNN (see Sect. 3.3.1 for details).

3.2.5 Local Temporal Pooling

The proposed approach as described so far models posture by exploiting sample (dis-)similarities of single images. However, to learn the fine-grained similarities required to distinguish short-time actions, for instance, gait cycles of *walking* vs. *jogging*, not only posture matters but also how posture changes over short periods of time. This means that not only similarities need to be exploited, but also temporal information has to be incorporated in the model in order to model fine-grained relationships. Fortunately, a vast majority of the image data available for unsupervised learning contains this temporal information since it exists in the form of video sequences (e.g. YouTube videos), which can be seen as sequences of exemplars $v_i = \{x_1^i, x_2^i, \ldots, x_q^i\}$ and x_j^i is the $j-$th exemplar (i.e. $j-$th frame) of the $i-$th video sequence.

In this chapter, we introduce an effective approach to incorporate temporal information in our model by performing a local average pooling of the exemplar similarities on the temporal dimension. Given a pair of exemplars appearing in two

different video sequences (v_i, v_j), computing a simple global pooling over the entire sequence, as typically done for action classification [203], will result in losing fine-grained similarity structures over sub-sequences. In addition, modelling temporal context with complex recurrent architectures like Long Short-Term Memory (LSTM) networks [54] has proven useful for action classification. However, the temporal context that LSTMs encode cannot be learned for each exemplar, given a large number of exemplars available for unsupervised learning (e.g. the number of exemplars used in our experiments is in the order of 10^5).

To overcome these issues, we locally pool the similarities in a small temporal neighborhood (i.e. a short sub-sequence) of p frames around each exemplar. Formally, let $s = \phi'(x_k^i)^\top \phi'(x_l^j)$ be the similarity between two exemplars, where ϕ' is the feature representation learned by the CNN. Then, the similarity obtained by employing local temporal average pooling (LTP) is defined as:

$$s' = \frac{1}{2p+1} \sum_{n \in \{-p,+p\}} \phi'(x_{k+n}^i)^\top \phi'(x_{l+n}^j) \tag{3.9}$$

This method of modeling temporal context is fast and effective, giving us a boost in performance (cf. Sect. 3.3.1, 3.3.2) when temporal information is available in the dataset.

3.2.6 Multiple Instance Learning of Similarities

After a training round over all batches and performing local temporal pooling (LTP) we impute the similarities using the representation learned by the CNN. This is motivated by the fact that once the training process converges, the similarities that are learned are more reliable than the ones used for initialization, and thus, enable the grouping algorithm from Sect. 3.2.2 to find larger cliques of mutually related samples.

Since the number of unreliable similarities decreases after training the CNN, more samples can be comprised in a training batch and overall fewer batches already cover the same fraction of data as before training the CNN. Therefore, we alternately train the CNN, perform local temporal pooling on the resulting similarities and recompute cliques and batches using the similarities inferred in the previous step. This alternating imputation of similarities and training of the CNN follows the idea of multiple-instance learning and has shown to converge in less than four iterations.

To evaluate the improvement of the similarities Fig. 3.4 analyzes the eigenvalue spectrum of $\mathbf{S}$ on the Olympic Sports dataset, see Sect. 3.3.1. The plot shows the normalized cumulative sum of the eigenvalues as a function of the number of eigenvectors. Compared to the similarities used for initialization, transitivity relations are learned and the approach can generalize from an exemplar to more related samples. Therefore, the similarity matrix becomes more structured (cf. Fig.

Method	Olympic Sports	UCF Sports
HOG-LDA [97]	0.62	0.67
Ex-SVM [187]	0.72	0.71
Ex-CNN [55]	0.64	0.68
AlexNet [149]	0.65	0.68
1-sample CNN	0.67	0.59
NN-CNN	0.69	0.66
Doersch et al. [51]	0.62	-
Shuffle&Learn [196]	0.63	-
Ours	**0.83**	**0.78**
Ours + LTP	**0.84**	**0.79**

Table 3.1: Avg. ROC AUC for each method on Olympic Sports and UCF Sports datasets.

3.1) and random noisy relations disappear. As a consequence, it can be represented using very few basis vectors.

3.3 Experimental Evaluation

To compare our exemplar-based approach for unsupervised similarity learning with previous works we perform both quantitative and qualitative analysis. We conduct experiments on unsupervised fine-grained posture retrieval on 3 different Sports datasets: Olympic Sports [204], UCF Sports [226] and Leeds Sports Pose [131]. Furthermore, to demonstrate the capabilities of our model in the semi-supervised scenario we also tackled pose estimation on Leeds Sports [131] and MPII Pose Dataset [6]. Finally, provided the wide applicability of the proposed approach we also undertake the unsupervised setup of object classification (Pascal VOC 2007 dataset [67]).

3.3.1 Olympic Sports Dataset: Posture Analysis

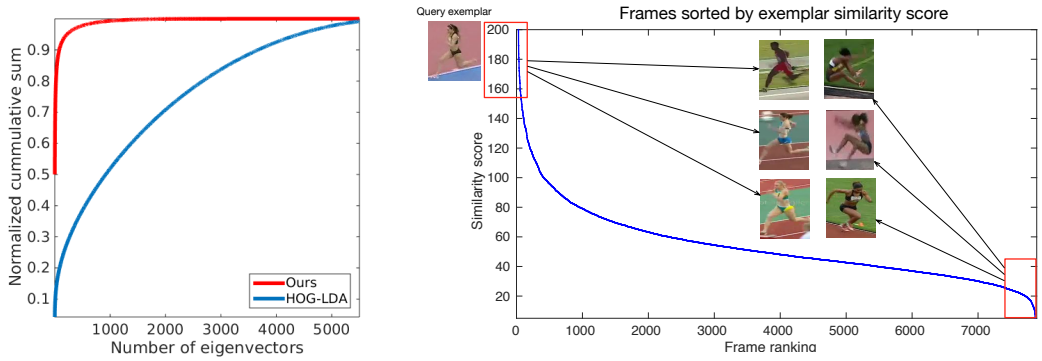
The Olympic Sports dataset [204] consists of video sequences of athletes practicing 16 different sports. The dataset contains an overall number of 113 516 frames, covering a rich set of human postures. As we aim to evaluate fine-grained pose similarity, we had independent annotators manually label 20 positive (similar) and negative (dissimilar) frames for 1033 query exemplars. We want to emphasize that these annotations are solely used for testing since our approach is unsupervised and does not utilize any labels during training.

We consider the following baselines for comparison with the proposed approach: HOG-LDA [97], Exemplar-SVMs [187], AlexNet [149] pretrained on ImageNet [48], 1-sample CNN and NN-CNN models (in a very similar spirit to [243]), Exemplar-CNN [55], the two-stream approach of Doersch et al. [51], and Shuffle&Learn [196]. To compute person bounding boxes we use the approach of [71] as it shows

Figure 3.3: Visual example of a resulting batch of cliques for long jump category of Olympic Sports dataset. Each clique contains at least 20 samples and is represented as their average.

reasonable performance in object and person detection. *(i)* The evaluation must explore the benefit of the unsupervised selecting of batches of cliques for deep learning of exemplars using standard CNN architectures. For that reason, we incarnate our approach by adopting the widely used architecture of Krizhevsky et al. [149]. To build batches for training the neural network we solve the optimization problem in Eq. (3.1) with $B = 100$, $K = 100$, and $r = 20$ and fine-tune the model for 10^5 iterations. For the temporal average pooling we took a temporal neighborhood of 3 frames around each exemplar. After that we measure similarities using features extracted from layer fc7 in the *caffe* implementation of [149]. *(ii)* Exemplar-CNN is trained using the best performing parameters reported in [55] and the 64c5-128c5-256c5-512f architecture. Then we extract fc4 features and compute 4-quadrant max pooling. *(iii)* Exemplar-SVM is trained on the query exemplars using the HOG descriptor. Hard negative mining is run on all the samples from all sports categories except the one which the exemplar belongs to. We find an optimal number of negative mining rounds (less than three) using cross-validation and set the class weights of the linear Support Vector Machine (SVM) as $C_1 = 0.5$ and $C_2 = 0.01$. *(iv)* We compute LDA whitened HOG using approach from [97]. *(v)* The 1-sample CNN is trained by defining a separate class for each exemplar sample plus one class containing all other samples. *(vi)* In a similar fashion, the NN-CNN is trained using the exemplar plus 10 nearest neighbors obtained using the whitened HOG similarities. Both CNNs were implemented using the model of [149] and fine-tuning it for 10^5 iterations. We employ AdaGrad [58] solver with a batch size of 128, learning rate of 0.001 and smoothing term of 0.0001. Each image in the training set was augmented with 10 transformed versions by performing random translation, scaling, rotation and color transformation, to improve invariance with respect to these.

In Tab. 3.1 we report the average area under the Receiver Operating Characteristic curve (ROC AUC) for each method over all categories of the Olympic Sports dataset. More specifically, the experiments witness that the 1-sample CNN fails to model the positive distribution, due to the high imbalance between positives and negatives and the resulting biased gradient. In contrast, extra nearest neighbors to the exemplar (NN-CNN) yield a better model of the intra-class variability of the exemplar leading to a 2% performance boost over the 1-sample CNN. However, NN-CNN also sees



(a) (b)

Figure 3.4: (a) Cumulative distribution of the spectrum of the similarity matrices obtained by our method and the HOG-LDA initialization. (b) Sorted similarities with respect to one exemplar, where only similarities at the ends of the distribution can be trusted.

a large set of negatives, which are partially similar and dissimilar. Due to lack of structure in the negative set, NN-CNN fails to thoroughly capture the fine-grained similarities in the negative samples. To avoid this issue we compute sets of mutually distant compact cliques resulting in a performance increase of 14% over NN-CNN. In addition, using local temporal pooling on the exemplar similarities with a neighborhood radius $p = 3$ yields a further improvement of 1%.

Qualitatively, Fig. 3.1 renders the similarity matrices obtained by different approaches for a video sequence of the *long jump* category. In these matrices, the parallel diagonals indicate the gait cycle of a person running before leaping into the sandpit. We can see how the method proposed in this chapter clearly highlights these gait cycles while filtering noisy similarity relationships. In addition, to visually assess the similarities we average the top 50 nearest neighbors for a randomly chosen exemplar frame in the Olympic Sports dataset. Fig. 3.2 shows how the neighbors obtained by our approach depict a sharper average posture since they come from compact cliques of mutually similar exemplar frames. Therefore frames are more similar to the original and more details of the posture are retained than in case of the other methods. Finally, in Fig. 3.5 we show nearest neighbors for few representative query images of the dataset.

3.3.2 UCF Sports Dataset: Transferring Posture Representations

The UCF Sports dataset [226] contains a set of actions from various sports. Originally, this dataset consists of 12 categories. We disregarded the categories in which the posture does not change (e.g. Horse Riding) keeping 7: diving side, golf swing side, kicking (kicking-front and kicking-side were merged together), weight lifting, run side, swing bench, swing side angle. Having a total of 5148 frames over all categories, this dataset fails to fulfill with data volumes required to train deep CNN

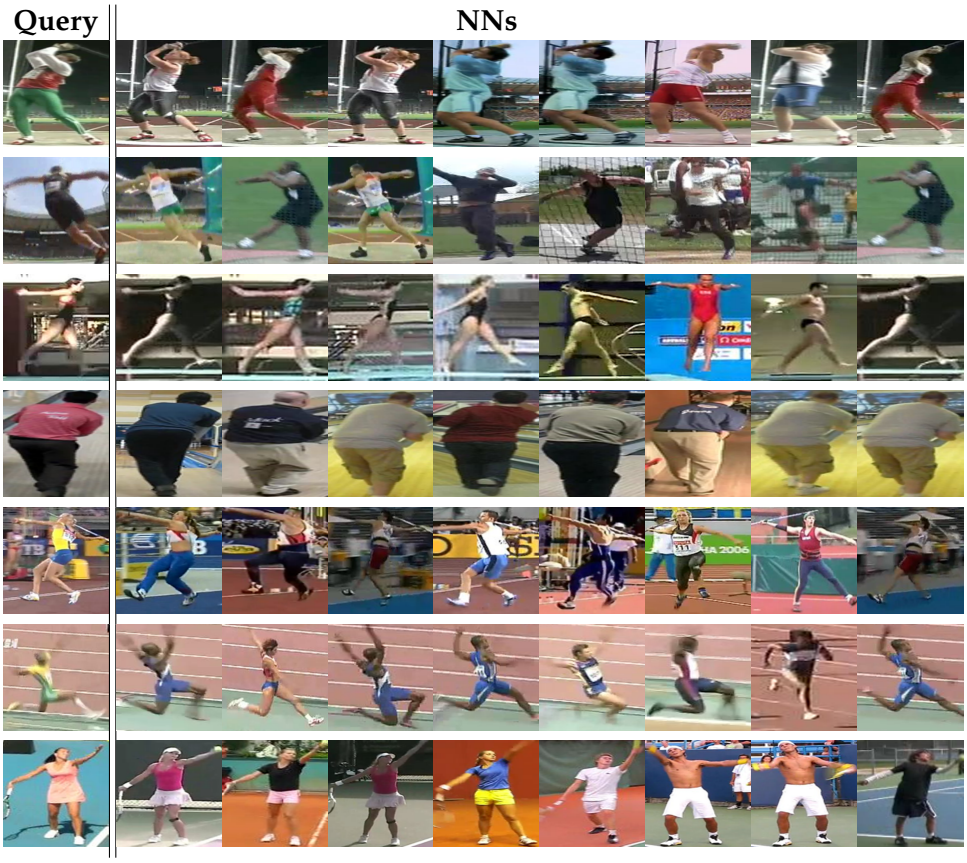


Figure 3.5: Nearest neighbors retrieved by the proposed approach for representative query images of the Olympic Sports dataset.

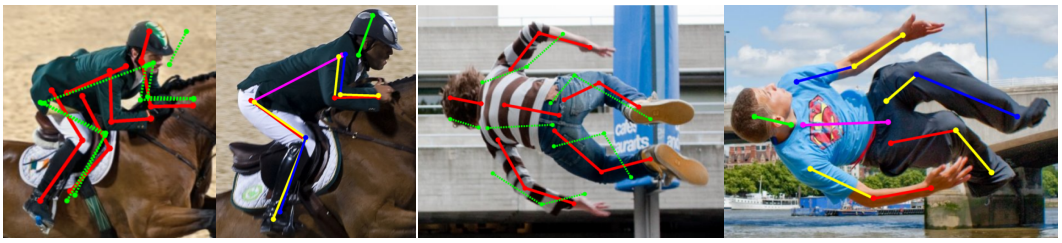
models. In such scenarios, where little training data is available, transfer learning has been proved to be a useful approach.

Therefore, we leverage the bigger Olympic Sports dataset and transfer the models learned on Olympic Sports categories using them solely for computing similarities of on the data of the UCF Sports dataset. We visually matched 4 categories of Olympic Sports to UCF Sports and transfer the learned models: hammer-throw and kicking, hammer-throw and swing-bench, diving-springboard-3m and swing-side-angle, long-jump and run-side. A visual example of the matching postures between UCF Sports and Olympic Sports dataset is shown in Fig. 3.6.

Analogously to the Olympic Sports dataset, independent annotators manually labeled 20 positive (similar) and negative (dissimilar) frames for around 150 exemplars in the above selected 4 categories. These annotations are solely used for testing since we do not train on UCF Sports dataset at all.

We report the average ROC AUC for our approach, Exemplar-CNN [55], 1-sample CNN, NN-CNN models, AlexNet [149], Exemplar-SVMs [187], and HOG-LDA [97]. For each of the CNN-based approaches we simply transfer the learned

Figure 3.6: Based on the similarity structure learned by our model on Olympic Sports, postures are matched between Olympic (top row) and UCF (bottom row) Sports dataset. At the bottom is the most similar frame to the one at the top.



(a) (b) (c) (d)

Figure 3.7: Pose prediction results. (a) and (c) are test images with the superimposed ground truth skeleton depicted in red and the predicted skeleton in green. (b) and (d) are corresponding nearest neighbors, which were used to transfer pose.

Method	Torso	Upper legs	Lower legs	Upper arms	Lower arms	Head	Total
Shuffle&Learn [196]	60.4	33.2	28.9	16.8	7.1	33.8	30.0
AlexNet [149]	76.9	47.8	41.8	26.7	11.2	42.4	41.1
HOG-LDA [97]	73.7	41.8	39.2	23.2	10.3	42.2	38.4
Ours	80.1	50.1	45.7	27.2	12.6	45.5	43.5
Ground Truth	**93.7**	78.8	74.9	58.7	36.4	72.4	69.2
Pose Machines [222]	93.1	**83.6**	**76.8**	**68.1**	**42.2**	**85.4**	**72.0**

Table 3.2: PCP measure for each method on Leeds Sports dataset, using the retrieval-based estimation for joint positions.

representations from the matched categories of the Olympic Sports dataset, so no additional training is required. The experimental settings are the ones described in Sect. 3.3.1.

Tab. 3.1 shows the average AUC for each of the compared methods on the 4 categories of UCF Sports. In particular, our approach attains a significant performance improvement of at least 7% with respect to all compared methods. Furthermore, when temporal information is incorporated in the model by pooling the similarities using local temporal pooling we obtain a further improvement of 1%. These results support the fact that the feature representation learned by our approach in Olympic

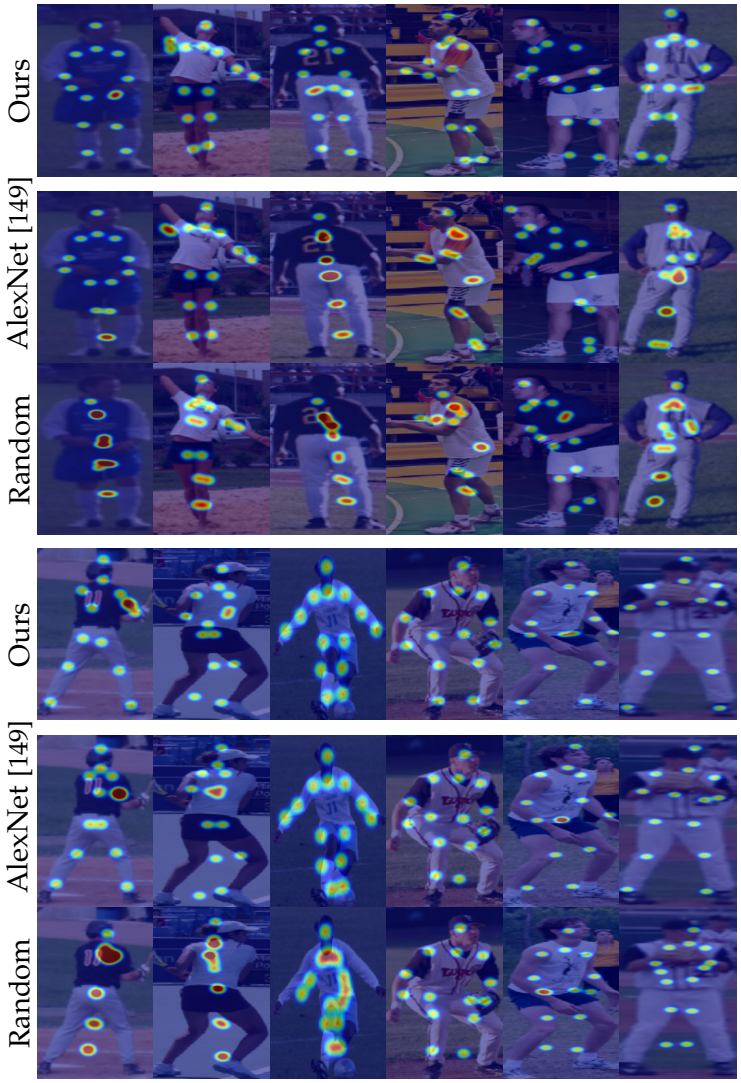


Figure 3.8: Heatmaps obtained by DeepPose (stg-1) [266] trained on LSP using different models as initialization.

Sports encodes a general notion of posture, and therefore can be transferred without requiring any further learning to different categories of the UCF dataset.

3.3.3 Leeds Sports Dataset: Pose Estimation

The Leeds Sports Pose (LSP) Dataset [131] is a well-known and widely used benchmark for pose estimation. This dataset consists of 1000 images for training which we combine with 4000 images from its extended version, where each image is annotated with all 14 joint locations from a person-centric viewpoint. Finally, the test set consists of 1000 images.

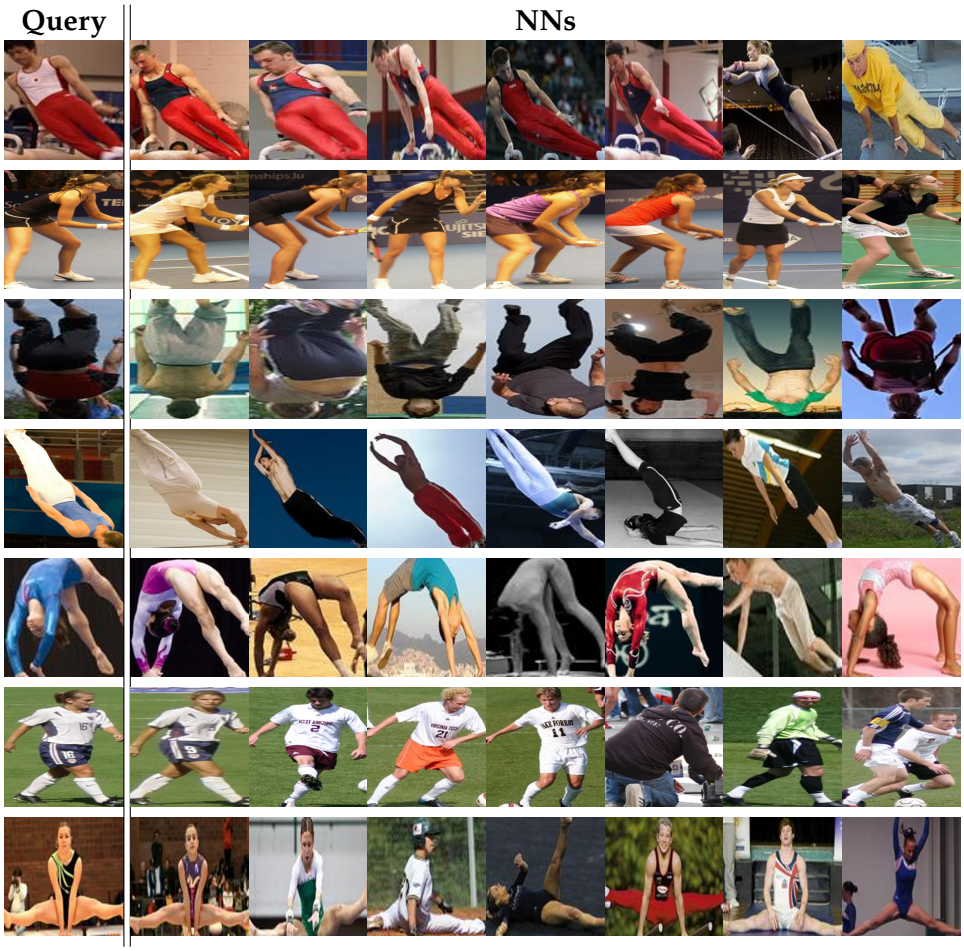


Figure 3.9: Nearest neighbors retrieved by the proposed approach for representative query images of the Leeds Sports dataset.

We now evaluate the proposed approach on the problem of unsupervised pose estimation on LSP. During training, we disregard all joint annotations from the training set and learn pose similarities. During testing, these pose similarities yield by our approach are used to find frames similar in posture to a query frame. The joint locations of the test image are then estimated by identifying its nearest neighbor from the training set and transferring its joint locations to the test image.

For training our model we use the parameters described in Sect. 3.3.1. The similarity between two images is measured as Pearson correlation on features extracted from layer fc6. To evaluate the results we use the Percentage of Correct Parts (PCP) measure, which is the standard metric for benchmarking pose estimation methods.

For comparison with other methods, we follow the same testing protocol and retrieve similar postures using HOG-LDA [97], and fc6 representations of AlexNet [149] and Shuffle&Learn [196]. In addition, we also report an upper bound on the

Initialization	Torso	Upper legs	Lower legs	Upper arms	Lower arms	Head	Total
Random	87.3	52.3	35.4	25.4	7.6	44.0	42.0
Shuffle&Learn [196]	90.4	62.7	45.7	33.3	11.8	52.0	49.3
AlexNet [149]	92.8	68.1	53.0	39.8	17.5	62.8	55.7
Ours	**93.9**	**71.2**	**55.0**	**44.5**	**21.6**	**63.2**	**58.2**

Table 3.3: PCP measure for each method on Leeds Sports dataset using different models as initialization for training DeepPose [266].

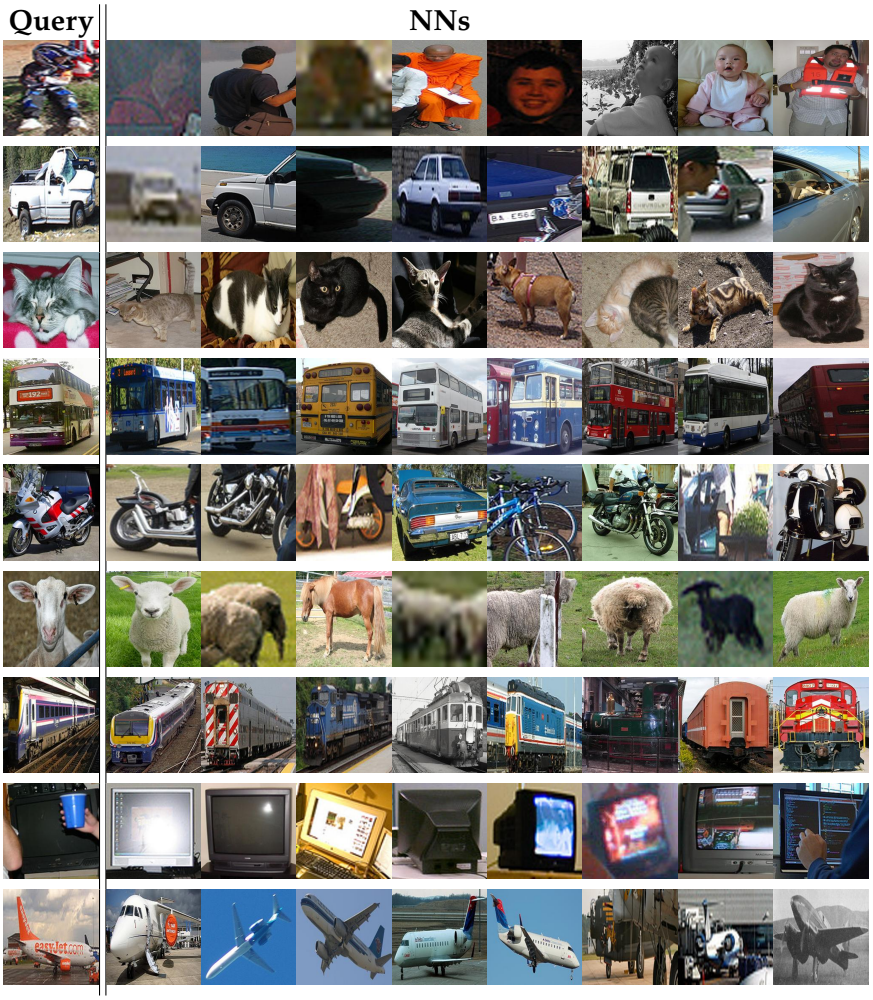


Figure 3.10: Nearest neighbors retrieved by the proposed approach for representative query images of the VOC2007 dataset.

performance that can be achieved by the nearest neighbor joint transfer, using ground-truth similarities to retrieve nearest neighbors. Therefore, the nearest training pose for a test image is identified by minimizing the average Euclidean distance between their ground-truth pose annotation. This is the best result one

Initialization	Head	Neck	LR Shoulder	LR Elbow	LR Wrist	LR Hip	LR Knee	LR Ankle	Thorax	Pelvis	Total
Random	79.5	87.1	71.6	52.1	34.6	64.1	58.3	51.2	85.5	70.1	65.4
Shuffle&Learn [196]	75.8	86.3	75.0	59.2	42.2	73.3	63.1	51.7	87.1	79.5	69.3
AlexNet [149]	87.2	93.2	85.2	69.6	52.0	81.3	69.7	62.0	93.4	86.6	78.0
Ours	**89.5**	**93.7**	**85.9**	**71.6**	**56.3**	**82.7**	**72.4**	**67.3**	**93.8**	**88.3**	**80.2**

Table 3.4: PCKh@0.5 measure for different limbs on MPII Pose benchmark dataset using different initializations for the DeepPose approach [266].

can achieve by finding the most similar pose, when not provided with a supervised parametric model (the performance gap to 100% shows the degree of difference between training and test poses). For completeness, we also compare with a fully supervised state-of-the-art approach for pose estimation [222]. We use the same experimental settings described in Sect. 3.3.1.

The Percentage of Correct Parts @0.5 (PCP) for different approaches is reported in Tab. 3.2. In Tab. 3.2 our approach improves the visual similarities learned using both AlexNet and HOG-LDA. It is noteworthy that even though our approach for estimating the pose is *fully unsupervised* it achieves a competitive performance when compared to the upper-bound of supervised ground truth similarities. Qualitative results of nearest neighbors for several query frames are presented in Fig. 3.9.

In addition, Fig. 3.7 shows success (a) and failure (c) cases of our method. In Fig. 3.7(a) we can see that the pose is correctly transferred from the nearest neighbor (b) from the training set, resulting in a PCP score of 0.6 for that particular image. Moreover, Fig. 3.7(c), (d) witness that our method learns the representation invariant to front-back flips (matching a person facing away from the camera to one facing the camera). Since our approach learns pose similarity in an unsupervised manner, it becomes invariant to changes in appearance as long as the shape is similar, thus explaining this confusion. Adding extra training data or directly incorporating face detection-based features could resolve this.

Furthermore, in addition to the fully unsupervised experiment, we evaluate the representation learned by the proposed approach on LSP, in a semi-supervised scenario, by using it as initialization for the supervised DeepPose [266] method. We train DeePose (stg-1) [266] with using different initializations: (a) random initialization, (b) ImageNet pre-trained AlexNet [149] (c) Shuffle&Learn [196] and (d) our model trained on LSP dataset. We then follow the training procedure described in [266], where the train split includes 11000 images (using the extended LSP data), and the test split includes 1000 images. We use a batch size of 128, learning rate of 5×10^{-4} and optimize the CNN parameters using AdaGrad [58].

Tab. 3.3 shows the PCP@0.5 score of DeepPose (stg-1) model trained using different methods as initialization. Using our model to initialize DeepPose (stg-1) yields a performance boost of 2.5% over ImageNet pretrained AlexNet [149] initialization. Showing, as a result, that the representation learned by our model successfully encodes relevant pose information which is not only good for unsupervised pose retrieval but can further facilitate the training of supervised pose estimation methods.

HOG-LDA	Wang et al. [285]	Wang et al. [285] + **Ours**	AlexNet [149]	R-CNN [85]
0.118	0.450	0.481	0.616	0.683

Table 3.5: K-nearest neighbors classification results on Pascal VOC 2007 using the visual representations learned by different methods.

Finally, in Fig. 3.8 we show the predicted joint heatmaps obtained by DeepPose [266] when using the three different initialization models for several representative images of LSP dataset [131].

3.3.4 MPII Dataset: Pose Estimation

Next, to further assess the reliability and robustness of the pose representation learned by our model, we tackle the challenging MPII Pose dataset [6]. MPII Pose dataset [6] is a state of the art benchmark for evaluation of articulated human pose estimation. MPII Pose is a particularly challenging dataset because of the clutter, occlusion and number of persons appearing in images. To evaluate our approach on MPII Pose we follow the semi-supervised training protocol used for LSP and compare the performance obtained by DeepPose (stg-1) [266], when trained using as initialization each of the following models: Random initialization, ImageNet pre-trained AlexNet [149], Shuffle&Learn [196] and our approach trained on LSP in unsupervised manner (Sec. 3.3.3). We use PCKh@0.5 on all the keypoints of the full body as evaluation metric which is the standard for MPII dataset [6]. PCKh@0.5 measures accuracy of the predicted body joint coordinates, where the matching threshold equals to 50% of the head segment length. Tab. 3.4 reports the PCKh@0.5 obtained by the DeepPose (stg-1) models [266] with different initializations. In particular, when comparing our unsupervised initialization with a random initialization we obtain a 15% performance boost, which indicates that our features encode a notion of pose that is robust to the clutter present in MPII dataset. Furthermore, we obtain a 2.2% improvement over ImageNet-pretrained AlexNet model [149]. The performance obtained on MPII Pose dataset corroborates that the representation learned by our method captures fine-grained posture details and successfully deals with clutter, occlusions, and presence of multiple persons in this dataset.

3.3.5 Pascal VOC 2007: Object Classification

Provided the wide applicability of our method, in addition to the experiments on pose estimation datasets in the previous sections we now evaluate the learning of similarities over object categories. For this purpose, we classify object bounding boxes of the Pascal VOC 2007 dataset [67]. Instead of predicting the bounding box position and category, we assume that bounding boxes are given, provided recent

outstanding results for object [85] and objectness [3] detection, and focus directly on the object classification.

To initialize our model we use the visual similarities of Wang et al. [285] without applying any fine tuning on Pascal and also compare against this approach. Thus, neither ImageNet nor Pascal VOC labels are utilized during training or pre-training. We then evaluate how our model performs in comparison with features obtained by HOG-LDA [97], Wang et al. [285], AlexNet [149] pretrained on ImageNet, and R-CNN [85] which is pretrained on ImageNet and finetuned in a supervised manner on Pascal VOC. For our method and HOG-LDA we use the same experimental settings as described in Sect. 3.3.1.

At test time, we perform K-nearest neighbors classification for all methods. The k nearest neighbors are computed using similarities (Pearson correlation) based on the feature representation obtained for each method. In Tab. 3.5 we show the classification accuracies of all approaches for $k = 5$ (for $k > 5$ there was only insignificant performance improvement). We can see how our approach improves upon Wang et al. [285] used as initialization to our model to yield a performance gain of 3% without requiring any supervision information or fine-tuning on Pascal. Finally, in Fig. 3.10 we show the retrieved nearest neighbors for few query samples of different categories of the Pascal VOC dataset [67].

3.4 Conclusion

In this chapter, we have proposed a technique for deep unsupervised learning of visual similarities between a large number of exemplars. We analyze the shortcomings of exemplar learning on CNNs and address the single positive exemplar setup, the imbalance between exemplar and negatives, and inconsistent labels within SGD batches. We address these key problems by optimizing a single cost function yielding SGD batches of compact, mutually dissimilar cliques of samples. Each of these cliques then gets assigned a surrogate label, and the learning of visual similarities is then posed as a categorization task on individual batches.

In the experimental evaluation the proposed approach has shown competitive performance compared to the state-of-the-art, providing significantly finer similarity structure that is particularly crucial for detailed posture analysis. Furthermore, the experimental evaluation in several pose datasets shows that the pose representation learned by our model in an unsupervised manner is transferable across pose datasets and can be used in conjunction with supervised parametric models for pose estimation to boost their performance. Finally, the proposed approach also demonstrates competitive performance in general object classification problems. Overall, our experimental results show that the representation learned by our model generalizes well to a spectrum of different tasks and datasets.

4 Unsupervised Representation Learning using Partially Ordered Sets[1]

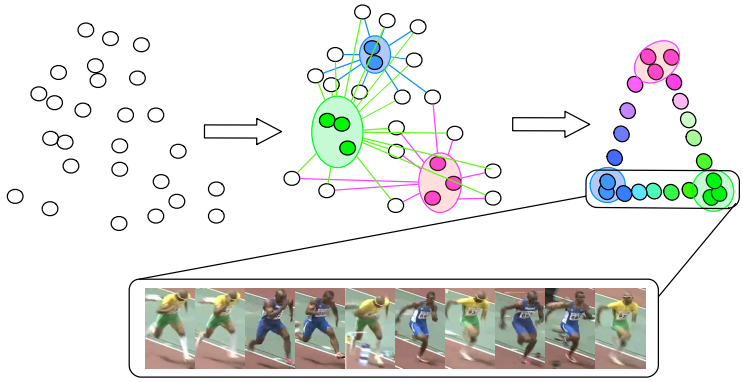


Figure 4.1: Visualization of the interaction between surrogate classes and partially ordered sets (posets). Our approach starts with a set of unlabeled samples, building small surrogate classes and generating posets to unlabeled samples to learn fine-grained similarities.

As discussed in the previous chapters, to utilize the vast amounts of available unlabeled training data, there is a quest to leverage context information intrinsic to images/video for *self*-supervision. However, this context is typically highly local (i.e position of patches in the same image [51], object tracks through short number of frames [285] or image inpainting [215]), establishing relations between tuples [51] or triplets [196, 285, 307] of images. Hence, these approaches utilize loss functions that order a positive x_p and a negative x_n image with respect to an anchor image x_a so that, $d(x_a, x_p) < d(x_a, x_n)$, where d is a distance in the representation space. During training, these methods rely on the Convolutional Neural Network (CNN) to indirectly learn comparisons between samples that were processed in independent training batches, and generalize to unseen data.

Instead of relying on the CNN to indirectly balance and learn sample comparisons unseen during training, a more natural approach is to explicitly encode richer

[1]This chapter is based on joint work [12] with Miguel A. Bautista and Björn Ommer, originally presented at CVPR 2017. References to prior work (such as "existing approaches", "recent methods", or "state-of-the-art methods") should be read with this context in mind.

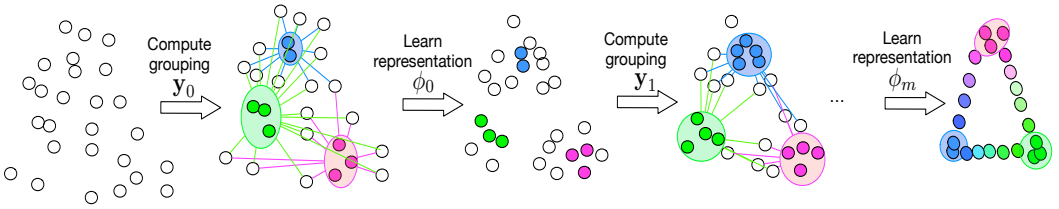


Figure 4.2: Visual summary of our approach. In the $\mathbf{y}$-steps, the clustering procedure computes surrogate classes (shaded in color) based on the current representation. In the ϕ-steps, we learn a representation using the surrogate classes and partial orders of samples not assigned to any surrogate class (samples in white) by pulling them closer to their nearest classes and pushing them further from the rest.

relationships between samples as supervision. In this sense, an effective approach for unsupervised representation learning is to frame it as a series of surrogate (i.e., artificially created) classification tasks, which was proposed in Chapter 3. Therefore, mutually similar samples are assigned the same class label, otherwise a different label. To obtain surrogate classification tasks, compact groups of mutually similar samples are computed by clustering [11] over a weak initial representation (e.g, standard features such as Histogram of Oriented Gradients (HOG)). Then, each group receives a mutually exclusive label, and a CNN is trained to solve the associated classification problem, thereby learning a representation that encodes similarity in the intermediate layers. However, given the unreliability of initial representation, many training samples are neither mutually similar nor dissimilar and are, thus, not assigned to any of the compact surrogate classes. Consequentially they are ignored during training, hence overlooking important information. Also, classification can yield relatively coarse similarity measure, considering the discrete nature of the classes. Furthermore, the similarities learned by the different classification tasks are not optimized jointly, which can lead to mutually contradicting relationships since transitivity is not captured.

To overcome these fundamental limitations, we propose to: *(i)* Cast representation learning as a surrogate classification task, using compact groups of mutually related samples as surrogates classes (as in Chapter 3). *(ii)* Combine classification with a partial ordering of samples. Even samples, which cannot be assigned to any surrogate class due to unreliable initial similarities, are thus incorporated during training and, in contrast to discrete classification, more fine-grained relationships are obtained due to the ordering. *(iii)* Explicitly optimize similarities in a given representation space, instead of using the representation space indirectly learned by intermediate layers of a CNN trained for classification. *(iv)* Jointly optimize the surrogate classification tasks for similarity learning and the underlying grouping in a recurrent framework that is end-to-end trainable. Fig. 4.2 shows a conceptual pipeline of the proposed approach.

Experimental evaluation on diverse tasks of pose estimation and object classification shows an improvement over the method introduced in Chapter 3 achieving state-of-the-art performance on standard benchmarks, thus underlining the broad applicability of the proposed approach. In the pose estimation experiments, we show that our method learns a generalizable representation, which can be transferred across datasets and is even valuable for the initialization of supervised methods. Also, in the object classification experiments, we successfully leverage large unlabeled datasets to learn representations in the fashion of zero-shot learning [150].

4.1 Related Work

Representation learning has been a problem of major interest for the vision community from its early beginnings due to its broad applications. With the advent of CNNs, several approaches have been proposed for supervised representation learning using either pairs [313], or triplets [281] of images. Furthermore, recent works by Misra et al. [196], Wang et al. [285], and Doersh et al. [51] showed that temporal information in videos and spatial context information in images can be utilized as a convenient supervisory signal for learning feature representation with CNNs in an unsupervised manner. However, either supervised or unsupervised, all these formulations for learning representations (or similarities induced by the representations) require that the supervisory information scales quadratically for pairs of images, or cubically for triplets. This results in a very large training time. Furthermore, tuple and triplet formulations advocate on the CNN to indirectly learn to conceal unrelated pairs of samples (i.e., pairs that were not tied to any anchor) that are processed in different, independent batches during training. Another recent approach that has been proposed for learning similarities in an unsupervised manner is to build a surrogate (i.e., an artificial) classification task either by utilizing heavy data augmentation [55] or by clustering based on initial weak estimates of similarities (presented in Chapter 3). The advantage of these approaches over tuple or triplet formulations is that several relationships of similarity (samples in the same class) and dissimilarity (samples in other classes) between samples are utilized during training. This results in more efficient training procedures, avoiding to sample millions of pairs or triplets of samples, and encoding richer relationships between samples.

In Chapter 1, Section 1.1.2, we also discussed unsupervised representation learning approaches that appeared after we published the work presented in the current chapter. In addition, in Chapter 1, Section 1.1.1 and Chapter 2 we have studied representation learning from the perspective of Deep Metric Learning approaches.

Recently, Wang et al. [289] leveraged low-density classifiers to enable the use of large volumes of unlabeled data during training. However, [289] cannot be

successfully applied to the unsupervised scenario since it requires a strongly supervised initialization, e.g., pretraining on ImageNet [48].

4.2 Approach

In this section, we show how to combine partially ordered sets (posets) of samples and surrogate classification to learn fine-grained similarities in an unsupervised manner. Key steps of the approach include: *(i)* Compute compact groups of mutually related samples and use each group as a surrogate class in a classification task. *(ii)* Learn fine-grained similarities by modeling partial orderings to also leverage those samples that cannot be assigned to a surrogate class. *(iii)* Due to the interdependence of grouping and similarity learning, we jointly optimize them in a recurrent framework. Fig. 4.2 shows a visual example of the main steps of our approach.

4.2.1 Grouping

To formulate unsupervised similarity learning as a classification approach, we need to define surrogate classes since labels are not available. To compute these surrogate classes, we first gather compact groups of samples using distances based on standard features like HOG-LDA (LDA whitened HOG [61, 97, 231]). HOG-LDA is a computationally effective foundation for estimating similarities between a large number of samples. Let our training set be defined as $X \in \mathbb{R}^{n \times p}$, where n is the total number of samples, and x_i is the $i-$th sample. Then, the HOG-LDA similarity between a pair of samples x_i and x_j is defined as $s_{ij} = \exp(-\|\phi(x_i) - \phi(x_j)\|_2)$. Here $\phi(x_i) \in \mathbb{R}^{1 \times d}$ is the $d-$dimensional representation of sample x_i in the HOG-LDA feature space.

Albeit unreliable to relate all samples to another, HOG-LDA similarities can be used to find the nearest and furthest neighbors, as highly similar and dissimilar samples to a given anchor sample x_i stand out from the similarity distribution. Therefore, to build surrogate classes (i.e., compact groups of samples), we group each x_i with its immediate neighborhood (samples with similarity within the top 5%) so that all merged samples are mutually similar. These groups are compact, differ in size, and may be mutually overlapping. To reduce redundancy, highly overlapping classes are subsequently merged by agglomerative clustering, which terminates if the intra-class similarity of a surrogate class is less than half of its constituents. We denote the set of samples assigned to the c-th surrogate class as $\mathcal{C}_c$, and the label assigned to each sample as $\mathbf{y} \in \{-1, 0, \ldots, C-1\}^{1 \times n}$, where the label assigned to sample x_i is denoted as y_i. All samples that are not assigned to any surrogate class get label -1.

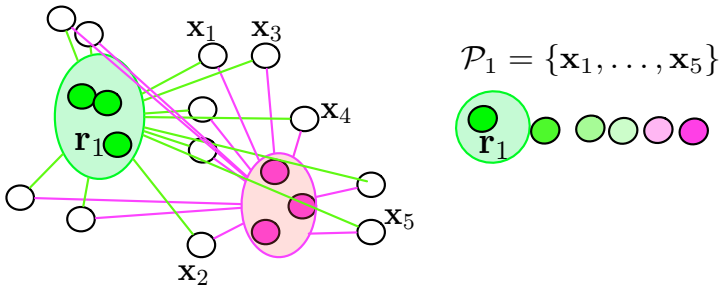


Figure 4.3: Visual interpretation of a poset. Samples assigned to a surrogate class are shaded in a particular color, while samples not assigned to surrogate classes are represented in white.

4.2.2 Partially Ordered Sets

Provided the unreliability of similarity estimates used for building surrogate classes, a large number of samples cannot be assigned to any class, because they are neither similar nor dissimilar to any sample. This deprives the optimization of using all available data during training. As a result, fine-grained similarities are poorly represented, since learning to classify surrogate classes does not model relative similarities of samples that are not assigned to any class. To overcome this limitation, we leverage the information encoded in posets of samples relative to a surrogate class. That is, for each sample not assigned to any surrogate class (i.e. $x_i : y_i = -1$) we compute a soft assignment (i.e. a similarity score) to the Z nearest surrogate classes $\mathcal{C}_z : z \in \{1, \ldots, Z\}$. Once all unlabeled points are softly assigned to their Z nearest classes, we obtain as a result, a poset $\mathcal{P}_c$ for each class. Thus, a poset $\mathcal{P}_c$ is a set of samples which are softly assigned to class $\mathcal{C}_c$. Posets can be of variable size and partially overlapping. We show a visual example of a poset in Fig. 4.3.

Formally, given a deep feature representation ϕ^θ (e.g an arbitrary layer in a CNN with parameters θ), and a surrogate class $\mathcal{C}_c$, a poset of unlabeled samples $\mathcal{P}_c = \{x_j, \ldots, x_k\} : y_j = y_k = -1 \; \forall \; j, k$ with respect to $\mathcal{C}_c$ is defined as:

$$\begin{aligned} &\forall_{x_i \in \mathcal{C}_c} \{\exp(-\|\phi^\theta(x_i) - \phi^\theta(x_j)\|_2) > \\ &\exp(-\|\phi^\theta(x_i) - \phi^\theta(x_k)\|_2)\} \iff j < k \forall j, k. \end{aligned} \tag{4.1}$$

In Eq. (4.1) a poset is defined by computing the similarity of unlabeled sample x_j to all the samples in class $\mathcal{C}_c$, which during training is costly to optimize. However, due the compactness of our grouping approach, which only gathers very similar samples into surrogate $\mathcal{C}_c$, we can effectively replace the similarities to all points in $\mathcal{C}_c$ by the similarity to a representative sample $\bar{x}_c$ in $\mathcal{C}_c$, which is the class medioid, $\bar{x}_c = \underset{x_i \in \mathcal{C}_c}{\arg\min} \sum_{x_j \in \mathcal{C}_c} \|\phi^\theta(x_i) - \phi^\theta(x_j)\|_2$.

Following the definition of a poset in Eq. (4.1), the widely adopted tuple and triplet formulations [51, 196, 285, 307] are a specific case of a poset in which $\mathcal{P}$ contains at most 2 samples, and $\mathcal{C}_c$ contains just one. In this sense, deep feature representations ϕ (i.e., CNNs) trained using triplet losses seek to sort two pairs of samples (i.e., anchor-positive and anchor-negative) according to their similarity. As a result, triplet formulations rely on the CNN to *indirectly* learn to compare and reconcile the vast number of *unrelated* sampled pairs that were processed on different, independent mini-batches during training. In contrast, posets, explicitly encode an ordering between a large number of sample pairs (i.e., pairs consisting of an unlabeled sample and its nearest class representative). Therefore, using posets during training enforces the CNN to order all unlabeled samples $x_i : y_i = -1$ according to their similarity to the Z nearest class representatives $\mathbf{r}_i^z : z \in \{1, \dots, Z\}$, where $\mathbf{r}_i^z$ is the $z-$th nearest $\bar{x}_c$ to sample x_i, learning fine-grained interactions between samples. Posets generalize tuple and triplet formulations by encoding similarity relationships between unlabeled samples to make a decision whether to move closer to a surrogate class. This effectively increases our training set when compared to just using the samples assigned to surrogate classes and allows us to model finer relationships.

4.2.3 Objective function

In our formulation, we strive for a trade-off model in which we jointly optimize a surrogate classification task and a metric loss to capture the fine-grained similarities encoded in posets. Therefore, we seek an objective function $\mathcal{L}$ which penalizes: *(i)* misclassifications of samples x_i with respect to their surrogate label y_i, and *(ii)* similarities of samples $x_i : y_i = -1.$ with respect to their Z nearest class representatives. The objective function should inherit the reliability of framing similarity learning as surrogate classification tasks while using posets to incorporate those training samples that were previously ignored because they could not be assigned to any surrogate class. In particular, we require the CNN to pull samples from posets $x_i \in \mathcal{P}_c$ closer to their Z nearest class representatives, while pushing them further from all other class representatives in a training mini-batch. Furthermore, we require that unreliable similarities (i.e., samples that are far from all surrogate classes) vanish from the loss, rendering the learning process robust to outliers. In addition, in order to capture fine-grained similarity relationships, we want to directly optimize the feature space ϕ in which similarities are computed.

Therefore, let $\mathbf{R}^z \in \mathbb{R}^{n \times d}$ denote the z-th nearest class representatives of each unlabeled sample $x_i : y_i = -1$, where $\mathbf{r}_i^z$ is the z-th nearest class representative of sample x_i, and θ be the parameters of the CNN. Then, our objective function combines the surrogate classification loss $\mathcal{L}_1$ with our poset loss $\mathcal{L}_2$:

$$\mathcal{L}(x_i, y_i, \mathbf{R}; \theta) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_1(x_i, y_i) + \lambda \mathcal{L}_2(x_i, \mathbf{R}, \phi), \tag{4.2}$$

where λ is a scalar and,

$$\mathcal{L}_1(x_i, y_i; \theta) = -\log \frac{\exp(t^{\theta}_{i,y_i})}{\sum_{j=0}^{C-1} \exp(t^{\theta}_{i,j})} \mathbb{1}_{y_i \neq -1}, \tag{4.3}$$

$$\mathcal{L}_2(x_i, \mathbf{R}; \theta) = \\ = -\log \frac{\sum_{z=1}^{Z} \exp(\frac{-1}{2\sigma^2}(\| \phi^{\theta}(x_i) - \phi^{\theta}(\mathbf{r}_i^z)\|_2^2 - \gamma))}{\sum_{j=1}^{C'} \exp(\frac{-1}{2\sigma^2}\| \phi^{\theta}(x_i) - \phi^{\theta}(\mathbf{r}_j)\|_2^2)}. \tag{4.4}$$

In Eq. (4.3), $\mathbf{t}_i^{\theta} = \mathbf{t}^{\theta}(x_i)$ are the logits of sample x_i for a CNN with parameters θ. In Eq. (4.4) C' is the number of surrogate classes in the batch, σ is the standard deviation of the current assignment of samples to surrogate classes, and γ is the margin between surrogate classes. It is note-worthy that Eq. (4.4) can scale to an arbitrary number of classes, since it does not depend on a fixed-sized output target layer, avoiding the shortcomings of large output spaces in CNN learning [275] [2].

Finally, note that if $Z = 1$, the problem reduces to a cross-entropy based classification, where the standard logits (i.e., outputs of the last layer) are replaced by the similarity to the surrogate class representative in feature space ϕ. However, for $Z > 1$, relative similarities between surrogate classes enter into play and posets encoding fine-grained interactions arise naturally (cf. Fig. 4.5). In all our experiments, we set $Z \geq 2$. During training, CNN parameters θ are updated by error-backpropagation with stochastic mini-batch gradient descent. In typical classification scenarios, the training set is randomly shuffled to avoid biased gradient computations that hamper the learning process. Therefore, at training time we build our mini-batches of samples by selecting a random set of samples not assigned to a surrogate class $x_i \ : \ y_i = -1$, and retrieving all the surrogate classes $\mathcal{C}_c$ which contain x_i in their poset $x_i \in \mathcal{P}_c$. In Fig. 4.4 we take as a study case the *long jump* category of the Olympic Sports dataset (cf. Sec. 4.3) and show the $\mathcal{L}$ decreases along iterations. In particular, we show that if $\mathbf{y}$ and θ are optimized jointly, we attain better performance.

4.2.4 Joint Optimization

In our setup, the grouping and similarity learning tasks are mutually dependent on each other. Therefore, we strive to jointly learn a representation ϕ^{θ}, which captures similarity relationships, and an assignment of samples to surrogate classes $\mathbf{y}$. A natural way to model such dependence in variables is to use a Recurrent Neural Network (RNN) [194]. In particular, RNNs have shown a great potential to model relationships on sequential problems, where each prediction depends on previous observations. Inspired by this insight, we employ a recurrent optimization technique. Following the standard process for learning RNNs we jointly learn

[2] In our experiments we successfully scaled the output space to 20K surrogate classes.

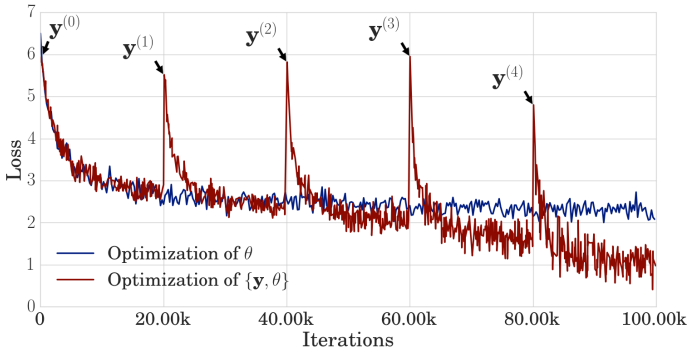


Figure 4.4: Loss value $\mathcal{L}$ for long jump category over each unrolling step. Evidently, the model benefits from jointly optimizing $\{\mathbf{y}, \theta\}$.

$\{\mathbf{y}, \theta\}$ by unrolling the optimization into steps. At time step m we update $\mathbf{y}$ and θ as follows:

$$\begin{aligned} \mathbf{y}^{(m)} &= \arg\max_{\mathbf{y}} \mathcal{G}(X; \phi^{\theta^{(m-1)}}, \mathbf{y}^{(m-1)}) \\ \text{s.t.} \sum_{i:y_i=c}^{n} 1 &> t, \forall_{c \in \{0,\dots,C-1\}}, \end{aligned} \tag{4.5}$$

$$\theta^{(m)} = \arg\min_{\theta} \mathcal{L}(X, \mathbf{y}^{(m)}, \mathbf{R}^{(m)}; \theta^{(m-1)}). \tag{4.6}$$

Where $\mathcal{G}$ is a cost function of pairwise clustering that favors compactness based on sample similarities, which are entailed by the representation $\phi^{\theta^{(m-1)}}$, and t is a lower bound on the number of samples of each cluster.

$$\mathcal{G}(X; \phi^{\theta}, \mathbf{y}) = \sum_{c=0}^{C-1} \sum_{i:y_i=c}^{n} \frac{\sum_{j:y_j=c}^{n} \exp(-\|\phi^{\theta}(x_i) - \phi^{\theta}(x_j)\|_2)}{\left(\sum_{j:y_j=c}^{n} 1\right)^2}. \tag{4.7}$$

In order to avoid the trivial solution of assigning a single sample to each cluster, we initialize $\mathbf{y}^{(0)}$ with the grouping introduced in Sect. 4.2.1 using HOG-LDA as our initial ϕ. In our implementation, $\mathbf{y}$ follows a relaxed one-hot encoding, which can be interpreted as an affinity of samples to clusters. Then, Eq. (4.5) becomes differentiable and is optimized using Stochastic gradient descent (SGD). Subsequently, $\mathcal{L}$ learns a deep similarity encoding representation $\phi^{\theta(m)}$ on samples X using assignments $\mathbf{y}^{(m)}$ and partial orders of X with respect to representatives $\mathbf{R}^{(m)}$. In a typical RNN scenario, for each training iteration the RNN is unrolled m steps. However, this would be inefficient in our setup, as the CNN representation ϕ^{θ} is learned using SGD, and thus, requires to be optimized for a large number of

iterations to be reliable, especially at the first unrolled steps. Therefore, at each step m, we find $\theta^{(m)}$ by optimizing Eq. (4.6) for a number of iterations, fixing $\mathbf{y}^{(m)}$ and $\mathbf{R}^{(m)}$. Then, we use $\theta^{(m)}$ to find the optimal $\mathbf{y}^{(m+1)}$ by optimizing $\mathcal{G}$ using SGD. The presented RNN can also be interpreted as block-coordinate descent [300], where the grouping $\mathbf{y}$ is fixed while updating the representation parameters θ and vice versa. The convergence of block coordinate-descent methods has been largely discussed, obtaining guarantees of convergence to a stationary point [13, 268].

4.3 Experiments

In this section, we present a quantitative and qualitative analysis of our poset-based approach on the challenging and diverse scenarios of human pose estimation and object classification. In all our experiments, we adopt the AlexNet architecture [149].

4.3.1 Human Pose Estimation

To evaluate the proposed approach in the context of pose estimation, we consider 3 different datasets, Olympic Sports (OS), Leeds Sports Pose (LSP), and MPII-Pose (MPI). We show that our unsupervised method is valuable for a range of retrieval problems: For OS we evaluate zero-shot retrieval of detailed postures. On LSP, we perform zero-shot and semi-supervised estimation of pose. Finally, on MPII, we evaluate our approach as an initialization for a supervised learning approach for pose estimation. In contrast to other methods that finetune supervised initializations of CNNs, we train our AlexNet [149] architecture from scratch.

Olympic Sports

The Olympic Sports dataset [204] is a compilation of video sequences of different 16 sports competitions, containing more than 110000 frames overall. We use the approach of [71] to compute person bounding boxes and utilize this large dataset to learn a general representation that encodes fine-grained posture similarities. In order to do so, we initially compute 20000 surrogate classes consisting of 8 samples on average. Then, we utilize partially ordered sets of samples not assigned to any surrogate classes. To train our RNN, we use the optimization approach described in Sect. 4.2.4, where the RNN is unrolled on $m = 10$ steps. At each unrolled step, θ is updated during 20000 iterations of error-backpropagation. To evaluate our representation on fine-grained posture retrieval, we utilize the same annotations which we collected in Chapter 3 and which are available online[3] and follow the same evaluation protocol, using these annotations only for testing. We compare our method with CliqueCNN [11] from Chapter 3, the triplet formulation of Shuffle&Learn [196], the tuple approach of Doersch et al. [51], Exemplar-CNN

[3] https://asanakoy.github.io/cliquecnn/

Method	Olympic Sports
HOG-LDA [97]	0.62
Ex-SVM [187]	0.72
Ex-CNN [55]	0.64
AlexNet [149] - ImageNet	0.65
Doersch et al. [51]	0.62
Shuffle&Learn [196]	0.63
CliqueCNN - ImageNet	0.83
Ours - Scratch	0.78
Ours - ImageNet	**0.85**

Table 4.1: Average ROC AUC for each method on Olympic Sports dataset.

[55], AlexNet [149], Exemplar-SVMs [187], and HOG-LDA [97]. For completeness, we also include a version of our model that was initialized with ImageNet model [149]. During training we use as ϕ the *fc7* output representation of AlexNet and compute similarities using cosine distance. We use *Tensorflow* [1] for our implementation. (*i*) For Shuffle& Learn [196], and Doersh et al. [51] methods, we use the models downloaded from their respective project websites. *(ii)* Exemplar-CNN is trained using the best performing parameters reported in [55] and the 64c5-128c5-256c5-512f architecture. Then we use the output of fc4 and compute 4-quadrant max pooling. *(iii)* Exemplar-SVM was trained on the exemplar frames using the HOG descriptor. The samples for hard negative mining come from all categories except the one that an exemplar is from. We performed cross-validation to find an optimal number of negative mining rounds (less than three). The class weights of the linear Support Vector Machine (SVM) were set as $C_1 = 0.5$ and $C_2 = 0.01$. During training of our approach, each image in the training set is augmented by performing random translation, scaling, and rotation to improve invariance with respect to these.

In Tab. 4.1, we show the average area under the Receiver Operating Characteristic curve (ROC AUC) over all categories for the different methods. When compared with the best runner up [11], the proposed approach improves the performance 2% (the method in [11] was pre-trained on ImageNet). This improvement is due to the additional relationships established by posets on samples not assigned to any surrogate class, which [11] ignored during training. In addition, when compared to the state-of-the-art methods that leverage tuples [51] or triplets [196] for training a CNN from scratch, our approach shows 16% higher performance. This is explained by the more detailed similarity relationships encoded in each poset, which in tuple methods the CNN has to learn implicitly.

In addition to the quantitative analysis, we also perform a qualitative evaluation of the similarities learned by the proposed method. In order to do so, we take a sequence from the *long jump* category of Olympic Sports and select two representatives $\{\mathbf{r}_1, \mathbf{r}_r\}$ with a gap of 8 frames between them and show in Fig. 4.5 the poset learned by our approach. The top row shows two representatives of the same sequence highlighted in red and the remaining sub-sequence between them in blue.

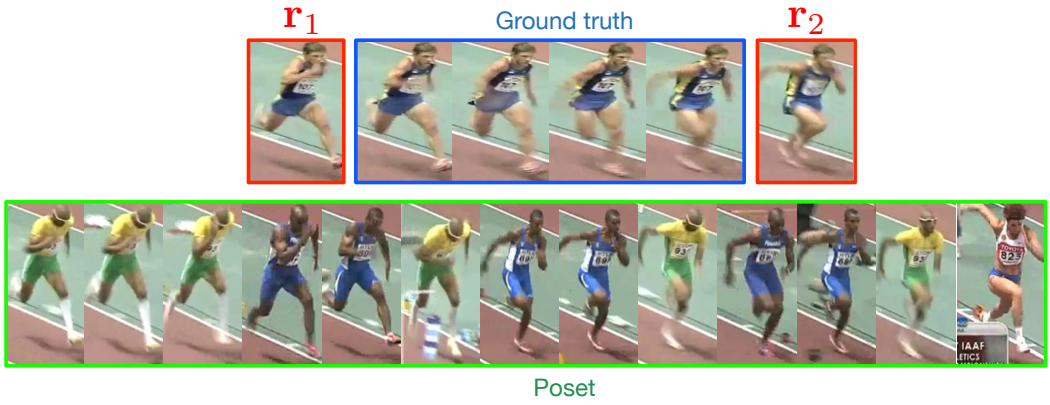


Figure 4.5: Partially ordered set learned by the proposed approach. The top row shows two surrogate class representatives (highlighted in red) of the same sequence and the ground truth sub-sequence between them highlighted in blue. The bottom row shows the predicted poset highlighted in green, successfully capturing fine-grained similarities.

In the bottom row, we present the poset learned by our approach. Since $\mathbf{r}_1$ and $\mathbf{r}_2$ show different parts of a short gait cycle, the similarity relations in the poset should set other frames into perspective and order them. And indeed, we observe that the poset successfully encodes this temporal coherence by ordering frames from other sequences that fit in this gap. This is even more interesting since during training absolutely no temporal structure was introduced in the model, as we were training on only individual frames. These results spurred our interest to also apply the learned posets for video reconstruction using only a few sparse representatives per sequence; additional results can be found in our github repository[4].

Leeds Sports Pose

After evaluating the proposed method for fine-grained posture retrieval, we tackle the problem of zero-shot pose estimation on the LSP dataset. That is, we transfer the pose representation learned on Olympic Sports to the LSP dataset and retrieve similar poses based on their similarity. The LSP [131] dataset is one of the most widely used benchmarks for pose estimation. In order to evaluate our model, we then employ the fine-grained pose representation learned by our approach on OS and transfer it to LSP without doing any further training. For evaluation, we use the representation to compute visual similarities and find the nearest neighbors to a query frame. Since the evaluation is zero-shot, ground-truth joint locations are not available. At test time, we, therefore, estimate the joint coordinates of a query person by finding the most similar frame from the training set and taking its joint coordinates. We then compare our method with AlexNet [149] pre-trained on

[4] https://github.com/asanakoy/deep_unsupervised_posets

Method	Torso	Upper legs	Lower legs	Upper arms	Lower arms	Head	Total
AlexNet [149] - ImageNet	76.9	47.8	41.8	26.7	11.2	42.4	41.1
CliqueCNN - ImageNet	80.1	50.1	45.7	27.2	12.6	45.5	43.5
Ours - ImageNet	**83.5**	**54.0**	**46.8**	**34.1**	**16.8**	**54.3**	**48.3**
Shuffle&Learn [196]	60.4	33.2	28.9	16.8	7.1	33.8	30.0
Ours - Scratch	**67.0**	**38.6**	**34.9**	**20.5**	**9.8**	**35.1**	**34.3**
Ground Truth	93.7	78.8	74.9	58.7	36.4	72.4	69.2
Pose Machines [222]	93.1	83.6	76.8	68.1	42.2	85.4	72.0

Table 4.2: PCP measure for each method on Leeds Sports dataset for zero-shot pose estimation.

ImageNet, the triplet approach of Misra et al. (Shuffle&Learn) [196] and CliqueCNN [11]. In addition, we also report an upper bound on the performance that can be achieved by zero-shot evaluation using ground-truth similarities. Here the most similar pose for a query is given by the frame, which is closest in average distance of ground-truth pose annotations. This is the best one can achieve without a parametric model of pose (the performance gap to 100% shows the discrepancy between poses in the test and the train set). For completeness, we compare with a fully supervised state-of-the-art approach for pose estimation [222]. For computing similarities, we use the same experimental settings as described in Sect. 4.3.1, where ϕ is the representation extracted from *pool5* layer of AlexNet. In Tab. 4.2, we show the PCP@0.5 (Percentage of Correct Parts) obtained by different methods. For a fair comparison with CliqueCNN [11] (which was pre-trained on ImageNet), we include a version of our method trained using ImageNet initialization. Our approach significantly improves the visual similarities learned using both ImageNet-pretrained AlexNet and CliqueCNN [11], obtaining a performance boost of at least 4% in PCP score. In addition, when trained from scratch without any pretraining on ImageNet, our model outperforms the recent triplet model of [196] by 4%, due to the fact that posets are a natural generalization of triplet models, which encode finer relationships between samples. Finally, it is notable that even though our pose representation is *transferred from a different dataset* without finetuning on LSP, it obtains state-of-the-art performance. In Fig. 4.6, we show a qualitative comparison of the part predictions of the supervised approach in [266] trained on LSP, with the heatmaps yielded by our zero-shot approach.

In addition to the zero-shot learning experiments, we also used our pose representation learned on Olympic Sports as initialization for learning the DeepPose method [266] on LSP in a semi-supervised fashion. Our implementation of this method is available on github[5]. To evaluate the validity of our representation, we compare the performance obtained by DeepPose [266], when trained with one of the following models as initialization: random initialization, Shuffle&Learn [196] (triplet model), and our approach trained on Olympic Sports. For completeness, we also compared with ImageNet pre-trained AlexNet [149]. Tab. 4.3 shows the

[5] `https://github.com/asanakoy/deeppose_tf`

Initialization	Torso	Upper legs	Lower legs	Upper arms	Lower arms	Head	Total
Random	87.3	52.3	35.4	25.4	7.6	44.0	42.0
Shuffle&Learn [196]	**90.4**	**62.7**	45.7	33.3	11.8	52.0	49.3
Ours - Scratch	89.7	62.1	**48.2**	**36.0**	**16.0**	**54.2**	**51.0**
AlexNet [149] - ImageNet	92.8	68.1	53.0	39.8	17.5	62.8	55.7

Table 4.3: PCP measure for the DeepPose [266] on Leeds Sports dataset trained using different methods as initialization. "Ours - Scratch" is the network which was trained from scratch on Olympic Sports dataset.

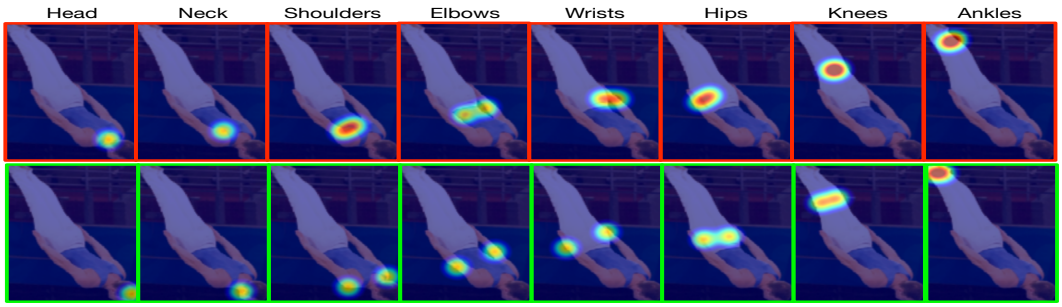


Figure 4.6: Top row: Heatmaps obtained by DeepPose (stg-1) [266] trained on LSP, highlighted in red. Bottom row: Heatmaps obtained by our zero-shot unsupervised approach, highlighted in green.

PCP@0.5 obtained by training DeepPose (stg-1) using their best reported parameters. The obtained results show that our representation successfully encodes pose information, obtaining a performance boost of 9% when compared with a random initialization (that our model starts from), since we learn general pose features that act as a regularizer during training. A note-worthy comparison is that the difference between utilizing ImageNet pretraining, which uses 1.2 million labeled images, and our unsupervised learning approach is just 5%.

MPII Pose

We now evaluate our approach in the challenging MPII Pose dataset [6] which is a state of the art benchmark for evaluation of articulated human pose estimation. The dataset includes around 25K images containing over 40K people with annotated body joints. MPII Pose is a particularly challenging dataset because of the clutter, occlusion, and number of persons appearing in images. To evaluate our approach in MPII Pose, we follow the semi-supervised training protocol used for LSP and compare the performance obtained by DeepPose [266], when trained using as initialization each of the following models: Random initialization, Shuffle&Learn [196] (triplet model) and our approach trained on OS. For completion, we also evaluate ImageNet pre-trained AlexNet [149] as initialization. Following the standard evaluation metric on MPII dataset, Tab. 4.4 shows the PCKh@0.5

Initialization	Head	Neck	LR Shoulder	LR Elbow	LR Wrist	LR Hip	LR Knee	LR Ankle	Thorax	Pelvis	Total
Random	79.5	87.1	71.6	52.1	34.6	64.1	58.3	51.2	85.5	70.1	65.4
Shuffle&Learn [196]	75.8	86.3	75.0	59.2	42.2	73.3	63.1	51.7	87.1	79.5	69.3
Ours	**83.8**	**90.9**	**77.5**	**60.8**	**44.4**	**74.6**	**65.4**	**57.4**	**90.5**	**81.3**	**72.7**
AlexNet - ImageNet	87.2	93.2	85.2	69.6	52.0	81.3	69.7	62.0	93.4	86.6	78.0

Table 4.4: PCKh@0.5 measure for different limbs on MPII Pose benchmark dataset using different initializations for the DeepPose approach [266].

obtained by training DeepPose (stg-1) using their best reported parameters with the different initializations.

The performance obtained on MPII Pose benchmark shows that our unsupervised representation successfully scales to challenging datasets, successfully dealing with clutter, occlusions, and multiple persons. In particular, when comparing our unsupervised initialization with a random initialization, we obtain a 7% performance boost, which indicates that our features encode a robust notion of pose that is robust to the clutter present in MPII dataset. Furthermore, we obtain a 3% improvement over the Shuffle&Learn [196] approach, due to the finer-grained relationships encoded by posets. Finally, it is important to note that the difference between utilizing ImageNet-pretrained AlexNet[149] and our unsupervised learning approach is just 5%.

4.3.2 Object Classification on Pascal VOC

To evaluate the general applicability of our approach, let us now switch from human pose estimation to the challenging, diverse problem of object classification. We classify object bounding boxes of the Pascal VOC 2007 [67] dataset in zero-shot fashion by predicting the most similar images to a query. The object representation needed for computing similarities, we obtain without supervision information, using visual similarities of the triplet model of Wang et al. [285] as initialization. Neither this initialization nor our method apply pretraining or finetuning on ImageNet or Pascal VOC. Using this initialization, we then compute an initial clustering on 1000 surrogate classes with 8 samples on average, on the training set images. We then utilize partially ordered sets of samples not assigned to any class, and jointly optimize assignments and representation using the recurrent optimization approach describe in Sect. 4.2.4. We use the *fc6* layer as the representation ϕ to compute similarities on the Pascal datasets for every CNN method that we now compare. We compare our approach with HOG-LDA [97], the triplet approach of [285], CliqueCNN [11] (from Chapter 3), AlexNet [149] pre-trained on ImageNet, and R-CNN [85] which is pretrained on ImageNet and finetuned in a supervised fashion on Pascal VOC. In Tab. 4.5, we show the classification accuracy for all methods using K-nearest neighbors classifier with $k = 5$ (for $k > 5$ there was only insignificant performance improvement). Our approach improves upon the initial representation of the unsupervised triplet approach of [285] to yield a performance

gain of 6% without requiring any supervision information or finetuning on Pascal. Note that the method of Wang at al. [285] cannot be trained directly on Pascal dataset as it requires video training data.

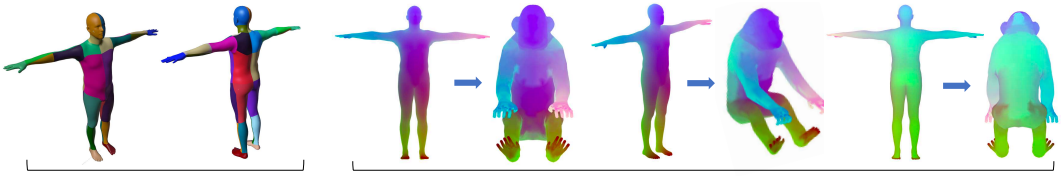
HOG-LDA	Wang et al. [285]	CliqueCNN[11]	Wang et al. [285] + **Ours**	AlexNet [149]	R-CNN [85]
0.1180	0.4501	0.4812	0.5101	0.6160	0.6825

Table 4.5: K-nearest neighbors classification results on Pascal VOC 2007 using the visual representations learned by different methods.

4.4 Conclusions

We have presented an unsupervised approach for representation learning based on CNNs by framing it as a combination of surrogate classification tasks and poset ordering. This generalizes the widely used tuple and triplet losses to establish relations between large numbers of samples. Representation learning then becomes a joint optimization problem of grouping samples into surrogate classes while learning the deep representation encoding image similarities. In the experimental evaluation, the proposed approach has shown competitive performance when compared to state-of-the-art results. The learned representations encode fine-grained image similarity relationships in the context of human pose estimation and object classification and generalize to novel datasets.

5 Self-Training for Transferring Dense Pose to Proximal Animal Classes[1]



(a) 3D model charting

(b) established dense mapping *human* ⇔ *chimpanzee*

Figure 5.1: 3D shape re-mapping from the SMPL [176] model for humans to new object categories (chimps). Manually defined semantic charting (a) on both models is used to establish dense correspondences (b) based on continuous semantic descriptors

In the previous chapters, we explored supervised and self-supervised representation learning approaches. This chapter will combine supervised pretraining and self-supervised training into a self-training approach to adapt the network representation to solving a novel task where no ground-truth annotations are available for training. Recent contributions have demonstrated that it is possible to recognize the pose of humans densely and accurately. In principle, the same approach could be extended to any animal class, but the effort required for collecting new annotations for each case makes this strategy impractical, despite important applications in nature conservation, science and business. Therefore, we show that, at least for proximal animal classes such as chimpanzees, it is possible to transfer the knowledge existing in dense pose recognition for humans, as well as in more general object detectors and segmenters, to the problem of dense pose recognition in other classes.

In the past few years, computer vision has made significant progress in human pose recognition. Deep networks can effectively detect and segment humans [100], localize their sparse 2D keypoints [202], lift these 2D keypoints to 3D [208], and even fit complex 3D models such as Skinned Multi-Person Linear Model (SMPL) [135,

[1]This chapter is based on joint work [237] with Vasil Khalidov, Maureen S. McCarthy, Andrea Vedaldi, and Natalia Neverova, originally presented at CVPR 2020. References to prior work (such as "existing approaches", "recent methods", or "state-of-the-art methods") should be read with this context in mind.

136], all from a single picture or video. DensePose [92] has shown that it is even possible to estimate a dense parameterization of pose by mapping individual image pixels to a canonical embedding space for the human body.

Such advances have been made possible by the introduction of large human pose datasets manually annotated with sparse or dense 2D keypoints, or even in 3D by means of capture systems such as domes. For example, the DensePose-COCO dataset [92] contains 50K COCO images manually annotated with more than 5 millions human body points. Clearly, collecting such data is very tedious, but is amply justified by the importance of human understanding in applications. However, the natural world contains much more than just people. For example, as of today scientists have identified 6,495 species of mammals, 60K vertebrates and 1.2M invertebrates [124]. The methods that have been developed for human understanding could likely be applied to most of these animals as well, provided that one is willing to incur the data annotation burden. Unfortunately, while the applications of animal pose recognition in conservation, natural sciences, and business are numerous, just learning about one more animal may be difficult to justify economically, let alone learning about *all* animals.

Yet, there is little reason to believe that these challenges are intrinsic. Humans can understand the pose of most animals almost immediately, with good accuracy, and without requiring any data annotations at all. Furthermore, images and videos of animals are abundant, so the bottleneck is the inability of machines to learn without external supervision.

In this chapter, we thus consider the problem of learning to recognize the pose of animals with as little supervision as possible. However, rather than starting from scratch, we want to make use of the rich annotations that are *already* available for several animals, and humans in particular. Thus, we focus on the problem of taking the existing annotated data as well as additional unlabelled images and videos of a target animal species and learn to recognize the pose of the latter. Furthermore, for this study we restrict our attention to an animal species that is reasonably close to the available annotations, and elect to focus on the particular example of chimpanzees due to their evolutionary closeness to humans.[2] However, the findings in this chapter are likely to generalize to many other classes as well.

We make several contributions in this work. First, we introduce a dataset for chimpanzees, *DensePose-Chimps*, labelled in the DensePose fashion, which we mostly use to assess quantitatively the performance of our methods. We carefully design the canonical mapping for chimpanzees to be compatible with the one for humans in the original DensePose-COCO, in the sense that points in the two animal models are in as close a correspondence as possible. This is important to be able to transfer dense pose recognition results from humans to chimpanzees while being able to asses the quality of the obtained results.

[2]The idea is to eventually extend pose recognition to more and more animal species, in an incremental fashion.

Second, we study in detail several strategies to transfer existing animal detectors, segmenters, and dense pose extractors from the available annotated data to chimpanzees. In particular, while dense pose annotations exist only for humans, bounding box and mask annotations have been collected for several other object categories as well. As a representative source dataset we thus consider COCO and we investigate how the different COCO classes can be combined to train an object detector and segmenter that transfers optimally to chimpanzees. Surprisingly, we find that transfer from humans alone is not optimal, nor human is the best class for training a model for chimpanzees. In addition to the DensePose-Chimps data, we collect human annotations for instance masks on the *Chimp&See*[3] videos of chimpanzees captured with camera traps in the wild to evaluate the detection performance in the most challenging conditions (with severe occlusions, low visibility and motion blur).

Finally, we propose a framework for augmenting and adapting the human DensePose datasets to new species by self-supervision and pseudo-labeling with zero ground truth annotations on the target class. [4]

5.1 Related work

Human pose recognition. There is abundant work on the recognition of human body pose, both in 2D and in 3D. Given that our focus is 2D pose recognition, we discuss primarily the first class of methods. 2D human pose recognition has flourished by the introduction of deep neural networks [26, 202, 290] trained on large manually-annotated datasets of images and videos such as COCO [168], MPII [6], Leeds Sports Pose Dataset (LSP) [132, 133], PennAction [317] and Posetrack [5]. Furthermore, Dense Pose [92] has introduced a dataset with dense surface point annotations, mapping images to a UV representation of a parametric 3D human model (SMPL) [176].

While all such approaches are strongly-supervised, there are also methods that attempt to learn pose in a completely unsupervised manner [18, 177, 261, 262, 263, 318], including approaches presented in Chapters 3 and 4. Unfortunately, this technology is not sufficiently mature to compete with strong supervision in the wild.

Animal pose recognition. Also related to our work, several authors have learned visual models of animals for the purpose of detection, segmentation, and pose recognition. Some animals are included in almost all general-purpose 2D visual recognition datasets, and in COCO in particular. Hence, all recent detectors and segmenters have been tested on at least a few animal classes.

[3] Some of these videos are available at `http://www.zooniverse.org/projects/sassydumbledore/chimp-and-see`.

[4] Project page: https://asanakoy.github.io/densepose-evolution

For pose recognition, however, the existing body of research is more restricted. Some recent papers have focused on designing pose estimation systems and benchmarks for particular animal species such as Amur tigers [162], cheetahs [199] or drosophila melanogaster flies [93]. There have been a number of large efforts on designing annotation tools for animals, such as DeepLabCut [190] and Anipose [137]. These tools also provide functionality for lifting 2D keypoints to 3D by using multiple views and triangulation. A more detailed overview on applying computer vision and machine learning methodology in neuroscience and zoology is given in [191]. One of the main challenges in this field remains the narrow focus of existing research on specific kinds of animals and particular environments.

There have been few works focusing on the problem of animal understanding from visual data alone and in a more systematic way. This includes the estimation of facial landmarks through domain adaptation [223, 306], and very recently full body pose estimation [25] of four-legged animals by combining large-scale human datasets with a smaller number of animal annotations in a cross-domain adaptation framework. Finally, a line of work from Zuffi et al. [328, 329, 330] is exploring the problem of model-based 3D pose and shape estimation for animal classes. Their research is based on parametric linear model, Skinned Multi-Animal Linear (SMAL), obtained from 3D scans of toy animals and having the capacity to represent multiple classes of mammals. SMAL is the animal analogous of the popular SMLP [176] model for humans. It has since been used in other publications [17] for 3D animal reconstruction, but these methods may still be insufficiently robust for deployment in the wild.

Unsupervised and less supervised pose recognition. Recent methods such as [125, 177, 261, 262, 263, 318] learn sparse and dense object landmarks for simple classes without making use of any annotation, but are too fragile to be used in our application. Also relevant to our work, Slim DensePose [201] looked at reducing the number of annotations required to learn a good DensePose model for humans.

Self-training for dense prediction. Recent studies [299, 305] have demonstrated effectiveness of self-training on the task of image classification when scaled to large amounts of unlabeled data. Pseudo-labeling by averaging predictions from multiple transformed versions of unlabeled samples has been shown effective for keypoint estimation [221]. However, there has been very little research on self-training in the context of dense prediction tasks. Our recent work [9] explored the idea of self-training for segmentation of seismic images and showed promising results on this task for the first time, however, it is out of the scope of this dissertation. We presented more exhaustive discussion of other self-training methods in Chapter 1.1.3.

5.2 Method

We wish to develop a methodology to learn Dense Pose models for new classes with minimal annotation effort. Existing labelled datasets for object detection, segmentation and pose estimation, provide a significant source of supervision that can be harnessed for this task. For detection and segmentation, COCO provide extensive annotations for a variety of object classes, including several animals. For pose recognition, however, the available supervision is generally limited to humans, with a few exceptions. Furthermore, for *dense* pose recognition only human datasets are available — the best example of which is DensePose-COCO [92].

In this work, we raise a number of questions most critical for this setup, namely:

- defining learning and evaluation protocols on new animal categories allowing for training class-specific or class-agnostic DensePose models on a variety of species in a unified way (described in Sect. 5.2.1);
- improving quality of DensePose models and their robustness to unseen data distributions at test time (discussed in Sect. 5.2.2 and 5.2.3);
- optimally combining the existing variety of data sources in order to initialize a detection model for a new animal species (discussed in Sect. 5.2.4);
- defining strategies for mining dense pseudo-labels for gradual domain adaptation from humans to chimpanzees in a teacher-student setting (discussed in Sect. 5.2.5).

5.2.1 Annotation through 3D shape re-mapping

While our aim is to learn to reconstruct the dense pose of chimpanzees with zero supervision, a manually-annotated dataset for this class is required for evaluation. Here, we explain how to collect DensePose annotations for a new category, such as chimpanzees.

Dense Pose model. Recall that DensePose-COCO contains images of people collected 'in the wild' and annotated with dense correspondences. These dense keypoints are identified as the point $p \in S$ of a reference 3D model $S \subset \mathbb{R}^3$ of the object.[5] Furthermore, the keypoints $p \in S$ are indexed by triplets $(c, u, v) \in \{1, \ldots, C\} \times [0, 1]^2$ where c is the *chart index*, corresponding to one of C model parts, and (u, v) are the coordinates within a chart. The DensePose-COCO dataset [92] contains bounding boxes, pixel-perfect foreground-background and part segmentations, and (c, u, v) annotations for a large number of foreground pixels.

Dense Pose for chimps. We wish to extend the DensePose annotations to the chimpanzee class. In order to do so, we rely on a separate artist-created 3D model[6]

[5] Dense Pose uses SMPL [176] to define S due to its popularity

[6] Purchased from http://hum3d.com/

of a chimpanzee as a reference for annotators to collect labels for the chimpanzee images (instead of the human model used by the original DensePose).

For each object, we use Amazon Mechanical Turk to collect the object bounding boxes, followed by pixel-perfect foreground/background segmentation masks, and finally the (c, u, v) chart coordinates for a certain number of pixels randomly sampled from the foreground regions. Differently from the original DensePose, we *do not* also collect dense annotations for the body parts as the latter was found to be very challenging for the annotators. Note however that the chart index c reveals the part identity for each of the annotated image pixels.

Semantic alignment. Finally, we wish to align the human and chimpanzee DensePose models by mapping the collected annotations back on the surface of the SMPL model using the mesh re-mapping strategy described below. The latter step unifies the evaluation protocols across different object categories and allows to transfer knowledge and annotations between different species.

In spite of the fact that humans and most mammals share topology and the skeletal structure, establishing precise semantic dense correspondences between the 3D models of humans and different animal species is challenging due to differences in body proportions and local geometry.

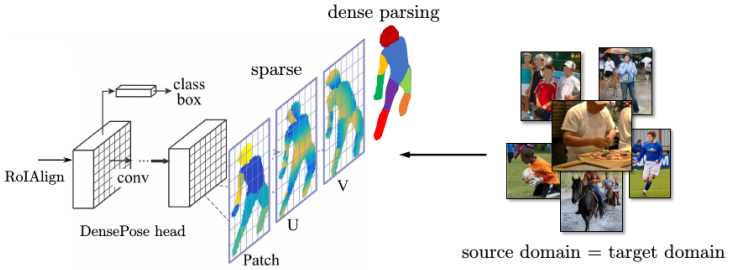
As preprocessing, we manually charted the SMPL and the chimp meshes into $L = 32$ semantically-corresponding parts to guide the mapping. Then, for each vertex p of each mesh S, we extracted an adapted version of the continuous semantic descriptor $\mathbf{d}(p)$ proposed by Léon et al. [158]:

$$\mathbf{d}(p) = (d_\ell(p))_{\ell=1}^{L}, \quad d_\ell(p) = \frac{1}{|S_\ell|} \sum_{s \in S_\ell} g(p, s; S_\ell) \tag{5.1}$$

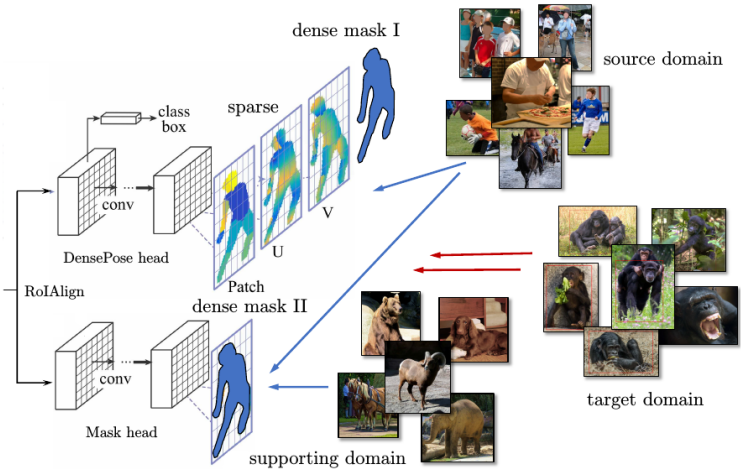
where $S_\ell \subset S$ is the set of all vertices in part ℓ of the mesh and $g(p, s)$ is the geodesic distance between two points on S.[7] With this, the mapping from the human mesh S to the chimp mesh S' is obtained by matching nearest descriptors: $S \to S'$, $p \mapsto \text{argmin}_{q \in S'} \|\mathbf{d}_S(p) - \mathbf{d}_{S'}(q)\|^2$.

This simple approach yields satisfactory results both in terms of alignment and smoothness, as shown in Fig. 5.1. It does not require any optimization in 3D space based on model fitting or mesh deformation and works on meshes of arbitrary resolutions. Interestingly. exploiting information about mesh geometry (such as high dimensional SHOT [235] descriptors or their learned variants [96]) instead or in addition to semantic features results in noisy mappings. This can likely be attributed to prominent inconsistencies in local geometry of some body regions between the object categories.

[7] To partially compensate for differences in proportions across different categories, we further normalized the descriptors by their part average: $d_\ell(p) \leftarrow d_\ell(p) / \langle d_\ell(q) \rangle_{q \in S_\ell}$.



(a) original DensePose-RCNN, human class (single domain)



(b) domain adaptation for the new category by self-training

Figure 5.2: Comparison of the original (a) and our (b) DensePose learning architecture. See Sect. 5.2.2 for detailed description of the architecture.

5.2.2 Multi-head R-CNN

Our goal is to develop a DensePose predictor for a new class. Such a predictor must detect the object via a bounding box, segment it from the background, and obtain the DensePose chart and *uv*-map coordinates for each foreground pixel. We implement this with a *single model* with multiple heads, performing the various tasks on top of the same trunk and shared image features (Fig.5.2.b).

The base model is R-CNN [100] modified to include the following heads. The first head refines the coordinates of the bounding box. The second head computes a foreground-background segmentation mask in the same way as Mask R-CNN. The third and the final head computes a part segmentation mask I, assigning each pixel to one of the 24 Dense Pose charts, and the uv map values for each foreground pixel.

Class-agnostic model. Compared to the standard Mask R-CNN, our model is *class agnostic*, i.e. trained for only one class type. This is true also when we make use

of a Mask R-CNN pre-trained on multiple source classes as the goal is always to only build a model for the final target chimpanzee class — we found that merging classes is an effective way of integrating information.

Heterogeneous training. Our training data can be heterogeneous. In particular, COCO provides segmentation masks for 80 categories, but DensePose-COCO provides DensePose annotations only for humans. While we train a single class-agnostic model, the Dense Pose head is trained only for the class human for which the necessary ground-truth data is available.

Note in particular that both the Mask R-CNN head and the DensePose head contain a foreground-background segmentation component — these are not equivalent, as the DensePose one is only valid (and trainable) for humans, while the Mask R-CNN one is generic (and trainable from all COCO classes). We will see in the experiments that their combination improves performance.

Fine-tuning. As shown later, for fine-tuning the model we generate pseudo-label on chimpanzees imagery. The pseudo-labels are generated for all components of the model (segmentations, *uv* maps), including in particular both foreground-background segmentation heads.

Other architectural improvements. Our model (Fig. 5.2.a) has a few mode differences compared to the original Dense Pose (Fig. 5.2.b) which we found useful to improved accuracy and/or data collection efficiency.

First, both the original and our implementations use dense (pixel-perfect) supervision for the foreground-background masks. However, in our version we *do not* use the pixel-perfect part segmentations in the original DensePose annotations — the part prediction head is trained only from the chart labels for the pixels that are annotated in the data. This is another reason why we do not collect pixel-perfect segmentations for the chimpanzee images.

We further improve the DensePose head by implementing it using Panoptic Feature Pyramid Networks [146], and use a configuration similar to DeepLab [32] that benefits from higher resolution.

5.2.3 Auto-calibrated R-CNN

As suggested above, pseudo-labelling can be used to fine-tune a pre-trained model on imagery containing the target class, chimpanzees in our case. The idea is to use a model pre-trained on a different class or set of classes to generate labels in the new domain, and then to retrain the model to fit those labels. Due to the domain gap, however, the pseudo-labels are somewhat unreliable. In this section, following [142] we develop a principled manner to let the neural network itself produce a *calibrated measure of uncertainty* which we can use to rank pseudo-labels by reliability.

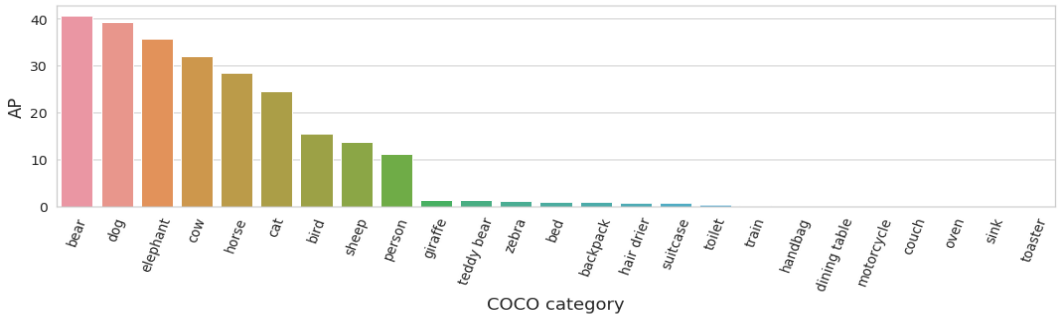


Figure 5.3: Instance Segmentation score (AP) on DensePose-Chimps for Mask R-CNN models trained using different COCO categories, ranked by decreasing performance.

Classification uncertainty. Our model performs categorical classification for two purposes: to associate a class label to a bounding box, and to classify individual pixels as background, foreground, or as one of the body parts. In order to estimate the uncertainty for these categorical predictions, we adopt the *temperature scaling* technique of [110].

Thus let z_y be the score that the neural network associates to hypothesis $y \in \{1, \ldots, K\}$ for a given input sample. We extend the network to compute an additional per-sample scalar $\alpha \geq 0$. With this scalar, the posterior probability of hypothesis y is given by the *scaled softmax*

$$\hat{\sigma}(y; z, \alpha) = \frac{\exp(\alpha z_y)}{\sum_{k=1}^{K} \exp(\alpha z_k)} \tag{5.2}$$

We can interpret the coefficient $\alpha = 1/T$ as an inverse temperature. A small α means that the model is fairly certain about the prediction, whereas a large α that it is not.

Note that, since α is also estimated by the neural network, we require a mechanism to learn it. This is in fact obtained automatically [110, 200] by simply minimizing the negative log-likelihood of the model, also known in this case as cross-entropy loss: $\ell(y, z, \alpha) = -\log \hat{\sigma}(y; z, \alpha)$.

Regression with uncertainty. Our model performs regression to refine the bounding box proposals (for four scalar outputs, two for each of the two corners of the box) and to obtain the DensePose uv-coordinates (for two scalar outputs for each image pixel in a proposal).

Thus let $y \in \mathbb{R}^D$ be the vector emitted by one of the regression heads (where D depends on the head). Similarly to the classification case, we use the network to also predict an *uncertainty score* $\sigma \in \mathbb{R}^D$. This time, however, we have a different scalar for each element in y (hence, for the uv-maps, we have two uncertainty scores for each pixel, which we can visualize as an image). The vector σ is interpreted as the diagonal variance of the regressed vector y, assuming the latter to have a

Gaussian distribution. The uncertainty scores σ can thus be trained jointly with the predictor $\hat{y}$ by minimizing the negative log-likelihood of the model:

$$\ell(y, \hat{y}, \sigma) = \frac{D}{2}\log 2\pi + \frac{1}{2}\sum_{i=1}^{D}\left(\log \sigma_i^2 + \frac{(\hat{y}_i - y_i)^2}{\sigma_i^2}\right) \tag{5.3}$$

For a fixed error $|\hat{y}_i - y_i|$, the quantity above is minimized by setting $\sigma_i = |\hat{y}_i - y_i|$ — hence the model is encouraged to guess the magnitude of its own prediction error. However, if $|\hat{y}_i - y_i| = 0$, the quantity above diverges to $-\infty$ for $\sigma_i \to 0$. Hence, we clamp σ_i from below to a minimum value $\sigma_{\text{min}} > 0$.

model	**AP**	**AP**$_{50}$	**AP**$_{75}$
DensePose-RCNN	50.88	80.40	54.80
DensePose-RCNN*	51.44	81.44	55.12
DensePose-RCNN* (σ)	54.13	82.32	58.06

model	**AP**	**AP**$_{50}$	**AP**$_{75}$
DensePose-RCNN	43.84	76.88	45.84
DensePose-RCNN*	43.84	77.52	45.60
DensePose-RCNN* (σ)	45.58	78.79	47.93

Table 5.1: Detection (left) and instance segmentation (right) performance on DensePose-COCO `minival`.

model	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_M$	**AP**$_L$	**AR**	**AR**$_{50}$	**AR**$_{75}$	**AR**$_M$	**AR**$_L$
DensePose-RCNN	46.8	84.5	47.7	41.8	48.0	54.7	89.5	58.9	43.3	55.5
DensePose-RCNN*	47.2	85.8	47.3	42.5	48.4	55.2	91.0	59.1	44.0	55.9
DensePose-RCNN* (σ)	53.2	88.3	57.0	48.6	54.6	61.2	92.4	67.2	50.0	61.9

Table 5.2: DensePose performance on DensePose-COCO `minival`. * denotes our improved architecture; (σ) denotes the proposed Auto-calibrated version of the network.

Details. For both classification and regression models, the uncertainties α and σ must be positive — in the network, they are obtained via a `softplus` activation.

5.2.4 Optimal transfer support

In this section, we investigate which object categories in the COCO dataset provide the best support for recognizing a new animal species, chimpanzees in our case. Among the animals in COCO, chimpanzees are most obviously related to humans, and we may thus expect that people may be the most transferable class. However, despite their overall structural similarity, people's appearance is fairly different, also due to the lack of fur and the presence of clothing. Furthermore, context is also often quite different. It is thus unclear if a deep network trained to recognise humans can transfer well at all on chimpanzees, or whether other object categories might do better.

Class selection. We test what is more important: biological proximity of the species (as a proxy to morphological similarity) or appearance similarity (as a combination

of typical poses and textures). We also search for a brute force solution for this particular dataset to back up or disprove our intuition for class selection. In our experiments, we have tested the following selections:

- *person* class only (due to morphological similarity).
- *animal* classes only (due to higher pose and texture similarity): *bear*, *dog*, *elephant*, *cat*, *horse*, *cow*, *bird*, *sheep*, *zebra*, *giraffe*, *mouse*.
- *top-N* scoring classes on the new category (brute force solution). In this setting, we first train a set of C single-class models for each of the $C = 90$ object classes in the COCO dataset and rank them according to their instance segmentation performance on the DensePose-Chimps dataset (see Fig. 5.3). Then for each combination of $S \in \{1, \ldots, C\}$ top scoring classes we train the same network from scratch. The solution that have we found optimal corresponds to $C_{\text{opt}} = 9$, where the top-C scoring classes are: *bear*, *dog*, *elephant*, *cat*, *horse*, *cow*, *bird*, *person*, *sheep*.

As shown in Tab. 5.5, the top-N solution produces similar results compared to combination *person+animals*. *Person* class only is ineffective for training in this setting.

Class fusion. We have also explored the question of class-agnostic vs multi-class training as a trade-off between the number of training samples per class vs granularity of prediction modes. For the task of adapting the new model to a single category (on the given dataset) class-agnostic training showed convincingly stronger results (see Tab. 5.5).

5.2.5 Dense label distillation

Finally, we aim at finding an effective strategy for exploiting unlabeled data for the target domain in the teacher-student training setting and performing *distillation* in dense prediction tasks. In our setting, the *teacher* network trained on the selected classes of the COCO dataset with DensePose is used to generate *pseudo-labels* for fine-tuning the *student* network on the augmented data. The *student* network is initialized with *teacher*'s weights.

Once teacher predictions on unlabeled data are obtained, we start by filtering out low confidence detections using calibrated detection scores. After that, the bounding boxes and segmentation masks on remaining samples are used for augmented training. For mining DensePose supervision, we consider three different dense sampling strategies driven by each of the tasks solved by the teacher network, in addition to uniform sampling:

- **uniform sampling** – all points from the selected detections are sampled with equal probability;

		DensePose-Chimps			Chimp&See	
sampling	k	$\mathbf{AP}_{DPose}$	$\mathbf{AP}_D$	$\mathbf{AP}_S$	$\mathbf{AP}_D$	$\mathbf{AP}_S$
–	–	33.4	62.1	56.4	50.5	43.5
uniform	5	$34.5 \pm .4$	$63.3 \pm .3$	$58.0 \pm .3$	$58.9 \pm .5$	$49.0 \pm .5$
mask-based	5	$34.7 \pm .4$	$63.3 \pm .3$	$58.0 \pm .2$	$58.8 \pm .6$	$49.0 \pm .5$
I-based	5	$\mathbf{34.9 \pm .6}$	$\mathbf{63.4 \pm .3}$	$\mathbf{58.0 \pm .2}$	$\mathbf{59.2 \pm .4}$	$49.2 \pm .5$
uv-based	5	$34.6 \pm .3$	$63.3 \pm .3$	$58.2 \pm .3$	$59.0 \pm .1$	$\mathbf{49.6 \pm .1}$

Table 5.3: AP of the *student* network trained with different sampling strategies. Optimal number of sampled points k per detection is reported for each sampling. The first row corresponds to the *teacher* network. *Mean±std* for 20 runs.

- **coarse classification uncertainty [mask-based]** – sampling top k from ranked calibrated posteriors produced by the mask branch for the task of binary classification;
- **fine classification uncertainty [I-based]** – selection of top k from ranked calibrated posteriors from the 24-way segmentation outputs of the DensePose head;
- **regression uncertainty sampling [uv-based]** – sampling of top k points based on ranked confidences in the uv-outputs of the DensePose head.

In Sect. 5.3 we provide experimental evidence that sampling based on confidence estimates from fine-grained tasks (I-estimation, uv-maps) results in the best *student* performance.

5.3 Experiments

We now describe the results of empirical evaluation and provide detailed descriptions of ablation studies.

5.3.1 Datasets

We use a combination of human and animal datasets with different kinds of annotations or no annotations at all. A brief description of each of them is provided below.

DensePose-COCO dataset [92]. This is the dataset for human dense pose estimation, that we use for training the teacher model. It contains 50k annotated instances totalling to more than 5 million ground truth correspondences. We also augment the teacher training with other object categories from the original **COCO dataset** [168].

	DensePose-Chimps			Chimp&See	
k	$\mathbf{AP}_{DensePose}$	$\mathbf{AP}_D$	$\mathbf{AP}_S$	$\mathbf{AP}_D$	$\mathbf{AP}_S$
0	$33.8 \pm .2$	$63.1 \pm .2$	$57.9 \pm .2$	$59.0 \pm .3$	$49.2 \pm .4$
1	$34.7 \pm .5$	$63.0 \pm .2$	$57.9 \pm .3$	$\mathbf{59.3 \pm .3}$	$49.3 \pm .6$
2	$34.6 \pm .6$	$63.4 \pm .3$	$57.9 \pm .3$	$59.2 \pm .4$	$49.3 \pm .4$
5	$\mathbf{34.9 \pm .5}$	$\mathbf{63.4 \pm .3}$	$\mathbf{58.0 \pm .2}$	$59.2 \pm .4$	$49.2 \pm .5$
10	$34.6 \pm .6$	$63.3 \pm .3$	$58.0 \pm .3$	$59.2 \pm .4$	$\mathbf{49.4 \pm .4}$
1000	$33.1 \pm .6$	$63.2 \pm .2$	$57.8 \pm .3$	$59.2 \pm .5$	$49.4 \pm .5$
10000	27.6 ± 4.6	$60.2 \pm .4$	$55.7 \pm .5$	$58.0 \pm .7$	$49.1 \pm .6$

Table 5.4: DensePose, detection and instance segmentation AP of the *student* network trained with *I*-sampling for different number of sampled points k. *Mean±std* for 20 runs.

Chimp&See dataset. For training our models in a self-supervised setting, we used unlabeled videos containing chimpanzees from the *Chimp&See* project[8]. This data is being collected under the umbrella of The Pan African Programme[9]: The Cultured Chimpanzee (PanAf) by installing camera traps in more than 40 natural habitats of chimpanzees on different sites in Africa. In this work, we used a subset of the collected data consisting of 18,556 video clips, from 10 sec to 1 min long each, captured with cameras in either standard or night vision mode depending on lighting conditions. These recordings were motion triggered automatically by passing animals. As a result, some clips may not contain any chimps beyond first several frames.

For evaluation, we chose videos from one site, sampled frames at 1 fps, removed the near duplicates and collected human annotations for instance masks. This resulted into 1054 images containing 1528 annotated instances, that we use to benchmark detection performance in our models. However, due to in-the-wild nature of this data and presence of motion blur, severe occlusions, and low resolution in some cases, we found it infeasible to collect precise human annotations at the level of dense correspondences.

DensePose-Chimps test set. For the task of evaluating DensePose performance on this new category, we collected a set of 662 higher quality images that contain 933 instances of chimpanzees. We annotated this data with bounding boxes, binary masks, body part segmentation and dense pose correspondences as explained in Sect. 5.2.1.

5.3.2 Implementation Details

In this subsection we provide more details on our implementation of the Multi-head R-CNN network. Our codebase and network configuration files are available on

[8] A subset of the videos from the Chimp&See dataset is publicly available at `http://www.zooniverse.org/projects/sassydumbledore/chimp-and-see`.

[9] `http://panafrican.eva.mpg.de`

selected COCO object classes	**AP**	**AP**$_{50}$	**AP**$_{75}$
top-9 classes	57.29	85.63	63.45
bear-only	40.69	70.88	44.23
person-only	9.39	19.32	8.21
animals-only	52.28	80.62	58.60
person + animals	**57.34**	**85.76**	**63.59**
person + animals: class agnostic	57.34	85.76	63.59
person + animals: class specific	50.47	72.85	54.30

Table 5.5: Instance segmentation AP on DensePose-Chimps for Mask R-CNN trained on different subsets of classes.

github[10]. We introduced a number of changes and improvements in the DensePose head of the standard DensePose R-CNN architecture of [92] with ResNet-50 [102] backbone. These changes are listed below for the affected branches; other branches remained unchanged and correspond exactly to the Mask R-CNN architecture of [100].

- We have increased the RoI resolution from 14×14 to 28×28 in the DensePose head, as proposed in [308].
- We have replaced the 8-layer DensePose head with the geometric and context encoding (GCE) module [308], combining a non-local convolutional layer [284] with the atrous spatial pyramid pooling (ASPP) [31].
- We have replaced the original Feature Pyramid Network (FPN) of DensePose R-CNN with a Panoptic FPN [146].

Each of these modifications led to increase in network performance due to improved multi-scale context aggregation. We refer the reader to the work of [308] for ablation studies whose results are aligned well with our own observations.

The *teacher* and the *students* networks share the same architecture. To predict α or σ we simply extend the output layer of the corresponding head by doubling the number of its neurons.

5.3.3 Results

Ablations on architectural choices. First, we compare our model to the original DensePose-RCNN [92] (detectron2 implementation). We also ablate our improvements in the architecture and provide results with and without auto-calibration. Tab. 5.1, 5.2 show consistent improvements on all tasks for both modifications.

Optimal transfer support. We (a) benchmarked every strategy for class selection described in Sect. 5.2.4 and (b) experimented with multi-class and class-agnostic

[10] https://github.com/facebookresearch/detectron2

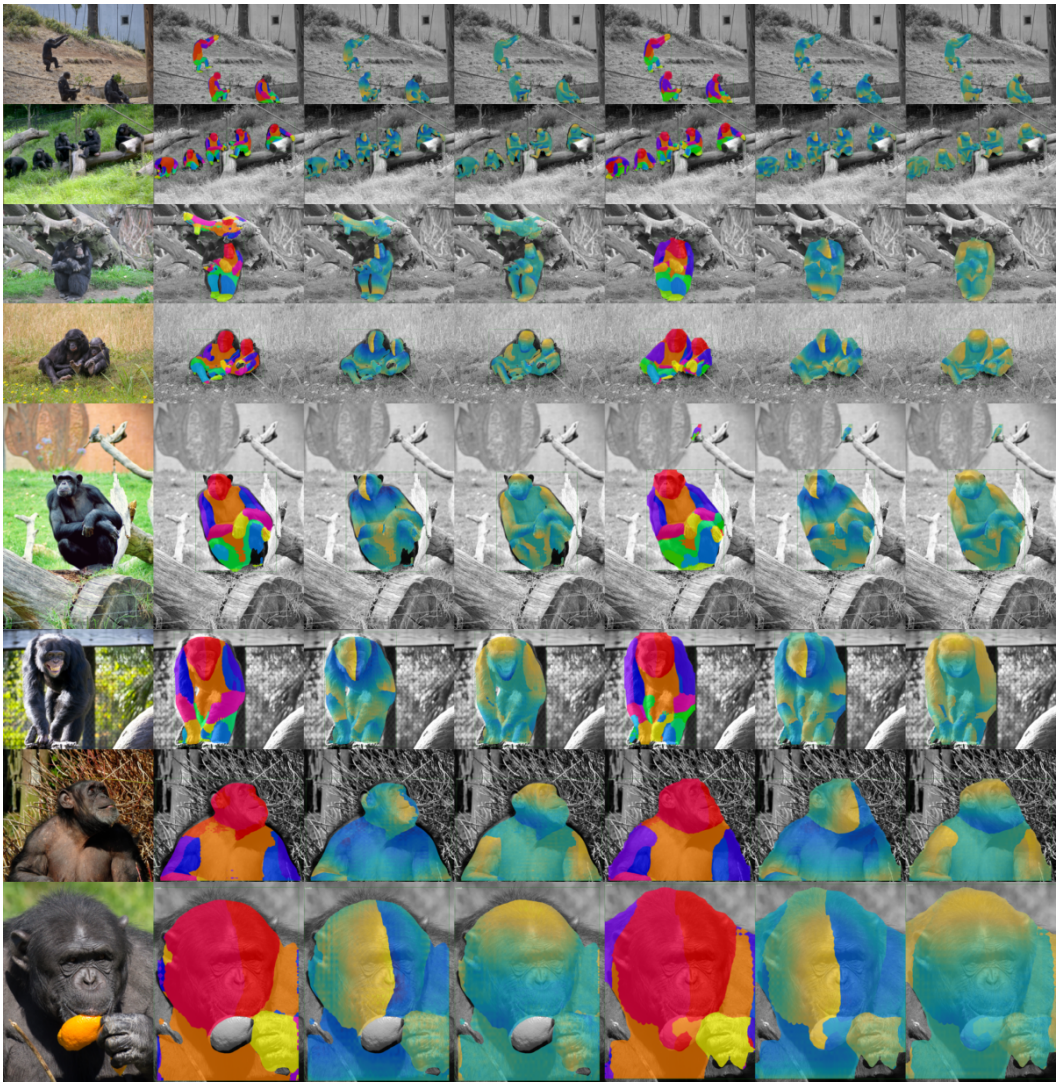


Figure 5.4: Visual results: **(left)** *teacher* network predictions vs (**right**) predictions of *student* network trained using *I*-sampling. The *student* produces more accurate boundaries and *uv*-maps. Zoom-in for details. Image source: [4, 68, 69, 184, 193, 264, 265, 273].

models. From Tab. 5.5 we can see that class agnostic training on the *animals+person* subset shows the best transferability for DensePose-Chimps dataset. Therefore, it was used for training all our DensePose models.

Dense label distillation. We conducted experiments with different sampling strategies and different numbers of sampled points k per detection. In Tab. 5.3 we show performance of the *teacher* (first row) and the *student* networks trained using

different sampling strategies along with the corresponding optimal k. I-based sampling showed most impressive gains, followed by uv-based sampling. Uniform selection produces poor results. In Tab. 5.4 we report performance for different number of sampled points in every detection for I-based sampling.

Computational cost. Our auto-calibrated model has a negligible computational overhead ($< 1\%$) compared to the baseline model. Before training the *student*, sampling of the pseudo-labels requires one forward pass of the *teacher* network over the unlabeled dataset.

Qualitative results Qualitative results for the *teacher* and the *student* networks are shown in Fig. 5.4. In addition, we also point the readers to the video samples[11] from the Chimp&See dataset showing frame-by-frame predictions produced by our model before (*teacher*) and after self-training (*student*). The results produced by the *student* network are generally significantly more stable.

5.4 Conclusions

We have studied the problem of extending dense body pose recognition to animal species and suggested that doing this at scale requires learning from unlabelled data. Encouragingly, we have demonstrated that existing detection, segmentation, and dense pose labelling models can transfer very well to a proximal animal class such as chimpanzee despite significant inter-class differences. We have shown that substantial improvements can be obtained by carefully selecting which categories to use to pre-train the model, by using a class-agnostic architecture to integrate different sources of information, and by modelling labelling uncertainty to grade pseudo-label for self-training. In this manner, we have been able to achieve excellent performance without using a single labelled image of the target class for training.

In the future, we would like to investigate how a limited amount of target supervision can be best used to improve the results, and how other techniques from domain adaptation could also be used for this purpose.

[11] https://asanakoy.github.io/densepose-evolution

6 Style-Aware Content Loss for Real-time High-resolution Style Transfer[1]

A picture may be worth a thousand words, but at least it contains a lot of very diverse information. This not only comprises *what* is portrayed, e.g., composition of a scene and individual objects, but also *how* it is depicted, referring to the artistic style of a painting or filters applied to a photo. Especially when considering artistic images, it becomes evident that not only content but also style is a crucial part of the message an image communicates (just imagine van Gogh's Starry Night in the style of Pop Art). Here, we follow the common wording of our community and refer to 'content' as a synonym for 'subject matter' or 'sujet', preferably used in art history. A vision system then faces the challenge to decompose and separately represent the content and style of an image to enable a direct analysis based on each individually. The ultimate test for this ability is style transfer [79] – exchanging the style of an image while retaining its content.

In contrast to the seminal work of Gatys et al. [79], who have relied on powerful but slow iterative optimization, there has recently been a focus on feed-forward generator networks [59, 117, 130, 164, 269, 270, 286]. The crucial representation in all these approaches has been based on a VGG16 or VGG19 network [249], pre-trained on ImageNet [48]. However, a recent trend in deep learning has been to avoid supervised pre-training on a million images with tediously labeled object bounding

[1] This chapter is based on joint work [238] with Dmytro Kotovenko, Sabine Lang, and Björn Ommer, originally presented at ECCV 2018. References to prior work (such as "existing approaches", "recent methods", or "state-of-the-art methods") should be read with this context in mind.

Figure 6.1: Evaluating the fine details preserved by our approach. Can you guess which of the cut-outs are from Monet's artworks and which are generated? Solution is on p. 108.

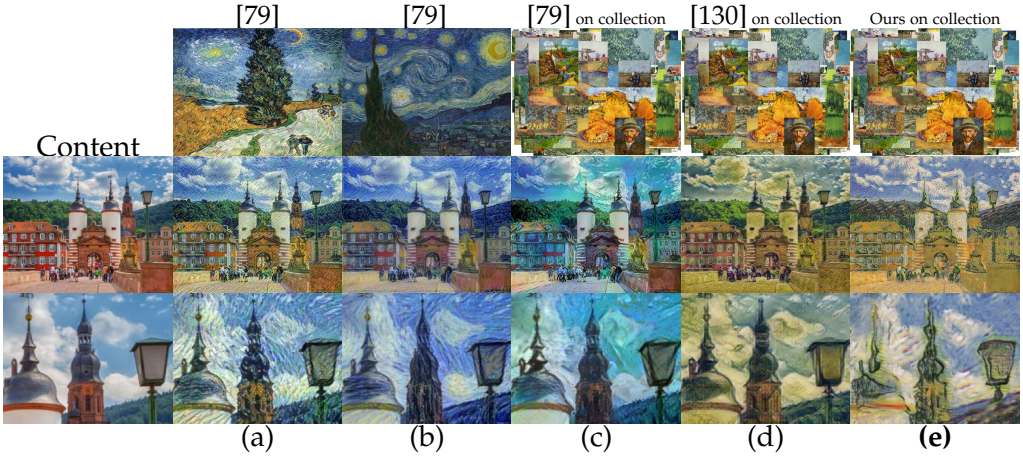


Figure 6.2: Style transfer using different approaches on 1 and a collection of reference style images. (a) [79] using van Gogh's "Road with Cypress and Star" as reference style image; (b) [79] using van Gogh's "Starry night"; (c) [79] using the average Gram matrix computed across the collection of Vincent van Gogh's artworks; (d) [130] trained on the collection of van Gogh's artworks alternating target style images every SGD mini-batch; (e) our approach trained on the same collection of van Gogh's artworks. Stylizations *(a)* and *(b)* depend significantly on the particular style image, but using a collection of the style images *(c)*, *(d)* does not produce visually plausible results, due to oversmoothing over the numerous Gram matrices. In contrast, our approach *(e)* has learned how van Gogh is altering particular content in a specific manner (edges around objects also stylized, cf. bell tower)

boxes [285]. In the setting of style transfer this has the particular benefit of avoiding from the outset any bias introduced by ImageNet, which has been assembled without artistic consideration. Rather than utilizing a separate pre-trained VGG network to measure and optimize the quality of the stylistic output [59, 79, 130, 164, 269, 270, 286], we employ an encoder-decoder architecture with adversarial discriminator, Fig. 6.3, to stylize the input content image and also use the encoder to measure the reconstruction loss. In essence the stylized output image is again run through the encoder and compared with the encoded input content image. Thus, we learn a style-specific content loss from scratch, which adapts to the specific way in which a particular style retains content and is more adaptive than a comparison in the domain of RGB images [326].

Most importantly, however, previous work has only been based on a *single* style image. This stands in stark contrast to art history which understands "style as an expression of a collective spirit" resulting in a "distinctive manner which permits the grouping of works into related categories" [72]. As a result, art history developed a scheme, which allows to identify groups of artworks based on shared qualities. Artistic style consists of a diverse range of elements, such as form, color, brushstroke, or use of light. Therefore, it is insufficient to only use a single artwork,

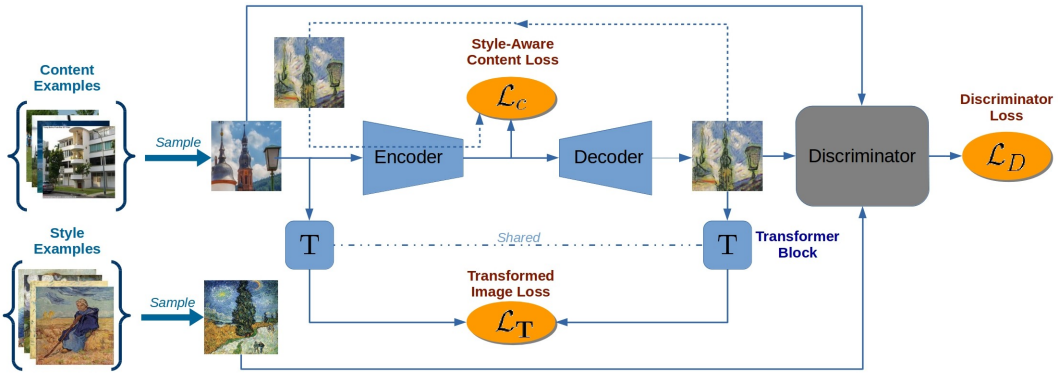


Figure 6.3: Pipeline of our approach. Encoder-decoder network with style-aware content loss, transformed image loss, and discriminator.

because it might not represent the full scope of an artistic style. Today, freely available art datasets such as Wikiart [138] easily contain more than 100K images, thus providing numerous examples for various styles. Previous work [59, 79, 130, 164, 269, 270, 286] has represented style based on the Gram matrix, which captures highly image-specific style statistics, cf. Fig. 6.2. To combine several style images in [59, 79, 130, 164, 269, 270, 286] one needs to aggregate their Gram matrices. We have evaluated several aggregation strategies and averaging worked the best, Fig. 6.2 (c). But, obviously, neither art history, nor statistics suggests aggregating Gram matrices. Additionally, we investigated alternating the target style images in every mini-batch while training [130], Fig. 6.2 (d). However, all these methods cannot make proper use of several style images, because combining the Gram matrices of several images forfeits the details of style, cf. the analysis in Fig. 6.2. In contrast, our proposed approach allows to combine an arbitrary number of instances of a style during training.

We conduct extensive evaluations of the proposed style transfer approach; we quantitatively and qualitatively compare it against numerous baselines. Being able to generate high quality artistic works in high-resolution, our approach produces visually more detailed stylizations than the current state of the art style transfer approaches and yet shows real-time inference speed. The results are quantitatively validated by experts from art history and by introduced in this chapter *deception rate* metric based on a deep neural network for artist classification.

6.1 Related Work

In recent years, a lot of research efforts have been devoted to texture synthesis and style transfer problems. Earlier methods [107] are usually non-parametric and are build upon low-level image features. Inspired by Image Analogies [107], approaches [77, 167, 246, 247] are based on finding dense correspondence between

content and style image and often require image pairs to depict similar content. Therefore, these methods do not scale to the setting of arbitrary content images.

In contrast, Gatys et al. [79, 80] proposed a more flexible iterative optimization approach based on a pre-trained VGG-19 network [249]. This method produces high quality results and works on arbitrary inputs, but is costly, since each optimization step requires a forward and backward pass through the VGG19 network. Subsequent methods [130, 159, 269] aimed to accelerate the optimization procedure [79] by approximating it with feed-forward convolutional neural networks. This way, only one forward pass through the network is required to generate a stylized image. Beyond that, a number of methods have been proposed to address different aspects of style transfer, including quality [35, 81, 128, 286, 294], diversity [163, 270], photorealism [180], combining several styles in a single model [30, 59, 278] and generalizing to previously unseen styles [83, 117, 164, 244]. However, all these methods rely on the fixed style representation which is captured by the features of a VGG [249] network pre-trained on ImageNet. Therefore they require a supervised pre-training on millions of labeled object bounding boxes and have a bias introduced by ImageNet, because it has been assembled without artistic consideration. Moreover, the image quality achieved by the costly optimization in [79] still remains an upper bound for the performance of recent methods. Other works like [11, 43, 65, 188, 293] learn how to discriminate different techniques, styles and contents in the latent space.

Zhu et al. [326] learn a bidirectional mapping between a domain of content images and paintings using generative adversarial networks. Employing cycle consistency loss, they directly measure the distance between a backprojection of the stylized output and the content image in the RGB pixel space. Measuring distances in the RGB image domain is not just generally prone to be coarse, but, especially for abstract styles, a pixel-wise comparison of backwards mapped stylized images is not suited. Then, either content is preserved and the stylized image is not sufficiently abstract, e.g., not altering object boundaries, or the stylized image has a suitable degree of abstractness and so a pixel-based comparison with the content image must fail. Moreover, the more abstract the style is, the more potential backprojections into the content domain exist, because this mapping is underdetermined (think of the many possible content images for a single cubistic painting). In contrast, we spare the ill-posed backward mapping of styles and compare stylized and content images in the latent space which is trained jointly with the style transfer network. Since both content and stylized images are run through our encoder, the latent space is trained to only pay attention to the commonalities, i.e., the content present in both. Another consequence of the cycle consistency loss is that it requires content and style images used for training to represent similar scenes [326], and thus training data preparation for [326] involves tedious manual filtering of samples, while our approach can be trained on arbitrary unpaired content and style images.

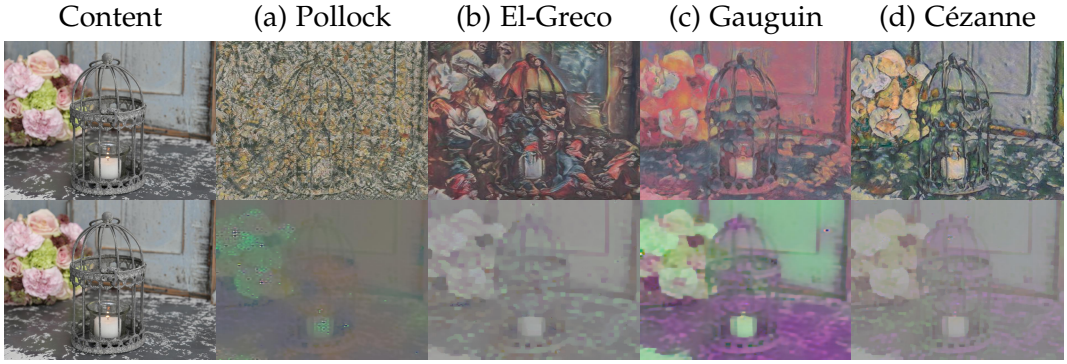


Figure 6.4: 1st row - results of style transfer for different styles. 2nd row - sketchy content visualization reconstructed from the latent space $E(x)$ using method of [185]. (a) The encoder for Pollock does not preserve much content due to the abstract style; (b) only rough structure of the content is preserved (coarse patches) because of the distinct style of El Greco; (c) latent space highlights surfaces of the same color and that fine object details are ignored, since Gauguin was less interested in details, often painted plain surfaces and used vivid colors; (d) encodes the thick, wide brushstrokes Cézanne used, but preserves a larger palette of colors.

6.2 Approach

To enable a fast style transfer that instantly transfers a content image or even frames of a video according to a particular style, we need a feed-forward architecture [130] rather than the slow optimization-based approach of [79]. To this end, we adopt an encoder-decoder architecture that utilizes an encoder network E to map an input content image x onto a latent representation $z = E(x)$. A generative decoder G then plays the role of a painter and generates the stylized output image $y = G(z)$ from the sketchy content representation z. Stylization then only requires a single forward pass, thus working in real-time.

6.2.1 Training with a Style-Aware Content Loss

Previous approaches have been limited in that training worked only with a single style image [59, 79, 117, 130, 164, 269, 286] or that style images used for training had to be similar in content to the content images [326]. In contrast, given a single style image y_0 we include a set Y of related style images $y_j \in Y$, which are automatically selected (see Sec. 6.2.2) from a large art dataset (Wikiart). We do *not* require the y_j to depict similar content as the set X of arbitrary content images $x_i \in X$, which we simply take from Places365 [322]. Compared to [326], we thus can utilize standard datasets for content and style and need no tedious manual selection of the x_i and y_j as described in Sect. 5.1 and 7.1 of [326].

To train E and G we employ a standard advers [237]arial discriminator D [88] to distinguish the stylized output $G(E(x_i))$ from real examples $y_j \in Y$,

$$\mathcal{L}_D(E,G,D) = \mathop{\mathbb{E}}_{y\sim p_Y(y)}[\log D(y)] + \mathop{\mathbb{E}}_{x\sim p_X(x)}[\log(1 - D(G(E(x))))] \tag{6.1}$$

However, the crucial challenge is to decide which details to retain from the content image, something which is not captured by Eq. (6.1). Contrary to previous work, we want to directly enforce E to strip the latent space of all image details that the target style disregards. Therefore, the details that need to be retained or ignored in z depend on the style. For instance, Cubism would disregard texture, whereas Pointillism would retain low-frequency textures. Therefore, a pre-trained network or fixed similarity measure [79] for measuring the similarity in content between x_i and y_i is violating the art historical premise that the manner, in which content is preserved, depends on the style. Similar issues arise when measuring the distance after projecting the stylized image $G(E(x_i))$ back into the domain X of original images with a second pair of encoder and decoder $G_2(E_2(G(E(x_i))))$. The resulting loss proposed in [326],

$$\mathcal{L}_{cycleGAN} = \mathop{\mathbb{E}}_{x\sim p_X(x)}[\|x - G_2(E_2(G(E(x))))\|_1], \tag{6.2}$$

fails where styles become abstract, since the backward projection of abstract art to the original image is highly underdetermined.

Therefore, we propose a style-aware content loss that is being optimized, while the network learns to stylize images. Since encoder training is coupled with training of the decoder, which produces artistic images of the specific style, the latent vector z produced for the input image x can be viewed as its style-dependent sketchy content representation. This latent space representation is changing during training and hence *adapts* to the style. Thus, when measuring the similarity in content between input image x_i and the stylized image $y_i = G(E(x_i))$ in the latent space, we focus only on those details which are relevant for the style. Let the latent space have d dimensions, then we define a style-aware content loss as normalized squared Euclidean distance between $E(x_i)$ and $E(y_i)$:

$$\mathcal{L}_c(E,G) = \mathop{\mathbb{E}}_{x\sim p_X(x)}\left[\frac{1}{d}\|E(x) - E(G(E(x)))\|_2^2\right] \tag{6.3}$$

To show the additional intuition behind the style-aware content loss we used the method [185] to reconstruct the content image from latent representations trained on different styles and illustrated it in Fig. 6.4. It can be seen that latent space encodes a sketchy, style-specific visual content, which is implicitly used by the loss function. For example, Pollock is famous for his abstract paintings, so reconstruction (a) shows that the latent space ignores most of the object structure; Gauguin was less interested in details, painted a lot of plain surfaces and used vivid colors which is reflected in the reconstruction (c), where latent space highlights surfaces of the same color and fine object details are ignored.

Since we train our model for altering the artistic style without supervision and from scratch, we now introduce extra signal to initialize training and boost the learning of the primary latent space. The simplest thing to do is to use an autoencoder loss which computes the difference between x_i and y_i in the RGB space. However, this loss would impose a high penalty for any changes in image structure between input x_i and output y_i, because it relies only on low-level pixel information. But we aim to learn image stylization and want the encoder to discard certain details in the content depending on style. Hence the autoencoder loss will contradict with the purpose of the style-aware loss, where the style determines which details to retain and which to disregard. Therefore, we propose to measure the difference after applying a weak image transformation on x_i and y_i, which is learned while learning E and G. We inject in our model a transformer block **T** which is essentially a one-layer fully convolutional neural network taking an image as input and producing a transformed image of the same size. We apply T to images x_i and $y_i = G(E(x_i))$ before measuring the difference. We refer to this as *transformed image loss* and define it as

$$\mathcal{L}_{\mathbf{T}}(E, G) = \mathop{\mathbb{E}}_{x \sim p_X(x)} \left[\frac{1}{CHW} ||\mathbf{T}(x) - \mathbf{T}(G(E(x))||_2^2 \right], \tag{6.4}$$

where $C \times H \times W$ is the size of image x and for training T is initialized with uniform weights.

Fig. 6.3 illustrates the full pipeline of our approach. To summarize, the full objective of our model is:

$$\mathcal{L}(E, G, D) = \mathcal{L}_c(E, G) + \mathcal{L}_t(E, G) + \lambda \mathcal{L}_D(E, G, D), \tag{6.5}$$

where λ controls the relative importance of adversarial loss. We solve the following optimization problem:

$$E, G = \arg \min_{E,G} \max_{D} \mathcal{L}(E, G, D). \tag{6.6}$$

6.2.2 Style Image Grouping

In this section we explain an automatic approach for gathering a set of related style images. Given a single style image y_0 we strive to find a set Y of related style images $y_j \in Y$. Contrary to [326] we avoid tedious manual selection of style images and follow a fully automatic approach. To this end, we train a VGG16 [249] network C from scratch on the Wikiart [138] dataset to predict an artist given the artwork. The network is trained on the 624 largest (by number of works) artists from the Wikiart dataset. Note that our ultimate goal is stylization and numerous artists can share the same style, e.g., Impressionism, as well as a single artist can exhibit different styles, such as the different stylistic periods of Picasso. However, we do *not* use any *style* labels. Artist classification in this case is the surrogate task

for learning meaningful features in the artworks' domain, which allows to retrieve similar artworks to image y_0.

Let $\phi(y)$ be the activations of the fc6 layer of the VGG16 network C for input image y. To get a set of related style images to y_0 from the Wikiart dataset $\mathcal{Y}$ we retrieve all nearest neighbors of y_0 based on the cosine distance δ of the activations $\phi(\cdot)$, i.e.

$$Y = \{y \mid y \in \mathcal{Y}, \delta(\phi(y), \phi(y_0)) < t\}, \tag{6.7}$$

where $\delta(a, b) = 1 + \frac{\phi(a)\phi(b)}{||a||_2||b||_2}$ and t is the 10% quantile of all pairwise distances in the dataset $\mathcal{Y}$.

6.3 Implementation Details

The basis for our style transfer model is an encoder-decoder architecture, cf. [130]. The encoder network contains 5 `conv` layers:
1×`conv-stride-1` and 4×`conv-stride-2`. The decoder network has 9 residual blocks [102], 4 upsampling blocks and 1×`conv-stride-1`. For upsampling blocks we used a sequence of nearest-neighbor upscaling and `conv-stride-1` instead of fractionally strided convolutions [175], which tend to produce heavier artifacts [209]. Discriminator is a fully convolutional network with 7×`conv-stride-2` layers. For a detailed network architecture description we refer to the supplementary material. We set $\lambda = 0.001$ in Eq. (6.5). During the training process we sample 768×768 content image patches from the training set of Places365 [322] and 768×768 style image patches from the Wikiart [138] dataset. We train for 300000 iterations with batch size 1, learning rate 0.0002 and Adam [145] optimizer. The learning rate is reduced by a factor of 10 after 200000 iterations.

6.3.1 Network Architecture

We incarnate our approach using an encoder-decoder architecture with a discriminator. Below we follow the naming convention similar to the one used in [326].

Let:

* `cFs1-k` denote $F \times F$ Convolution-InstanceNorm-ReLU layer with k filters and stride 1;
* `cFs1-k-sigmoid` denote $F \times F$ Convolution-InstanceNorm-Sigmoid layer with k filters and stride 1;
* `cFs1-k-noact` denote $F \times F$ Convolution-InstanceNorm layer with k filters and stride 1;
* `dF-k` denote a $F \times F$ Convolution-InstanceNorm-ReLU layer with k filters and stride 2;

* `dF-k-LReLU` denote a $F \times F$ Convolution-InstanceNorm-LeakyReLU layer with k filters and stride 2;

* `Rk` denote a residual block that contains two 3×3 convolutional layers followed by InstanceNorm [271];

* `uk` denote an upscaling block that contains nearest-neighbor upscaling by a factor of 2 followed by Convolution-InstanceNorm-ReLU layer with k filters and stride 1.

All convolutional layers use reflection padding.

Encoder-decoder architecture. Encoder has InstanceNorm [271] layer at the beginning and 5 convolutional layers: `InstanceNorm, c3s1-32, d3-32, d3-64, d3-128, d3-256`. Decoder contains 9 residual blocks, 4 upscaling blocks and 1 convolution 7×7 with sigmoid activation: `R256`$\times 9$`, u256, u128, u64, u32, c7s1-3-sigmoid`.

Transformer block T. Transformer block **T** contains a convolutional layer followed by a weight normalization layer with fixed norm $||W|| = 1$. Convolutional layer: 3 kernels of size 10×10, stride 1, uniformly initialized.

Discriminator architecture. Discriminator is implemented as a fully convolutional network and tries to classify if input image patches are real or fake. It contains 7 convolutional layers. We use Leaky ReLU activations with slope 0.2. Architecture is defined as: `d5-128-LReLU, d5-128-LReLU, d5-256-LReLU, d5-512-LReLU, d5-512-LReLU, d5-1024-LReLU, d5-1024-LReLU, c3s1-1-noact`.

We use 4 auxiliary classifiers (implemented as a convolutional layer with 1 filter each) to capture image details on different scales [119, 303] and to alleviate artifacts: auxiliary classifiers were added after 1st, 2nd, 4th and 6th convolutional layers of the discriminator. We simply sum up all the losses from different scales.

6.3.2 Training Details

All networks are trained from scratch on randomly cropped image patches of size 768×768 pix. λ in Eq. (6.5) equals to 0.001. We traine for 300000 iterations with batch size 1 and learning rate 2×10^{-4} using Adam [145] optimizer. The learning rate is reduced by a factor of 10 after 200000 iterations.

In each iteration we alternatively update encoder-decoder and discriminator. To balance out the discriminator and encoder-decoder training [45] we update the discriminator solely, if it has accuracy < 0.8, and update only encoder-decoder otherwise. The discriminator accuracy is calculated using the exponential moving average during training.

6.3.3 Style Image Grouping Details

As described in Sec. 6.2.2, the number of style images is not fixed and depends on the neighborhood size of the query image y_0 and how frequent the style is in the Wikiart [138] dataset. For different style examples in our experiments, it varied from 55 to 1391 related images. We observe: the more style examples the better, as long as they are from the same style. Ablation studies (Sec. 6.4.3) show that using too few style examples (e.g., one) leads to mode collapse; at the same time using a lot of images, which are less related to each other (e.g., all images of an artist), produces unsatisfactory stylizations.

6.4 Experiments

To compare our style transfer approach with the state-of-the-art, we first perform extensive qualitative analysis, then we provide quantitative results based on the *deception score* and evaluations of experts from art history. Afterwards in Sect. 6.4.3 we ablate single components of our model and show their importance.

Baselines. Since we aim to generate high-resolution stylizations, for comparison we run style transfer on our method and all baselines for input images of size 768×768, unless otherwise specified. We did not not exceed this resolution when comparing, because some other methods were reaching the GPU memory limit. We optimize Gatys et al. [79] for 500 iterations using L-BFGS [170]. For Johnson et al. [130] we used the implementation of [64] and trained a separate network for every reference style image on the same content images from Places365 [322] as our method. For Huang et al. [117], Chen et al. [35] and Li et al. [164] implementations and pre-trained models provided by the authors were used. Zhu et al. [326] was trained on exactly the same content and style images as our approach using the source code provided by the authors. Methods [35, 79, 117, 130, 164] utilized only one example per style, as they cannot benefit from more (cf. the analysis in Fig. 6.2).

6.4.1 Qualitative Results

Full image stylization. In Fig. 6.5 we demonstrate the effectiveness of our approach for stylizing different contents with various styles. Chen et al. [35] work on the overlapping patches extracted from the content image, swapping the features of the original patch with the features of the most similar patch in the style image, and then averages the features in the overlapping regions, thus producing an over-smoothed image without fine details (Fig. 6.5 (d)). [117] produces a lot of repetitive artifacts, especially visible on flat surfaces, cf. Fig. 6.5 (e, rows 1, 4–6). Method of Li et al. [164] fails to understand the content of the image and applies different colors in the wrong locations (Fig. 6.5 (f)). Johnson et al. [130] and Zhu et al. [326] often fail to alter content image and their effect may be characterized as shifting the color

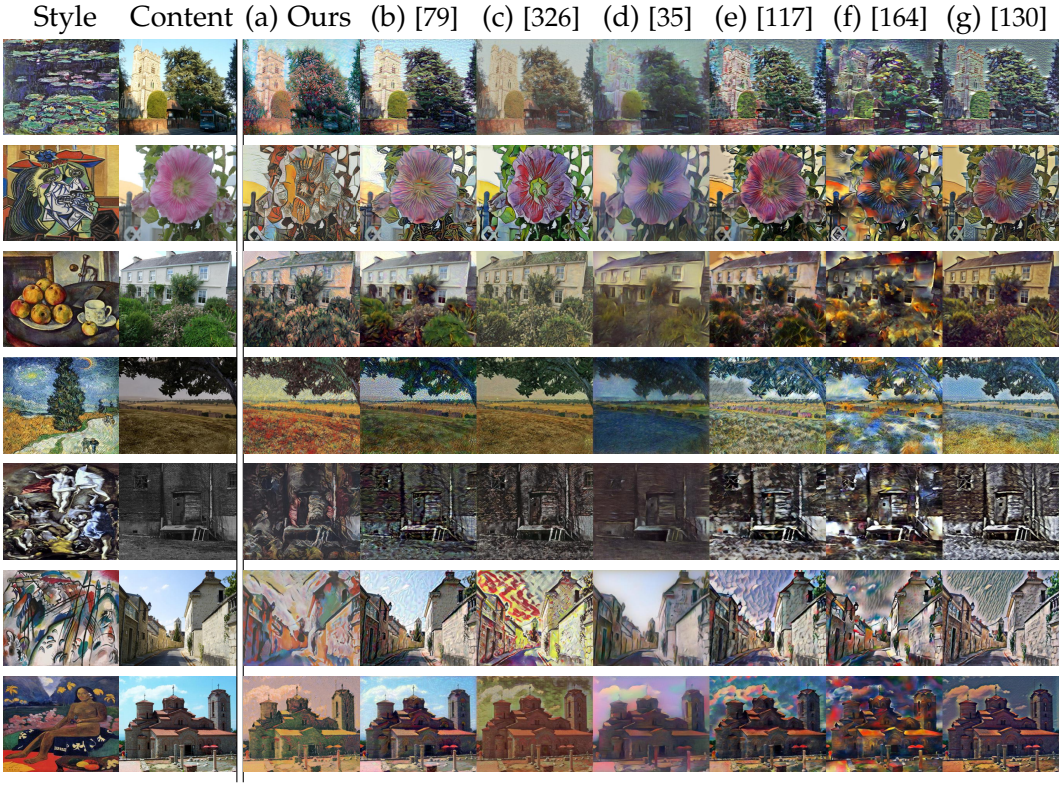


Figure 6.5: Results from different style transfer methods. We compare methods on different styles and content images.

histogram, e.g., Fig. 6.5 (g, rows 3, 7; c, rows 1, 3–4). One reason for such failure cases of [326] is the loss in the RGB pixel space based on the difference between a backward mapping of the stylized output and the content image. Another reason for this is that we utilized the standard Places365 [322] dataset and did not hand-pick training content images, as is advised for [326]. Thus, artworks and content images used for training differed significantly in their content, which is the ultimate test for a stylization that truly alters the input and goes beyond a direct mapping between regions of content and style images. The optimization-based method [79] often works better than other baselines, but produces a lot of prominent artifacts, leading to details of stylizations looking unnatural, cf. Fig. 6.5 (b, rows 4, 5, 6). This is due to an explicit minimization of the loss directly on the pixel level. In contrast to this, our model can not only handle styles, which have salient, simple to spot characteristics, but also styles, such as El Greco's Mannerism, with less graspable stylistic characteristics, where other methods fail (Fig. 6.5, b–g, 5th row).

Fine-grained style details. In Fig. 6.7 we show zoomed in cut-outs from the stylized images. Interestingly, the stylizations of methods [35, 79, 109, 117, 164] do not change much across styles (compare Fig. 6.7 (d, f–i, rows 1–3)). Zhu et

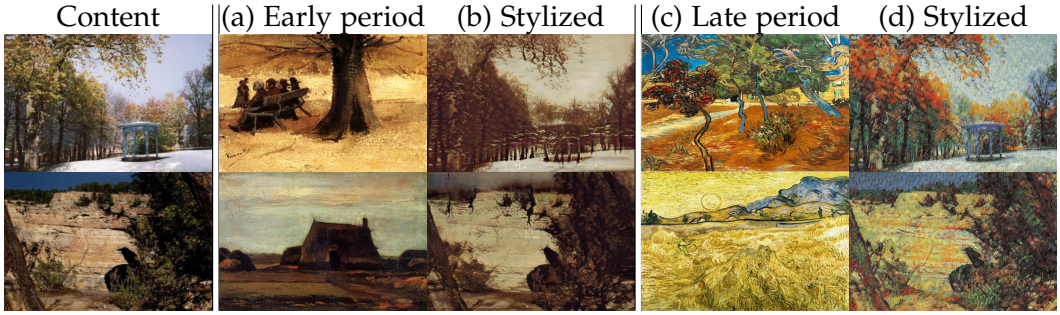


Figure 6.6: Artwork examples of the early artistic period of van Gogh (a) and his late period (c). Style transfer of the content image (1st column) onto the early period is presented in (b) and the late period in (d).

al. [326] produce more diverse images for different styles, but obviously cannot alter the edges of the content (blades of grass are *clearly visible* on all the cutouts in Fig. 6.7 (e)). Fig. 6.7 (c) shows the stylized cutouts of our approach, which exhibit significant changes from one style to another. Another interesting example is the style of Pollock, Fig. 6.7 (row 8), where the style-aware loss allows our model to properly alter content to the point of discarding it – as would be expected from a Pollock action painting. Our approach is able to generate high-resolution stylizations with a lot of style specific details and retains those content details which are necessary for the style.

Style transfer for different periods of van Gogh. We now investigate our ability to properly model fine differences in style *despite* using a group of style images. Therefore, we take two reference images Fig. 6.6 (a) and (c) from van Gogh's early and late period, respectively, and acquire related style images for both from Wikiart. It can be clearly seen that the stylizations produced for either period Fig. 6.6 (b, d) are fairly different and indeed depict the content in correspondence with the style of early (b) and late (d) periods of van Gogh. This highlights that collections of style images are properly used and do not lead to an averaging effect.

High-resolution image generation. Our approach allows us to produce high quality stylized images in high-resolution. Fig. 6.8 illustrates an example of the generated piece of art in the style of Berthe Morisot with resolution 1280×1280. The result exhibits a lot of fine details such as color transitions of the oil paint and brushstrokes of different sizes. In Fig. 6.13, 6.14 we show extra high-resolution images stylized by our approach. Additional comparison of our method against baselines and more high-resolution visual results are available on the project page[2].

Altering the style of an existing artwork. Our method is able to change the style of an existing artwork, rendering it in another style. It means, that our algorithm

[2] https://compvis.github.io/adaptive-style-transfer

Figure 6.7: Details from stylized images produced for different styles for a fixed content image (a). (b) is our entire stylized image, (c) the zoomed in cut-out and (d)-(i) the same region for competitors. Note the variation across different styles along the *column* for our method compared to other approaches. This highlights the ability to adapt content (not just colors or textures) where demanded by a style. Fine grained artistic details with sharp boundaries are produced, while altering the original content edges.

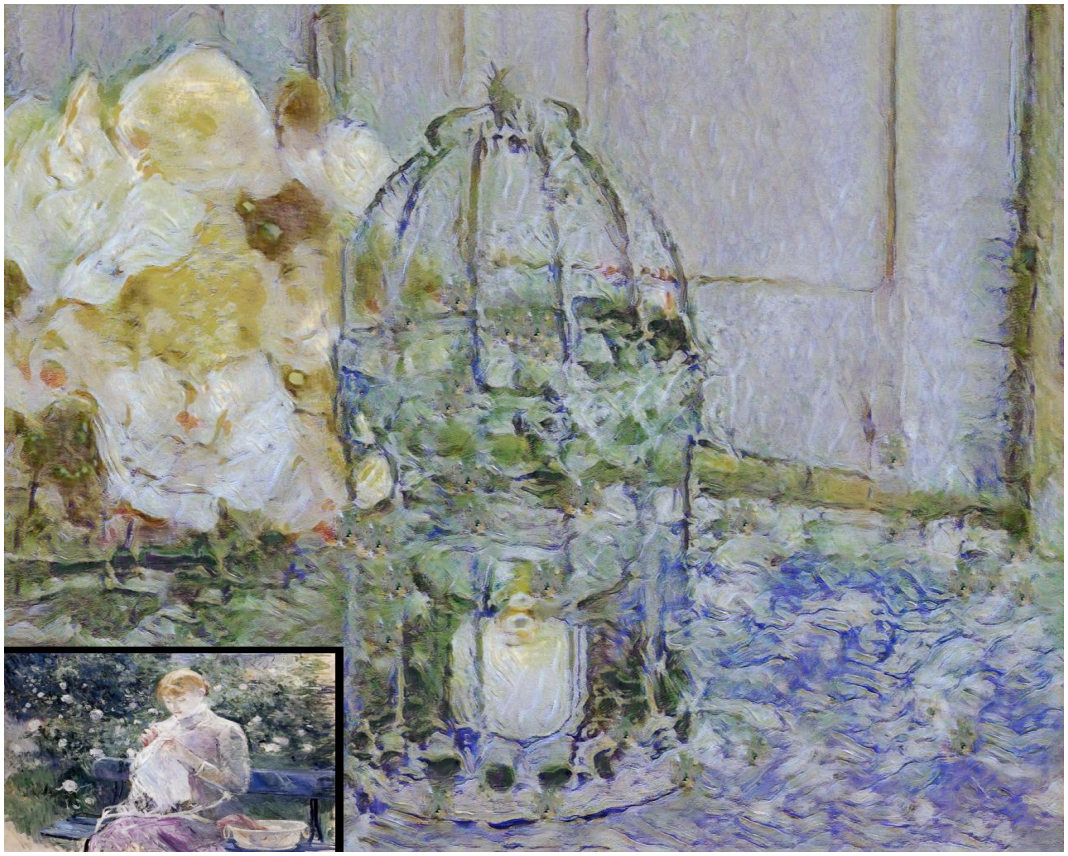


Figure 6.8: High-resolution image (1280x1280 pix) generated by our approach in style of Berthe Morisot. A lot of fine details and brushstrokes are visible. A style example is shown in the bottom left corner.

can also handle content images which are artistic, which, to our knowledge, was never shown in previous work before. We refer the reader to the results on the project page.

Real-time HD video stylization. We also apply our method to several videos. Our approach can stylize High-definition (HD) videos (1280×720) at 9 FPS. Fig. 6.9 shows stylized frames from a video. We did not use a temporal regularization to show that our method produces equally good results for consecutive frames with varying appearance without extra constraints. Stylized videos are available on the project page.

6.4.2 Quantitative Evaluation

Style transfer deception rate. While several metrics [29, 108, 234] have been proposed to evaluate the quality of image generation, until now no evaluation metric

Figure 6.9: Results of our approach applied to the HD video of Eadweard Muybridge "The horse in motion" (1878). Every frame was *independently* processed (no smoothing or post-processing) by our model in the style of Picasso. Video resolution is 1920×1280 pix. The full video is available on YouTube: `https://youtu.be/TtHJcL8Feu0`.

has been proposed for an automatic evaluation of style transfer results. To measure the quality of the stylized images, we introduce the *style transfer deception rate*. We use a VGG16 network trained from scratch to classify 624 artists on Wikiart. Style transfer deception rate is calculated as the fraction of generated images which were classified by the network as the artworks of an artist for which the stylization was produced. For fair comparison with other approaches, which used only one style image y_0 (hence only one artist), we restricted Y to only contain samples coming from the same artist as the query example y_0. We selected 18 different artists (i.e. styles). For every method we generated 5400 stylizations (18 styles, 300 per style). In Tab. 6.1 we report mean deception rate for 18 styles. Our method achieves 0.393 significantly outperforming the baselines. For comparison, mean accuracy of the network on hold-out real images of aforementioned 18 artists from Wikiart is 0.616.

Human art history experts perceptual studies. Three experts (with a Ph.D. in art history with a focus on modern and pre-modern paintings) have compared the results of our method against recent work. Each expert was shown 1000 groups of images. Each group consists of stylizations that were generated by different methods based on the same content and style images. Experts were asked to choose one image which best and most realistically reflects the current style. The score is computed as the fraction of times a specific method was chosen as the best in the group. We calculate a mean expert score for each method using 18 different styles and report them in Tab. 6.1. Here, we see that the experts selected our method in around 50% of the cases.

Speed and memory. Tab. 6.2 shows the time and memory required for stylization of a single image of size 768×768 px for different methods. One can see that our approach and that of [130] and [326] have comparable speed and only very modest demands on GPU memory, compared to modern graphics cards.

Method	Deception rate	Expert score
Content images	0.002	-
AdaIn [117]	0.074	0.060
PatchBased [35]	0.040	0.132
Johnson et al. [130]	0.051	0.048
WCT [164]	0.035	0.044
CycleGan [326]	0.139	0.044
Gatys et al. [79]	0.147	0.178
Ours	**0.393**	**0.495**

Table 6.1: Mean deception rate and mean expert score for different methods. The higher the better.

Method	Time	GPU memory
AdaIn [117]	0.16 sec	8872 MiB
PatchBased [35]	8.70 sec	4159 MiB
Johnson et al. [130]	**0.06 sec**	**671 MiB**
WCT [164]	5.22 sec	10720 MiB
CycleGan [326]	0.07 sec	1391 MiB
Gatys et al. [79]	200 sec	3887 MiB
Ours	0.07 sec	1043 MiB

Table 6.2: Average inference time and GPU memory consumption, measured on a Titan X Pascal, for different methods with batch size 1 and input image of 768×768 pix.

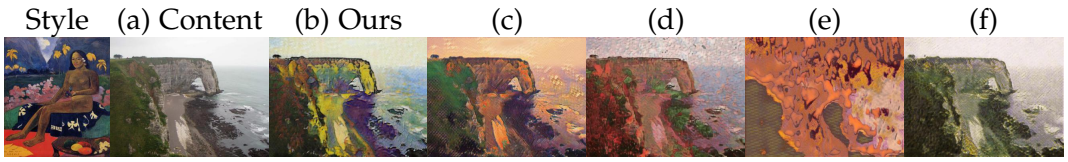


Figure 6.10: Different variations of our method for Gauguin stylization. See Sect. 6.4.3 for details. (a) Content image; (b) full model ($\mathcal{L}_c$, $\mathcal{L}_{rgb}$ and $\mathcal{L}_D$); (c) $\mathcal{L}_{rgb}$ and $\mathcal{L}_D$; (d) without transformer block; (e) only $\mathcal{L}_D$; (f) trained with all of Gauguin's artworks as style images. Please zoom in to compare.

6.4.3 Ablation Studies

Effect of different losses. We study the effect of different components of our model in Fig. 6.10. Removing the style-aware content loss significantly degrades the results, (c). We observe that without the style-aware loss training becomes instable and often stalls. If we remove the transformed image loss, which we introduced for a proper initialization of our model that is trained from scratch, we notice mode collapse after 5000 iterations. Training directly with pixel-wise L2 distance causes a lot of artifacts (grey blobs and flaky structure), (d). Training only with a discriminator neither exhibits the variability in the painting nor in the content, (e). Therefore we conclude that both the style-aware content loss and the transformed image loss are critical for our approach.

Single vs collection of style images. Here, we investigate the importance of the style image grouping. First, we trained a model with only one style image of Gauguin, which led to mode collapse. Second, we trained with all of Gauguin's artworks as style images (without utilizing style grouping procedure). It produced unsatisfactory results, cf. Fig. 10(f), because style images comprised several distinct styles. Therefore we conclude that to learn a good style transfer model it is important to group style images according to their stylistic similarity.

Figure 6.11: Encoder ablation studies: (a) stylization using our model; (b) stylization using pre-trained VGG16 encoder instead of E.

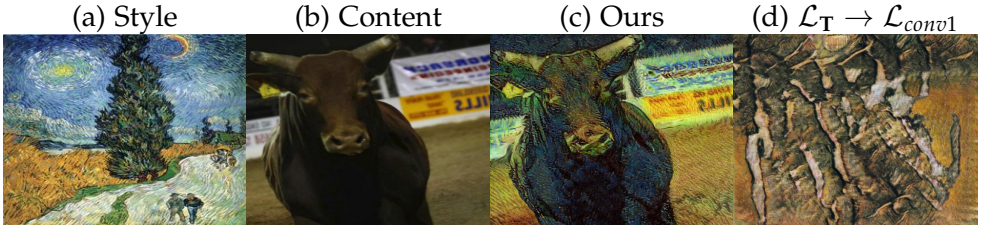


Figure 6.12: Investigation of the importance of an independent transformer block. (a) style image (van Gogh); (b) content image; (c) our stylization; (d) stylization with the loss $\mathcal{L}_{conv1}$ instead of $\mathcal{L}_{\mathbf{T}}$.

Encoder ablation. To investigate the effect of our encoder E, we substitute it with VGG16 [249] encoder (up to `conv5_3`) pre-trained on ImageNet. The VGG encoder retains features that separate object classes (since it was trained discriminatively), as opposed to our encoder which is trained to retain style-specific content details. Hence, our encoder is not biased towards class-discriminative features, but is style specific and trained from scratch. Fig. 6.11 (a, b) show that our approach produces better results than with pre-trained VGG16 encoder.

Transformer block ablation. To demonstrate the need in an independent transformer block, we replaced the transformed image loss $\mathcal{L}_{\mathbf{T}}$ with a loss $\mathcal{L}_{conv1}$ using `conv1` features of the encoder,

$$\mathcal{L}_{conv1} = (E, G) = \mathop{\mathbb{E}}_{x \sim p_X(x)} \left[\frac{1}{d_{conv1}} ||\phi_{conv1}(x) - \phi_{conv1}(G(E(x))||_2^2 \right], \tag{6.8}$$

where $\phi_{conv1}(x)$ are the activations of the `conv1` layer of the encoder E for input image x and d_{conv1} is the dimensianality of $\phi_{conv1}(x)$. This produced worse results (cf. Fig. 6.12 (d)), since the loss $\mathcal{L}_{conv1}$ is then directly tied to the convolutional layer used for our style-aware content representation in the encoder. In contrast, the proposed transformer block can be learned independently from the early layers of the encoder.

6.5 Conclusion

This chapter has addressed major conceptual issues in state-of-the-art approaches for style transfer. We overcome the limitation of only a single style image or the need for style and content training images to show similar content. Moreover, we exceed a mere pixel-wise comparison of stylistic images or models that are pre-trained on millions of ImageNet bounding boxes. The proposed style-aware content loss makes the encoder network extract merely content representation while adapting to the quirks of artistic style, determining how the content is preserved. Our approach enables a real-time, high-resolution encoder-decoder based stylization of images and videos and significantly improves stylization by capturing how style affects content. [3]

[3] *Solution to Fig. 6.1: patches* 3 *and* 5 *were generated by our approach, others by the artist.*

Figure 6.13: High-resolution (2288 × 1280 pix) Pissarro stylization by our approach. The top row depicts 4 randomly chosen from set Y instances of style obtained by the style image grouping.

Figure 6.14: High-resolution (1712 × 1280 pix) Roerich stylization by our approach. The top row depicts 4 randomly chosen from set Y instances of style obtained by the style image grouping.

7 Conclusion

The big question which was the focus of this dissertation is bridging the gap between current Computer Vision systems and future systems that are expected to be self-motivated (i.e., not requiring constant human feedback) and capable of accumulating experience for faster learning of new tasks. As a step towards this, we introduced several new approaches for learning such visual representations that can effectively generalize to previously unseen data. We considered two scenarios: (i) some amount of annotations is available, and (ii) absolutely no annotations are provided.

In Chapters 3, 4 we showed that it is possible to learn generalizable representations from raw input data without annotations. We proposed a novel method that can learn visual representations by discovering regular structures in unlabeled data and using them as a supervisory signal (Chapter 3). Even if the initial data representation is not powerful enough and cannot be used to estimate reliable relationships between certain samples, these samples can still be used for unsupervised learning by enforcing a partial ordering between them (Chapter 4).

Next, we tackled the problem of efficient learning from limited annotated data. In the same spirit as the above-mentioned techniques, in addition to the user-provided labels we exploited regularities in the data. We showed that training specialized learners for different subsets of the data discovered by unsupervised grouping facilitates learning more generalizable representations (Chapter 2). Furthermore, we advanced the methods for generalization to completely novel tasks without supervision. To this end, we introduced a self-training approach that leverages the system's prior knowledge to master a novel task without additional annotations (Chapter 5).

Finally, we studied the learning of disentangled visual representations. We argue that such representations are required to separate different visual factors within an image and enable further analysis and modification of each of the factors individually. We introduced a novel method for learning such a content representation of an image that is disentangled from its style. This task is especially challenging because it is not possible to collect annotations of this kind; thus, the proposed approach had to be self-supervised. Our representation made it possible to produce compelling high-resolution stylizations of arbitrary images and videos (Chapter 6).

Future directions. While significant progress has been made in large-scale supervised and unsupervised learning in recent years, the power-efficiency of the current

deep learning approaches should still be improved. If we look at us, humans, we are continually learning new information by observing and interacting with our environment, processing large amounts of visual data in real-time, and can learn to solve numerous seemingly unrelated tasks. This hints at the exceptional power-efficiency of our brain. In contrast, in deep learning, we see a trend of increasing power-consumption of models. Recent models require several megawatt-hours of energy to be trained while being limited to a specific domain [21, 56, 141]. Developing more energy-effective machine learning models can democratize the whole field of AI research and make them available not only to large corporations like Google, but to any academic research group, and will ultimately increase their integration in our everyday life. It will enable further development of the lifelong machine learning approaches [39] which aim at mimicking the human learning process – incremental and continuous attainment of new knowledge and skills during a long time. By the words of Zhiyuan Chen and Bing Liu [39]:

> *Without the capability of retaining and accumulating the knowledge learned in the past, making inference about it, and using the knowledge to help future learning and problem solving, achieving general intelligence is quite unlikely.*

Another underexplored research direction is multipurpose intelligent systems that can solve multiple tasks simultaneously. This can potentially enable learning richer representations by exploiting relationships between different tasks. The tasks may span different domains and modalities, e.g., images, sound, text, or any other sensory data. While there were several attempts at combining multiple tasks and modalities for learning [52, 134, 147, 314], we hope to see more research in this direction in the future.

Acronyms

AI	Artificial Intelligence
AUC	Area under curve
CNN	Convolutional Neural Network
DML	Deep Metric Learning
Exemplar-SVM	Exemplar Support Vector Machine
FPN	Feature Pyramid Network
GAN	Generative Adversarial Network
GPU	Graphics processing unit
HD	High-definition
HOG	Histogram of Oriented Gradients
ICA	Independent component analysis
LDA	Linear Discriminant Analysis
LSTM	Long Short-Term Memory
LTP	Local temporal average pooling
MDS	Multidimensional scaling
NMF	Non-negative matrix factorization
PCA	Principal component analysis
PCK	Percentage of Correct Keypoints
PCP	Percentage of Correct Parts
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SGD	Stochastic gradient descent
SMPL	Skinned Multi-Person Linear Model
SVM	Support Vector Machine

Bibliography

1. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. "TensorFlow: Large-scale machine learning on heterogeneous systems, 2015". *Software available from tensorflow. org* 1, 2015.
2. R. Abdal, Y. Qin, and P. Wonka. "Image2stylegan: How to embed images into the stylegan latent space?" In: *Proceedings of the IEEE international conference on computer vision*. 2019, pp. 4432–4441.
3. B. Alexe, T. Deselaers, and V. Ferrari. "Measuring the objectness of image windows". *Pattern Analysis and Machine Intelligence* 34:11, 2012.
4. [Ananabanana]. *CC BY-NC-SA 2.0*. `https://www.flickr.com/photos/ananabanana/14682376194/`. 2009.
5. M. Andriluka, U. Iqbal, E. Ensafutdinov, L. Pishchulin, A. Milan, J. Gall, and S. B. "PoseTrack: A Benchmark for Human Pose Estimation and Tracking". *CVPR*, 2018.
6. M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. "2d human pose estimation: New benchmark and state of the art analysis". In: *CVPR*. 2014.
7. Y. M. Asano, C. Rupprecht, and A. Vedaldi. "Self-labelling via simultaneous clustering and representation learning". *arXiv preprint arXiv:1911.05371*, 2019.
8. N. Aziere and S. Todorovic. "Ensemble Deep Manifold Similarity Learning Using Hard Proxies". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.
9. Y. Babakhin, A. Sanakoyeu, and H. Kitamura. "Semi-Supervised Segmentation of Salt Bodies in Seismic Images using an Ensemble of Convolutional Neural Networks". *German Conference on Pattern Recognition (GCPR)*, 2019.
10. R. Baillargeon. "How do infants learn about the physical world?" *Current Directions in Psychological Science* 3:5, 1994, pp. 133–140.
11. M. A. Bautista, A. Sanakoyeu, E. Tikhoncheva, and B. Ommer. "Cliquecnn: Deep Unsupervised Exemplar Learning". In: *Advances in Neural Information Processing Systems*. 2016, pp. 3846–3854.
12. M. Á. Bautista, A. Sanakoyeu, and B. Ommer. "Deep Unsupervised Similarity Learning using Partially Ordered Sets." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1923–1932.
13. A. Beck and L. Tetruashvili. "On the convergence of block coordinate descent type methods". *SIAM journal on Optimization* 23:4, 2013, pp. 2037–2060.

14. S. Bell and K. Bala. "Learning visual similarity for product design with convolutional neural networks". *ACM Transactions on Graphics (TOG)* 34:4, 2015, p. 98.
15. Y. Bengio, A. Courville, and P. Vincent. "Representation learning: A review and new perspectives". *IEEE transactions on pattern analysis and machine intelligence* 35:8, 2013, pp. 1798–1828.
16. D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. "Mixmatch: A holistic approach to semi-supervised learning". In: *Advances in Neural Information Processing Systems*. 2019, pp. 5049–5059.
17. B. Biggs, T. Roddick, A. Fitzgibbon, and R. Cipolla. "Creatures Great and SMAL: Recovering the shape and motion of animals from video". *ACCV*, 2018.
18. B. Brattoli, U. Buchler, A.-S. Wahl, M. E. Schwab, and B. Ommer. "Lstm self-supervision for detailed behavior analysis". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6466–6475.
19. A. Brock, J. Donahue, and K. Simonyan. "Large scale GaN training for high fidelity natural image synthesis". *7th International Conference on Learning Representations, ICLR 2019*, 2019, pp. 1–35.
20. J. Bromley, I. Guyon, Y. LeCun, E. Sickinger, and R. Shah. "Signature Verification using a" Siamese" Time Delay Neural Network", 1994.
21. T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. "Language models are few-shot learners". *arXiv preprint arXiv:2005.14165*, 2020.
22. U. Büchler, B. Brattoli, and B. Ommer. "Improving spatiotemporal self-supervision by deep reinforcement learning". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 770–786.
23. Burkard, Çela, Pardalos, and Pitsoulis. "The quadratic assignment problem". In: *Handbook of Combinatorial Optimization*. 1998.
24. F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff. "Deep metric learning to rank". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1861–1870.
25. J. Cao, H. Tang, F. Hao-Shu, X. Shen, C. Lu, and Y.-W. Tai. "Cross-Domain Adaptation for Animal Pose Estimation". *ICCV*, 2019.
26. Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. "Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields". *CVPR*, 2017.
27. M. Caron, P. Bojanowski, A. Joulin, and M. Douze. "Deep clustering for unsupervised learning of visual features". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 132–149.
28. O. Chapelle, B. Scholkopf, and A. Zien. *Semi-supervised learning*. MIT Press, 2006.

29. T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. "Mode regularized generative adversarial networks". *arXiv preprint arXiv:1612.02136*, 2016.
30. D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua. "Stylebank: An explicit representation for neural image style transfer". In: *Proc. CVPR*. 2017.
31. L. Chen, G. Papandreou, F. Schroff, and H. Adam. "Encoder-decoder with atrous separable convlution for semantic image segmentation". *ECCV*, 2018.
32. L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. "Rethinking atrous convolution for semantic image segmentation". *arXiv preprint arXiv:1706.05587*, 2017.
33. Q. Chen and V. Koltun. "Photographic image synthesis with cascaded refinement networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1511–1520.
34. T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. "Sketch2photo: Internet image montage". *ACM transactions on graphics (TOG)* 28:5, 2009, pp. 1–10.
35. T. Q. Chen and M. Schmidt. "Fast patch-based style transfer of arbitrary style". *arXiv preprint arXiv:1612.04337*, 2016.
36. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. "A simple framework for contrastive learning of visual representations". In: *International Conference on Machine Learning*. 2020.
37. T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton. "Big self-supervised models are strong semi-supervised learners". *Advances in Neural Information Processing Systems* 33, 2020.
38. X. Chen, H. Fan, R. Girshick, and K. He. "Improved baselines with momentum contrastive learning". *arXiv preprint arXiv:2003.04297*, 2020.
39. Z. Chen and B. Liu. "Lifelong machine learning". *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12:3, 2018, pp. 1–207.
40. F. Chollet. "Xception: Deep learning with depthwise separable convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1251–1258.
41. S. Chopra, R. Hadsell, and Y. LeCun. "Learning a similarity metric discriminatively, with application to face verification". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 539–546.
42. R. Chu, Y. Sun, Y. Li, Z. Liu, C. Zhang, and Y. Wei. "Vehicle re-identification with viewpoint-aware metric learning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 8282–8291.
43. J. Collomosse, T. Bui, M. J. Wilber, C. Fang, and H. Jin. "Sketching With Style: Visual Search With Sketches and Aesthetic Context". In: *The IEEE International Conference on Computer Vision (ICCV)*. 2017.

44. P. Comon. "Independent component analysis, a new concept?" *Signal processing* 36:3, 1994, pp. 287–314.
45. A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. "Generative Adversarial Networks: An Overview". *IEEE Signal Processing Magazine* 35:1, 2018, pp. 53–65.
46. E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. "Randaugment: Practical automated data augmentation with a reduced search space". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 702–703.
47. N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *CVPR*. Vol. 1. IEEE. 2005, pp. 886–893.
48. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *CVPR*. IEEE. 2009, pp. 248–255.
49. S. Ding, L. Lin, G. Wang, and H. Chao. "Deep feature learning with relative distance comparison for person re-identification". *Pattern Recognition* 48:10, 2015, pp. 2993–3003.
50. Doersch, Singh, Gupta, Sivic, and Efros. "What makes paris look like paris?" *ACM TOG* 31:4, 2012.
51. C. Doersch, A. Gupta, and A. A. Efros. "Unsupervised visual representation learning by context prediction". In: *ICCV*. 2015, pp. 1422–1430.
52. C. Doersch and A. Zisserman. "Multi-task self-supervised visual learning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2051–2060.
53. J. Donahue, P. Krähenbühl, and T. Darrell. "Adversarial feature learning". In: 2017.
54. J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. "Long-term recurrent convolutional networks for visual recognition and description". In: *CVPR*. 2015.
55. Dosovitskiy, Springenberg, Riedmiller, and Brox. "Discriminative unsupervised feature learning with convolutional neural networks". In: *NIPS*. 2014.
56. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *Proc. ICLR*. 2021.
57. Y. Duan, W. Zheng, X. Lin, J. Lu, and J. Zhou. "Deep Adversarial Metric Learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2780–2789.
58. Duchi, Hazan, and Singer. "Adaptive subgradient methods for online learning and stochastic optimization". *JMLR* 12:Jul, 2011, pp. 2121–2159.

59. V. Dumoulin, J. Shlens, and M. Kudlur. "A learned representation for artistic style". *Proc. of ICLR*, 2017.

60. A. A. Efros and W. T. Freeman. "Image quilting for texture synthesis and transfer". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 2001, pp. 341–346.

61. A. Eigenstetter, M. Takami, and B. Ommer. "Randomized max-margin compositions for visual recognition". In: *CVPR*. IEEE. 2014.

62. I. El-Naqa, Y. Yang, N. P. Galatsanos, R. M. Nishikawa, and M. N. Wernick. "A similarity learning approach to content-based image retrieval: application to digital mammography". *TMI* 23:10, 2004, pp. 1233–1244.

63. Y. Em, F. Gag, Y. Lou, S. Wang, T. Huang, and L.-Y. Duan. "Incorporating intra-class variance to fine-grained visual recognition". In: *Multimedia and Expo (ICME), 2017 IEEE International Conference on*. IEEE. 2017, pp. 1452–1457.

64. L. Engstrom. *Fast Style Transfer*. `https://github.com/lengstrom/fast-style-transfer/`. commit 55809f4e. 2016.

65. P. Esser, E. Sutter, and B. Ommer. "A Variational U-Net for Conditional Appearance and Shape Generation". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

66. M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. "The pascal visual object classes challenge: A retrospective". *International journal of computer vision* 111:1, 2015, pp. 98–136.

67. M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. "The pascal visual object classes (voc) challenge". *IJCV* 88:2, 2010.

68. S. Feather. *CC BY-NC-SA 2.0*. `https://www.flickr.com/photos/7317295@N04/8631651965/`. 2013.

69. N. Fedele. *CC BY-NC-SA 2.0*. `https://www.flickr.com/photos/nf4000/6677286321/`. 2011.

70. L. Fei-Fei, R. Fergus, and P. Perona. "One-shot learning of object categories". *IEEE transactions on pattern analysis and machine intelligence* 28:4, 2006, pp. 594–611.

71. P. Felzenszwalb, D. McAllester, and D. Ramanan. "A discriminatively trained, multiscale, deformable part model". In: *CVPR*. IEEE. 2008, pp. 1–8.

72. E. Fernie. *Art History and its Methods: A critical anthology*. p. 361. Phaidon, London, 1995, p. 361. ISBN: 978-0-7148-2991-3.

73. V. Ferrari, M. Marin-Jimenez, and A. Zisserman. "Pose search: retrieving people using their pose". In: *CVPR*. IEEE. 2009, pp. 1–8.

74. V. Ferrari and A. Zisserman. "Learning visual attributes". In: *Advances in neural information processing systems*. 2008, pp. 433–440.

75. M. Fink. “Object classification from a single example utilizing class relevance metrics”. *Advances in neural information processing systems* 17, 2004, pp. 449–456.

76. Y. Freund and R. E. Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. *Journal of computer and system sciences* 55:1, 1997, pp. 119–139.

77. O. Frigo, N. Sabater, J. Delon, and P. Hellier. “Split and match: Example-based adaptive patch sampling for unsupervised style transfer”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 553–561.

78. Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. “Domain-adversarial training of neural networks”. *The Journal of Machine Learning Research* 17:1, 2016, pp. 2096–2030.

79. L. A. Gatys, A. S. Ecker, and M. Bethge. “Image style transfer using convolutional neural networks”. In: *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE. 2016, pp. 2414–2423.

80. L. A. Gatys, A. S. Ecker, and M. Bethge. “Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks”. *arXiv preprint arXiv:1505.07376* 12, 2015.

81. L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman. “Controlling perceptual factors in neural style transfer”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

82. W. Ge, W. Huang, D. Dong, and M. R. Scott. “Deep Metric Learning with Hierarchical Triplet Loss”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 269–285.

83. G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens. “Exploring the structure of a real-time, arbitrary neural artistic stylization network”. *arXiv preprint arXiv:1705.06830*, 2017.

84. S. Gidaris, P. Singh, and N. Komodakis. “Unsupervised Representation Learning by Predicting Image Rotations”. In: *International Conference on Learning Representations*. 2018.

85. Girshick, Donahue, Darrell, and Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *CVPR*. 2014.

86. X. Glorot, A. Bordes, and Y. Bengio. “Deep sparse rectifier neural networks”. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 2011, pp. 315–323.

87. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

88. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

89. J. M. Gottwald and G. Gredebäck. "Infants' prospective control during object manipulation in an uncertain environment". *Experimental Brain Research* 233:8, 2015, pp. 2383–2390.

90. Y. Grandvalet and Y. Bengio. "Semi-supervised learning by entropy minimization". *Advances in neural information processing systems* 17, 2004, pp. 529–536.

91. J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. "Bootstrap your own latent-a new approach to self-supervised learning". *Advances in Neural Information Processing Systems* 33, 2020.

92. R. A. Güler, N. Neverova, and I. Kokkinos. "Densepose: Dense human pose estimation in the wild". *CVPR*, 2018.

93. S. Günel, H. Rhodin, D. Morales, J. H. Campagnolo, P. Ramdya, and P. Fua. "Deepfly3d, a deep learning-based approach for 3d limb and appendage tracking in tethered, adult Drosophila". *eLife*, 2019.

94. J. Guo and S. Gould. "Deep CNN ensemble with data augmentation for object detection". *arXiv preprint arXiv:1506.07224*, 2015.

95. R. Hadsell, S. Chopra, and Y. LeCun. "Dimensionality reduction by learning an invariant mapping". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. IEEE. 2006, pp. 1735–1742.

96. O. Halimi, O. Litany, E. Rodola, A. Bronstein, and R. Kimmel. "Self-supervised Learning of Dense Shape Correspondence". *CVPR*, 2019.

97. B. Hariharan, J. Malik, and D. Ramanan. "Discriminative decorrelation for clustering and classification". In: *ECCV*. 2012.

98. J. A. Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

99. B. Harwood, V Kumar, G Carneiro, I Reid, and T Drummond. "Smart mining for deep metric learning". In: *Proc. IEEE International Conference on Computer Vision (ICCV)*. 2017.

100. K. He, G. Gkioxari, and P. D. and. R. Girshick. "Mask R-CNN". *ICCV*, 2017.

101. K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9729–9738.

102. K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

103. X. He and S. Gould. "An exemplar-based CRF for multi-instance object segmentation". In: *CVPR*. IEEE. 2014.

104. O. J. Hénaff, A. Srinivas, J. De Fauw, A. Razavi, C. Doersch, S. Eslami, and A. van den Oord. "Data-Efficient Image Recognition with Contrastive Predictive Coding". In: *International Conference on Machine Learning*. 2019.

105. A. Hermans, L. Beyer, and B. Leibe. "In defense of the triplet loss for person re-identification". *arXiv preprint arXiv:1703.07737*, 2017.

106. J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe. "Deep clustering: Discriminative embeddings for segmentation and separation". In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE. 2016, pp. 31–35.

107. A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. "Image analogies". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM. 2001, pp. 327–340.

108. M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: *Advances in Neural Information Processing Systems*. 2017, pp. 6629–6640.

109. G. E. Hinton and R. R. Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". *Science* 313:5786, 2006, pp. 504–507. ISSN: 0036-8075. DOI: 10.1126/science.1127647. eprint: http://science.sciencemag.org/content/313/5786/504.full.pdf. URL: http://science.sciencemag.org/content/313/5786/504.

110. G. Hinton, O. Vinyals, and J. Dean. "Distilling the knowledge in a neural network". *arXiv preprint arXiv:1503.02531*, 2015.

111. G. E. Hinton and R. R. Salakhutdinov. "Reducing the dimensionality of data with neural networks". *science* 313:5786, 2006, pp. 504–507.

112. E. Hoffer and N. Ailon. "Deep metric learning using triplet network". In: *International Workshop on Similarity-Based Pattern Recognition*. Springer. 2015, pp. 84–92.

113. K. Hornik. "Approximation capabilities of multilayer feedforward networks". *Neural networks* 4:2, 1991, pp. 251–257.

114. C. Huang, C. Change Loy, and X. Tang. "Unsupervised learning of discriminative attributes and visual representations". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5175–5184.

115. C. Huang, C. C. Loy, and X. Tang. "Local similarity-aware deep feature embedding". In: *Advances in Neural Information Processing Systems*. 2016, pp. 1262–1270.

116. G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. "Deep networks with stochastic depth". In: *European conference on computer vision*. Springer. 2016, pp. 646–661.

117. X. Huang and S. Belongie. "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization". In: *ICCV*. 2017.

118. A. Hyvarinen and H. Morioka. "Unsupervised Feature Extraction by Time-Contrastive Learning and Nonlinear ICA". In: *Advances in Neural Information Processing Systems*. 2016, pp. 3765–3773.

119. S. Iizuka, E. Simo-Serra, and H. Ishikawa. "Globally and locally consistent image completion". *ACM Transactions on Graphics (TOG)* 36:4, 2017, p. 107.

120. B. Insider. *PLANET SELFIE: We're Now Posting A Staggering 1.8 Billion Photos Every Day*. `https://www.businessinsider.com/were-now-posting-a-staggering-18-billion-photos-to-social-media-every-day-2014-5`.

121. A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. "Label propagation for deep semi-supervised learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019, pp. 5070–5079.

122. A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. "Mining on Manifolds: Metric Learning without Labels". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

123. P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1125–1134.

124. *IUCN Red List of Threatened Species*. `https://www.iucn.org/resources/conservation-tools/iucn-red-list-threatened-species`.

125. T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi. "Unsupervised Learning of Object Landmarks through Conditional Image Generation". *NIPS*, 2018.

126. H. Jegou, M. Douze, and C. Schmid. "Product quantization for nearest neighbor search". *IEEE transactions on pattern analysis and machine intelligence* 33:1, 2011, pp. 117–128.

127. H. Jégou, M. Douze, C. Schmid, and P. Pérez. "Aggregating local descriptors into a compact image representation". In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE. 2010, pp. 3304–3311.

128. Y. Jing, Y. Liu, Y. Yang, Z. Feng, Y. Yu, and M. Song. "Stroke Controllable Fast Style Transfer with Adaptive Receptive Fields". *arXiv preprint arXiv:1802.07101*, 2018.

129. J. Johnson, M. Douze, and H. Jégou. "Billion-scale similarity search with GPUs". *arXiv preprint arXiv:1702.08734*, 2017.

130. J. Johnson, A. Alahi, and L. Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution". In: *European Conference on Computer Vision*. Springer. 2016, pp. 694–711.

131. S. Johnson and M. Everingham. "Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation". In: *BMVC*. 2010.

132. S. Johnson and M. Everingham. "Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation". In: *BMVC*. 2010.

133. S. Johnson and M. Everingham. "Learning Effective Human Pose Estimation from Inaccurate Annotation". *CVPR*, 2011.

134. L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit. "One model to learn them all". *arXiv preprint arXiv:1706.05137*, 2017.

135. A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. "End-to-end Recovery of Human Shape and Pose". *CVPR*, 2018.

136. A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik. "Learning 3D Human Dynamics From Video". *CVPR*, 2019.

137. P. Karashchuk. "lambdaloop/anipose: v0.5.0". *eLife*, 2019.

138. S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller. "Recognizing image style". *arXiv preprint arXiv:1311.3715*, 2013.

139. T. Karras, T. Aila, S. Laine, and J. Lehtinen. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *International Conference on Learning Representations*. 2018.

140. T. Karras, S. Laine, and T. Aila. "A style-based generator architecture for generative adversarial networks". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June, 2019, pp. 4396–4405. ISSN: 10636919. DOI: 10.1109/CVPR.2019.00453. arXiv: 1812.04948.

141. T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. "Analyzing and improving the image quality of stylegan". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8110–8119.

142. A. Kendall and Y. Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" *NIPS*, 2017.

143. S. Kim, D. Kim, M. Cho, and S. Kwak. "Proxy Anchor Loss for Deep Metric Learning". *arXiv preprint arXiv:2003.13911*, 2020.

144. W. Kim, B. Goyal, K. Chawla, J. Lee, and K. Kwon. "Attention-based ensemble for deep metric learning". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 736–751.

145. D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980*, 2014.

146. A. Kirillov, R. Girshick, K. He, and P. Dollár. "Panoptic feature pyramid networks". *CVPR*, 2019, pp. 6399–6408.

147. I. Kokkinos. "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6129–6138.

148. J. Krause, M. Stark, J. Deng, and L. Fei-Fei. "3D Object Representations for Fine-Grained Categorization". In: *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*. Sydney, Australia, 2013.

149. A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *NIPS*. 2012, pp. 1097–1105.

150. H. Larochelle, D. Erhan, and Y. Bengio. "Zero-data learning of new tasks." In: *AAAI*. Vol. 1. 2. 2008, p. 3.

151. G. Larsson, M. Maire, and G. Shakhnarovich. "Learning representations for automatic colorization". In: *European Conference on Computer Vision*. Springer. 2016, pp. 577–593.

152. LeCun, Boser, Denker, Henderson, Howard, Hubbard, and Jackel. "Backpropagation applied to handwritten zip code recognition". *Neural Comp.1* 4, 1989.

153. Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning". *nature* 521:7553, 2015, pp. 436–444.

154. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86:11, 1998, pp. 2278–2324.

155. D. D. Lee and H. S. Seung. "Learning the parts of objects by non-negative matrix factorization". *Nature* 401:6755, 1999, pp. 788–791.

156. D.-H. Lee. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks". In: *Workshop on Challenges in Representation Learning, ICML*. Vol. 3. 2013, p. 2.

157. H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. "Unsupervised learning of hierarchical representations with convolutional deep belief networks". *Communications of the ACM* 54:10, 2011, pp. 95–103.

158. V. Leon, N. Bonneel, G. Lavoue, and J.-P. Vandeborre. "Continuous semantic description of 3d meshes". *Computer & Graphics*, 2016.

159. C. Li and M. Wand. "Precomputed real-time texture synthesis with markovian generative adversarial networks". In: *European Conference on Computer Vision*. Springer. 2016, pp. 702–716.

160. D. Li, W.-C. Hung, J.-B. Huang, S. Wang, N. Ahuja, and M.-H. Yang. "Unsupervised visual representation learning by graph-based consistent constraints". In: *European Conference on Computer Vision*. Springer. 2016, pp. 678–694.

161. M. Li and Z.-H. Zhou. "SETRED: Self-training with editing". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2005, pp. 611–621.

162. S. Li, J. Li, W. Lin, and H. Tang. "Amur Tiger Re-identification in the Wild". *arXiv preprint arXiv:1906.05586*, 2019.

163. Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. "Diversified Texture Synthesis with Feed-forward Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

164. Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. "Universal style transfer via feature transforms". In: *Advances in Neural Information Processing Systems*. 2017, pp. 385–395.

165. Z. Li and J. Tang. "Weakly supervised deep metric learning for community-contributed image retrieval". *IEEE Transactions on Multimedia* 17:11, 2015, pp. 1989–1999.

166. Z. Li, J. Tang, and T. Mei. "Deep collaborative embedding for social image understanding". *IEEE transactions on pattern analysis and machine intelligence*, 2018.

167. J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang. "Visual attribute transfer through deep image analogy". *arXiv preprint arXiv:1705.01088*, 2017.

168. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. "Microsoft coco: Common objects in context". *ECCV*, 2014.

169. X. Lin, Y. Duan, Q. Dong, J. Lu, and J. Zhou. "Deep Variational Metric Learning". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 689–704.

170. D. C. Liu and J. Nocedal. "On the limited memory BFGS method for large scale optimization". *Mathematical programming* 45:1-3, 1989, pp. 503–528.

171. H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang. "Deep Relative Distance Learning: Tell the Difference Between Similar Vehicles". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2167–2175.

172. M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz. "Few-Shot Unsupervised Image-to-Image Translation". In: *IEEE International Conference on Computer Vision (ICCV)*. 2019.

173. W. Liu, Y. Wen, Z. Yu, and M. Yang. "Large-margin softmax loss for convolutional neural networks." In: *ICML*. Vol. 2. 3. 2016, p. 7.

174. Z. Liu, P. Luo, S. Qiua, X. Wang, and X. Tang. "DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

175. J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

176. M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. "SMPL: A skinned multi-person linear model". *ACM transactions on graphics (TOG)* 34:6, 2015, p. 248.

177. D. Lorenz, L. Bereska, T. Milbich, and B. Ommer. "Unsupervised Part-Based Disentangling of Object Shape and Appearance". *CVPR*, 2019.

178. D. G. Lowe. "Distinctive image features from scale-invariant keypoints". *International journal of computer vision* 60:2, 2004, pp. 91–110.

179. J. Lu, C. Xu, W. Zhang, L.-Y. Duan, and T. Mei. "Sampling Wisely: Deep Image Embedding by Top-k Precision Optimization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7961–7970.

180. F. Luan, S. Paris, E. Shechtman, and K. Bala. "Deep photo style transfer". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

181. F. Ma, U. Ayaz, and S. Karaman. "Invertibility of convolutional generative networks from partial measurements". *Advances in Neural Information Processing Systems* 31, 2018, pp. 9628–9637.

182. L. v. d. Maaten and G. Hinton. "Visualizing data using t-SNE". *Journal of machine learning research* 9:Nov, 2008, pp. 2579–2605.

183. J. MacQueen et al. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

184. M. Madrona. *CC BY-NC-SA 2.0*. `https://www.flickr.com/photos/andreatx/5586973009/`. 2017.

185. A. Mahendran and A. Vedaldi. "Understanding Deep Image Representations by Inverting Them". In: *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2015.

186. R. Mahjourian, M. Wicke, and A. Angelova. "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints". In: *Proc. CVPR*. 2018, pp. 5667–5675.

187. T. Malisiewicz, A. Gupta, and A. A. Efros. "Ensemble of exemplar-svms for object detection and beyond". In: *ICCV*. 2011, pp. 89–96.

188. H. Mao, M. Cheung, and J. She. "DeepArt: Learning Joint Representations of Visual Arts". In: *Proceedings of the 2017 ACM on Multimedia Conference*. ACM. 2017, pp. 1183–1191.

189. K. Mardia, J. Kent, and J. Bibby. "Multivariate analysis". *Probability and mathematical statistics. Academic Press*, 1979.

190. A. Mathis, P. Mamidanna, K. M. Cury, A. Taiga, V. N. Murthy, M. W. Mathis, and M. Bethge. "Deeplabcut: markerless pose estimation of user-defined body parts with deep learning". *Nature neuroscience*, 2018.

191. M. W. Mathis and A. Mathis. "Deep learning tools for the measurement of animal behavior in neuroscience". *arXiv preprint arXiv:1909.13868v2*, 2019.

192. L. McInnes, J. Healy, and J. Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". *arXiv preprint arXiv:1802.03426*, 2018.

193. N. McWilliams. *CC BY-NC-SA 2.0*. `https://www.flickr.com/photos/24256658@N06/17859787186/`. 2017.

194. T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. "Recurrent neural network based language model." In: *Interspeech*. Vol. 2. 2010, p. 3.

195. I. Misra and L. v. d. Maaten. "Self-supervised learning of pretext-invariant representations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6707–6717.

196. I. Misra, C. L. Zitnick, and M. Hebert. "Shuffle and learn: unsupervised learning using temporal order verification". In: *European Conference on Computer Vision*. Springer. 2016, pp. 527–544.

197. T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. "Virtual adversarial training: a regularization method for supervised and semi-supervised learning". *IEEE transactions on pattern analysis and machine intelligence* 41:8, 2018, pp. 1979–1993.

198. Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. "No fuss distance metric learning using proxies". In: *Proc. IEEE International Conference on Computer Vision (ICCV)*. 2017.

199. T. Nath, A. Mathis, A. C. Chen, A. Patel, and M. W. Bethge Matthias andd Mathis. "Using deeplab-cut for 3d markerless pose estimation across species and behaviors". *Nature protocols*, 2019.

200. L. Neumann, A. Zisserman, and A. Vedaldi. "Efficient Confidence Auto-Calibration for Safe Pedestrian Detection". *NIPS Workshop on Machine Learning for Intelligent Transportation Systems*, 2018.

201. N. Neverova, J. Thewlis, R. A. Güler, I. Kokkinos, and A. Vedaldi. "Slim DensePose: Thrifty Learning from Sparse Annotations and Motion Cues". *CVPR*, 2019.

202. A. Newell, K. Yang, and J. Deng. "Stacked hourglass networks for human pose estimation". *ECCV*, 2016.

203. Y.-H. Ng, Hausknecht, Vijayanarasimhan, Vinyals, Monga, and Toderici. "Beyond Short Snippets: Deep Networks for Video Classification". In: *CVPR*. 2015.

204. Niebles, Chen, and Fei-Fei. "Modeling temporal structure of decomposable motion segments for activity classification". In: *ECCV*. 2010.

205. K. Nigam and R. Ghani. "Analyzing the effectiveness and applicability of co-training". In: *Proceedings of the ninth international conference on Information and knowledge management*. 2000, pp. 86–93.

206. M. Noroozi and P. Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles". In: *European Conference on Computer Vision*. Springer. 2016, pp. 69–84.

207. M. Noroozi, H. Pirsiavash, and P. Favaro. "Representation learning by learning to count". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5898–5906.

208. D. Novotny, N. Ravi, B. Graham, N. Neverova, and A. Vedaldi. "C3DPO: Canonical 3D Pose Networks for Non-Rigid Structure From Motion". *ICCV*, 2019.

209. A. Odena, V. Dumoulin, and C. Olah. "Deconvolution and Checkerboard Artifacts". *Distill*, 2016. DOI: 10.23915/distill.00003. URL: http://distill.pub/2016/deconv-checkerboard.

210. H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. "Deep metric learning via lifted structured feature embedding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4004–4012.

211. M. Opitz, G. Waltner, H. Possegger, and H. Bischof. "BIER-Boosting Independent Embeddings Robustly". In: *International Conference on Computer Vision (ICCV)*. 2017.

212. M. Opitz, G. Waltner, H. Possegger, and H. Bischof. "Deep Metric Learning with BIER: Boosting Independent Embeddings Robustly". *arXiv preprint arXiv:1801.04815*, 2018.

213. T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. "Semantic image synthesis with spatially-adaptive normalization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2337–2346.

214. T. Park, J.-Y. Zhu, O. Wang, J. Lu, E. Shechtman, A. Efros, and R. Zhang. "Swapping Autoencoder for Deep Image Manipulation". *Advances in Neural Information Processing Systems* 33, 2020.

215. D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. "Context encoders: Feature learning by inpainting". In: *CVPR*. 2016, pp. 2536–2544.

216. H. Pirsiavash and D. Ramanan. "Parsing videos of actions with segmental grammars". In: *CVPR*. 2014.

217. T. Portenier, Q. Hu, A. Szabó, S. A. Bigdeli, P. Favaro, and M. Zwicker. "Faceshop: deep sketch-based face image editing". *ACM Transactions on Graphics (TOG)* 37:4, 2018, pp. 1–13.

218. M. C. Potter, B. Wyble, C. E. Hagmann, and E. S. McCourt. "Detecting meaning in RSVP at 13 ms per picture". *Attention, Perception, & Psychophysics* 76:2, 2014, pp. 270–279.

219. Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin. "SoftTriple Loss: Deep Metric Learning Without Triplet Sampling". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 6450–6458.

220. A. Radford, L. Metz, and S. Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". *arXiv preprint arXiv:1511.06434*, 2015.

221. I. Radosavovic, P. Dollar, R. Girshick, G. Gkioxari, and K. He. "Data distillation: Towards omni-supervised learning". *CVPR*, 2018.

222. Ramakrishna, Munoz, Hebert, Bagnell, and Sheikh. "Pose machines: Articulated pose estimation via inference machines". In: *ECCV*. 2014.

223. M. Rashid, X. Gu, and Y. J. Lee. "Interspecies Knowledge Transfer for Facial Keypoint Detection". *CVPR*, 2017.

224. E. Riloff and J. Wiebe. "Learning extraction patterns for subjective expressions". In: *Proceedings of the 2003 conference on Empirical methods in natural language processing*. 2003, pp. 105–112.

225. O. Rippel, M. Paluri, P. Dollar, and L. Bourdev. "Metric Learning with Adaptive Density Discrimination". *arXiv preprint arXiv:1511.05939*, 2015.

226. Rodriguez, Ahmed, and Shah. "Action mach a spatio-temporal maximum average correlation height filter for action recognition". In: *CVPR*. 2008.

227. K. Roth, B. Brattoli, and B. Ommer. "MIC: Mining Interclass Characteristics for Improved Metric Learning". In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2019.

228. K. Roth, T. Milbich, and B. Ommer. "PADS: Policy-Adapted Sampling for Visual Similarity Learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2020.

229. K. Roth, T. Milbich, S. Sinha, P. Gupta, B. Ommer, and J. P. Cohen. "Revisiting Training Strategies and Generalization Performance in Deep Metric Learning". *arXiv preprint arXiv:2002.08473*, 2020.

230. S. Roweis, G. Hinton, and R. Salakhutdinov. "Neighbourhood component analysis". *Advances in Neural Information Processing Systems (NIPS)* 17, 2005, pp. 513–520.

231. J. C. Rubio, A. Eigenstetter, and B. Ommer. "Generative regularization with latent topics for discriminative object recognition". *Pattern Recognition* 48:12, 2015, pp. 3871–3880.

232. S. L. Saavedra, P. van Donkelaar, and M. H. Woollacott. "Learning about gravity: segmental assessment of upright control as infants develop independent sitting". *Journal of Neurophysiology* 108:8, 2012, pp. 2215–2229.

233. M. Sajjadi, M. Javanmardi, and T. Tasdizen. "Regularization with stochastic transformations and perturbations for deep semi-supervised learning". *Advances in neural information processing systems* 29, 2016, pp. 1163–1171.

234. T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. "Improved techniques for training gans". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2234–2242.

235. S. Salti, F. Tombari, and L. D. Stefano. "Shot: Unique signatures of histograms for surface and texture description". *Computer Vision and Image Understanding*, 2014.

236. A. Sanakoyeu, M. A. Bautista, and B. Ommer. "Deep unsupervised learning of visual similarities". *Pattern Recognition* 78, 2018, pp. 331–343.

237. A. Sanakoyeu, V. Khalidov, M. S. McCarthy, A. Vedaldi, and N. Neverova. "Transferring Dense Pose to Proximal Animal Classes". In: *CVPR*. 2020.

238. A. Sanakoyeu, D. Kotovenko, S. Lang, and B. Ommer. "A Style-Aware Content Loss for Real-time HD Style Transfer". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 698–714.

239. A. Sanakoyeu, P. Ma, V. Tschernezki, and B. Ommer. "Improving Deep Metric Learning by Divide and Conquer". *Under review*, 2020.

240. A. Sanakoyeu, V. Tschernezki, U. Büchler, and B. Ommer. "Divide and Conquer the Embedding Space for Metric Learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 471–480.

241. F. Schroff, D. Kalenichenko, and J. Philbin. "Facenet: A unified embedding for face recognition and clustering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 815–823.

242. H. Schütze, C. D. Manning, and P. Raghavan. *Introduction to information retrieval*. Vol. 39. Cambridge University Press, 2008.

243. N. Shapovalova and G. Mori. "Clustered Exemplar-SVM: Discovering subcategories for visual recognition". In: *ICIP*. IEEE. 2015, pp. 93–97.

244. F. Shen, S. Yan, and G. Zeng. "Meta Networks for Neural Style Transfer". *arXiv preprint arXiv:1709.04111*, 2017.

245. Y. Shen, J. Gu, X. Tang, and B. Zhou. "Interpreting the latent space of gans for semantic face editing". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9243–9252.

246. Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand. "Style transfer for headshot portraits". *ACM Transactions on Graphics (TOG)* 33:4, 2014, p. 148.

247. Y. Shih, S. Paris, F. Durand, and W. T. Freeman. "Data-driven hallucination of different times of day from a single outdoor photo". *ACM Transactions on Graphics (TOG)* 32:6, 2013, p. 200.

248. E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. "Discriminative learning of deep convolutional feature point descriptors". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 118–126.

249. K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". *arXiv preprint arXiv:1409.1556*, 2014.

250. J. Sivic and A. Zisserman. "Video Google: A text retrieval approach to object matching in videos". In: *null*. IEEE. 2003, p. 1470.

251. L. Smith and M. Gasser. "The development of embodied cognition: Six lessons from babies". *Artificial life* 11:1-2, 2005, pp. 13–29.

252. K. Sohn. "Improved deep metric learning with multi-class n-pair loss objective". In: *Advances in Neural Information Processing Systems*. 2016, pp. 1857–1865.

253. H. O. Song, S. Jegelka, V. Rathod, and K. Murphy. "Deep metric learning via facility location". In: *Computer Vision and Pattern Recognition (CVPR)*. Vol. 8. 2017.

254. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". *The journal of machine learning research* 15:1, 2014, pp. 1929–1958.

255. Statista. *Hours of video uploaded to YouTube every minute as of May 2019*. https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/.

256. Statista. *Mobile broadband internet subscription rate in 2019, by region*. https://www.statista.com/statistics/370694/mobile-broadband-internet-penetration-region/.

257. Y. Suh, B. Han, W. Kim, and K. M. Lee. "Stochastic class-based hard example mining for deep metric learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7251–7259.

258. A. Tarvainen and H. Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results". In: *Advances in neural information processing systems*. 2017, pp. 1195–1204.

259. J. B. Tenenbaum, V. De Silva, and J. C. Langford. "A global geometric framework for nonlinear dimensionality reduction". *science* 290:5500, 2000, pp. 2319–2323.

260. J. B. Tenenbaum and W. T. Freeman. "Separating style and content with bilinear models". *Neural computation* 12:6, 2000, pp. 1247–1283.

261. J. Thewlis, H. Bilen, and A. Vedaldi. "Unsupervised learning of object landmarks by factorized spatial embeddings". *ICCV*, 2017.

262. J. Thewlis, H. Bilen, and A. Vedaldi. "Unsupervised object learning from dense invariant image labelling". *NIPS*, 2017.

263. J. Thewlis, S. Albanie, H. Bilen, and A. Vedaldi. "Unsupervised Learning of Landmarks by Descriptor Vector Exchange". *ICCV*, 2019.

264. tomosuke214. *CC BY-NC-SA 2.0*. https://www.flickr.com/photos/24256658@N06/14336165579/. 2014.

265. tomosuke214. *CC BY-NC-SA 2.0*. https://www.flickr.com/photos/24256658@N06/17859787186/. 2014.

266. A. Toshev and C. Szegedy. "Deeppose: Human pose estimation via deep neural networks". In: *CVPR*. 2014, pp. 1653–1660.

267. I. Triguero, S. Garcia, and F. Herrera. "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study". *Knowledge and Information Systems* 42:2, 2015, pp. 245–284.

268. P. Tseng. "Convergence of a block coordinate descent method for nondifferentiable minimization". *Journal of optimization theory and applications* 109:3, 2001, pp. 475–494.

269. D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky. "Texture Networks: Feed-forward Synthesis of Textures and Stylized Images." In: *ICML*. 2016, pp. 1349–1357.

270. D. Ulyanov, A. Vedaldi, and V. Lempitsky. "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis". In: *Proc. CVPR*. 2017.

271. D. Ulyanov, A. Vedaldi, and V. Lempitsky. "Instance Normalization: The Missing Ingredient for Fast Stylization". *arXiv preprint arXiv:1607.08022*, 2016.

272. P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, and K. Weinberger. "Deep feature interpolation for image content changes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7064–7073.

273. User:Colin. *CC BY-NC-SA 2.0*. https://www.flickr.com/photos/user-colin/8913175473/. 2013.

274. E. Ustinova and V. Lempitsky. "Learning deep embeddings with histogram loss". In: *Advances in Neural Information Processing Systems*. 2016, pp. 4170–4178.

275. P. Vincent, A. de Brébisson, and X. Bouthillier. "Efficient exact gradient update for training deep networks with very large sparse targets". In: *Advances in Neural Information Processing Systems*. 2015, pp. 1108–1116.

276. P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. "Extracting and composing robust features with denoising autoencoders". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 1096–1103.

277. C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. *The Caltech-UCSD Birds-200-2011 Dataset*. Technical report CNS-TR-2011-001. California Institute of Technology, 2011.

278. H. Wang, X. Liang, H. Zhang, D.-Y. Yeung, and E. P. Xing. "ZM-Net: Real-time Zero-shot Image Manipulation Network". *arXiv preprint arXiv:1703.07255*, 2017.

279. H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. "Cosface: Large margin cosine loss for deep face recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5265–5274.

280. J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin. "Deep metric learning with angular loss". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE. 2017, pp. 2612–2620.

281. J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. "Learning fine-grained image similarity with deep ranking". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 1386–1393.

282. M. Wang and W. Deng. *Deep Face Recognition: A Survey*. 2018. arXiv: 1804.06655 [cs.CV].

283. T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. "High-resolution image synthesis and semantic manipulation with conditional gans". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8798–8807.

284. X. Wang, R. Girshick, A. Gupta, and K. He. "Non-local neural networks". *CVPR*, 2018.

285. X. Wang and A. Gupta. "Unsupervised learning of visual representations using videos". In: *ICCV*. 2015.

286. X. Wang, G. Oxholm, D. Zhang, and Y.-F. Wang. "Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

287. X. Wang, Y. Hua, E. Kodirov, G. Hu, R. Garnier, and N. M. Robertson. "Ranked list loss for deep metric learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5207–5216.

288. X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott. "Multi-similarity loss with general pair weighting for deep metric learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5022–5030.

289. Y.-X. Wang and M. Hebert. "Learning from Small Sample Sets by Combining Unsupervised Meta-Training with CNNs". In: *Advances in Neural Information Processing Systems*. 2016, pp. 244–252.

290. S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. "Convolutional pose machines". *CVPR*, 2016.

291. K. Q. Weinberger and L. K. Saul. "Distance metric learning for large margin nearest neighbor classification". *Journal of Machine Learning Research* 10:Feb, 2009, pp. 207–244.

292. Y. Wen, K. Zhang, Z. Li, and Y. Qiao. "A discriminative feature learning approach for deep face recognition". In: *European conference on computer vision*. Springer. 2016, pp. 499–515.

293. M. J. Wilber, C. Fang, H. Jin, A. Hertzmann, J. Collomosse, and S. Belongie. "BAM! The Behance Artistic Media Dataset for Recognition Beyond Photography". In: *The IEEE International Conference on Computer Vision (ICCV)*. 2017.

294. P. Wilmot, E. Risser, and C. Barnes. "Stable and controllable neural texture synthesis and style transfer using histogram losses". *arXiv preprint arXiv:1701.08893*, 2017.

295. P. Wohlhart and V. Lepetit. "Learning descriptors for object recognition and 3d pose estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3109–3118.

296. C.-Y. Wu, R Manmatha, A. J. Smola, and P. Krähenbühl. "Sampling matters in deep embedding learning". In: *Proc. IEEE International Conference on Computer Vision (ICCV)*. 2017.

297. Z. Wu, Y. Xiong, X. Y. Stella, and D. Lin. "Unsupervised Feature Learning via Non-Parametric Instance Discrimination". In: *Proc. CVPR*. 2018, pp. 3733–3742.

298. H. Xia, S. C. Hoi, R. Jin, and P. Zhao. "Online multiple kernel similarity learning for visual search". *TPAMI* 36:3, 2014, pp. 536–549.

299. Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. "Self-training with noisy student improves imagenet classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10687–10698.

300. Y. Xu and W. Yin. "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion". *SIAM Journal on imaging sciences* 6:3, 2013, pp. 1758–1789.

301. H. Xuan, R. Souvenir, and R. Pless. "Deep Randomized Ensembles for Metric Learning". *arXiv preprint arXiv:1808.04469*, 2018.

302. H. Xuan, A. Stylianou, and R. Pless. "Improved embeddings with easy positive triplet mining". In: *The IEEE Winter Conference on Applications of Computer Vision*. 2020, pp. 2474–2482.

303. Y. Xue, T. Xu, H. Zhang, R. Long, and X. Huang. "SegAN: Adversarial Network with Multi-scale L_1 Loss for Medical Image Segmentation". *arXiv preprint arXiv:1706.01805*, 2017.

304. I. Z. Yalniz, H. Jégou, K. Chen, M. Paluri, and D. Mahajan. "Billion-scale semi-supervised learning for image classification". *arXiv preprint arXiv:1905.00546*, 2019.

305. I. Z. Yalniz, H. Jegou, K. Chen, M. Paluri, and D. Mahajan. "Billion-scale semi-supervised learning for image classification". *arXiv preprint arXiv:1905.00546v1*, 2019.

306. H. Yang, R. Zhang, and P. Robinson. "Human and Sheep Facial Landmarks Localisation by Triplet Interpolated Features". *WACV*, 2015.

307. J. Yang, D. Parikh, and D. Batra. "Joint unsupervised learning of deep representations and image clusters". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5147–5156.

308. L. Yang, Q. Song, Z. Wang, and M. Jiang. "Parsing R-CNN for Instance-Level Human Analysis". *CVPR*, 2018.

309. D. Yarowsky. "Unsupervised word sense disambiguation rivaling supervised methods". In: *33rd annual meeting of the association for computational linguistics*. 1995, pp. 189–196.

310. B. Yu and D. Tao. "Deep Metric Learning With Tuplet Margin Loss". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 6490–6499.

311. Y. Yuan, K. Yang, and C. Zhang. "Hard-aware deeply cascaded embedding". In: *Proc. IEEE International Conference on Computer Vision (ICCV)*. 2017.

312. A. L. Yuille and A. Rangarajan. "The concave-convex procedure (CCCP)". *Neural computation* 15:4, 2003, pp. 915–936.

313. S. Zagoruyko and N. Komodakis. "Learning to compare image patches via convolutional neural networks". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

314. A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. "Taskonomy: Disentangling task transfer learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3712–3722.

315. A. Zhai and H.-Y. Wu. "Making classification competitive for deep metric learning". *arXiv preprint arXiv:1811.12649*, 2018.

316. R. Zhang, P. Isola, and A. A. Efros. "Colorful image colorization". In: *European conference on computer vision*. Springer. 2016, pp. 649–666.

317. W. Zhang, M. Zhu, and K. G. Derpanis. "From actemes to action: A strongly-supervised representation for detailed action understanding". *ICCV*, 2013.

318. Y. Zhang, Y. Guo, Y. Jin, Y. Luo, Z. He, and H. Lee. "Unsupervised Discovery of Object Landmarks as Structural Representations". *CVPR*, 2018.

319. Y. Zhao, Z. Jin, G.-j. Qi, H. Lu, and X.-s. Hua. "An Adversarial Approach to Hard Triplet Generation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 501–517.

320. S. Zheng, Y. Song, T. Leung, and I. Goodfellow. "Improving the robustness of deep neural networks via stability training". In: *Proceedings of the ieee conference on computer vision and pattern recognition*. 2016, pp. 4480–4488.

321. W. Zheng, Z. Chen, J. Lu, and J. Zhou. "Hardness-Aware Deep Metric Learning". *arXiv preprint arXiv:1903.05503*, 2019.

322. B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. "Learning deep features for scene recognition using places database". In: *Advances in neural information processing systems*. 2014, pp. 487–495.

323. G. Zhou, S. Dulloor, D. G. Andersen, and M. Kaminsky. "EDF: ensemble, distill, and fuse for easy video labeling". *arXiv preprint arXiv:1812.03626*, 2018.

324. T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. "Unsupervised learning of depth and ego-motion from video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1851–1858.

325. J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. "Generative visual manipulation on the natural image manifold". In: *European conference on computer vision*. Springer. 2016, pp. 597–613.

326. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". In: *IEEE International Conference on Computer Vision*. 2017.

327. X. J. Zhu. *Semi-supervised learning literature survey*. Technical report. University of Wisconsin-Madison Department of Computer Sciences, 2005.

328. S. Zuffi, A. Kanazawa, T. Berger-Wolf, and M. J. Black. "Three-D Safari: Learning to Estimate Zebra Pose, Shape, and Texture from Images "In the Wild"". *ICCV*, 2019.

329. S. Zuffi, A. Kanazawa, and M. J. Black. "Lions and Tigers and Bears: Capturing Non-Rigid, 3D, Articulated Shape from Images". *ICCV*, 2018.

330. S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black. "3D Menagerie: Modeling the 3D shape and pose of animals". *CVPR*, 2017.