Visual Recognition From Structured Supervision

Rodrigo Santa Cruz

A thesis submitted for the degree of
Doctor of Philosophy
The Australian National University

December 2019

Except where otherwise indicated, this thesis is my own original work.

Rodrigo Santa Cruz
13 December 2019

To my beloved parents, without whom I would never have started, and to my wife and son, without whom I would never have finished.

Acknowledgments

I feel truly privileged and proud to have Stephen Gould as my PhD. supervisor. Steve, I would like to thank you for all the support, encouragement and mentorship over the past four years. Your technical clarity, professional standards, and research insight have shaped me as a scientist, while your casual style, modesty and sensibility have inspired me as a human being. Thanks for the science and life lessons.

I am also very grateful for my PhD advisors Basura Fernando and Anoop Cherian. I thank you both for always pointing me to valuable resources and research directions. Basura's eye for good research problems continue to be an important inspiration to build my research agenda. Anoops's technical background and rigour motivates me to keep studying and improving myself. I believe none of this work would be possible without your participation.

I would like to thank The Australia National University (ANU), The Centre of excellence for Robotic Vision (ACRV), and their staff. Thanks for supporting my research and provide a multitude of opportunities. In special, I thank Carol Taylor for organizing all of my trips, afternoon tea, group lunch, events, and for being such a nice person. Likewise, I thank professors Hongdong Li, Richard Hartley, and Robert Mahony. I consider myself very fortunate to have had the opportunity to share the workplace with these brilliant professionals. To my fellow PhD students, Arif Chowdhury, Cedric Scheerlinck, Christian Rodriguez, Dylan Campbell, Edison Guo, Jean-Luc Stevens, Jue Wang, Kartik Gupta, Mina Henein, Peter Anderson, Shihao Jiang, Suryansh Kumar, Tony Zhang, Xin Yu, Yon Hon Ng, and Zheyu Zhuang, a huge thanks for making my time at ANU very enjoyable.

Most importantly, I would like to thank my wonderful family. To my father Lincoln (*in memoriam*), I thank you for your support and encouragement. There are two moments that I want to leave registered. First, it is when I was accepted in the PhD. program and you, without even question, got a loan to pay my visa and health insurance costs allowing me to peruse my dreams. Second, it was when you got very sick and called from Brazil to say that I should not stop my PhD. independently what would happen with you. Unfortunately, you passed away in the first year of my PhD., but I hope you are proud with what I have accomplished. To my mother Lusanira, I thank you for inspiring me to follow the academic career and for your unconditional affection. There was not a single day that you have not called from Brazil to just ask "are you ok?", even having to wake up very early everyday. To my wife Flavia, thanks for your love and for crossing the globe to join me in this challenging PhD student life. You have all of the merits in raising our family. While all I had to do was write papers and code experiments from my comfortable desk, you had to fight for jobs, take care of our son, and provide a health environment for

us. To my son Pedro, I thank for bringing a lot of joy to my days with laughs, plays and discoveries. His curiosity and tenacity without limits make me very proud – it is probably the best recipe for a scientist. To my sisters Nara and Mariana, and my nephew Caua, I thank you for cheering me out and update me about all of the news from Brazil in our daily chats.

Abstract

Visual recognition of semantically meaningful entities like objects, actions, and poses in images and videos is a long standing goal of computer vision. In the last decades, we have seen progress towards this goal with the development of machine learning models that leverage huge volumes of human annotated data to perform very accurate recognition of a predefined set of visual entities. However, moving forward, this approach presents significant limitations since annotated datasets are expensive to collect, only contemplate a small fraction of the real world, and the labelling task itself is prone to inconsistency and ambiguity on denoting visual entities. Therefore, this reliance on exhaustive labeling is indeed the key obstacle to the fulfillment of such a goal.

In this thesis, we propose methods that reduce the need for human supervision by leveraging the structure in the visual world targeting visual recognition in difficult scenarios where annotated data is scarce and the visual concepts are innumerable or ambiguous. We call this approach *structured supervised learning* and explore three instances of structured supervision. We start by exploring structure in the output of visual recognition models to learn better models for ranking images according to a predefined criteria, like the visual attribute "smiling". Towards this end, we first cast the problem of image ranking as the problem of predicting the correct permutation of a set of images. Then, we leverage the geometrical structure of permutation matrices in order to learn accurate image rankers.

Next, we explore the self-supervision that can be extracted from input visual data itself. More specifically, unlabeled visual data itself encompasses rich spatial (and temporal) structure that can be explored in order to learn representations useful for generic visual recognition tasks. In contrast to human annotators, this form of self-supervision is cheap and abundant. Following this idea, we use the spatial layout of objects as a supervisory signal to learn transferable image representations from unlabeled data for object recognition tasks such as image classification, object detection, and object segmentation.

Last, we observe that the visual world is fundamentally compositional and complex visual concepts are structured compositions of simple primitive concepts. We build in this insight and formulate frameworks to unambiguously describe and recognize compositional visual concepts in images and videos by exploring structural information in model space. More specifically, we classify objects from boolean expressions of object attributes and infer activities from regular expressions of atomic actions. The proposed models can predict unseen, subcategories and specific instances of complex visual concepts without any additional annotation effort, resulting in a more feasible direction to fulfill the visual recognition goal.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"The revolution will not be supervised."

Yann LeCun, 2018

One of the most impressive human skills is the capability to recognize and understand the complex visual world by extracting semantically meaningful information from images and videos at first glance. For instance, humans can localize objects, identify actions, or say exactly which pixels belong to each object in an effortless, even unconscious manner [DiCarlo et al., 2012]. In computer vision and artificial intelligence, an important ongoing research topic, named visual recognition, is to endow computers with such an ability [Russell and Norvig, 2016; Szeliski, 2010]. This topic is very important since it is an essential step towards the development of autonomous agents (i.e., machines) that can reason and act in their environments. For example, self-driving vehicles need to localize objects such as cars, people, and traffic signals in order to safely navigate in the environment [Zhu et al., 2016; Pinggera et al., 2016], eliminating hazards in construction sites requires to recognize unsafe conditions and acts in video footage [Seo et al., 2015], and automatic diagnosing cancer in MR images consists of labeling image regions as infected or non-infected tissue [Nie et al., 2016; Cheng et al., 2016].

More specifically, visual recognition refers to the act of classifying, localizing, segmenting or even comparing semantic meaningful visual entities like objects, actions and poses in visual inputs like images and videos. This problem is as challenging as important, since we need to deal with view point, scale, occlusion, lighting, and appearance variations of every visual entity to be recognized. Despite these challenges, many computer vision researchers undertook the task of developing vision systems for visual recognition [Marr and Nishihara, 1978; Minsky, 1988; Viola and Jones, 2001; Lowe, 2004; Dalal and Triggs, 2005; Felzenszwalb et al., 2010; Dollár et al., 2014]. In a historical perspective, we highlight the pioneering work of Marr and Nishihara [1978] on proposing a computational theory for vision, the inspiring work of Minsky [1988] who built a robotic arm with mounted camera to build with children's blocks in the early 1970s, and Viola and Jones [2001] who, arguably, first brought visual recognition out of the lab environment by developing a face detection framework able to achieve satisfactory detection rates in real-time. These early

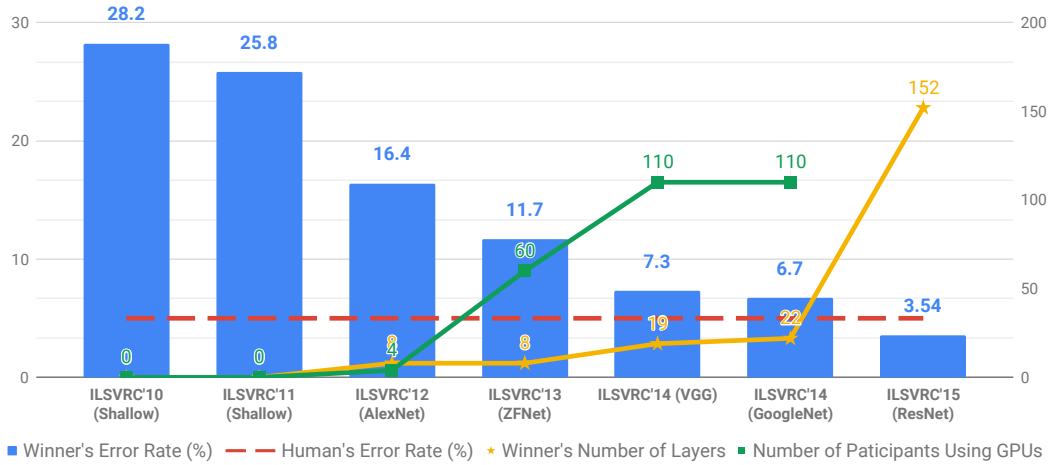


Figure 1.1: The results of the ImageNet Challenge along the years. It is important to observe the steady decreasing in the winner's error rate, the increasing in the number of participants using GPUs, and the increasing in the number of layers used by the winner's deep learning model after ILSVRC'12. These facts reflect the level of improvement brought by deep learning models to visual recognition problems.

works, however, rely on hand-crafted algorithms and failed to scale to large scale recognition problems and applications.

More recently, systems using a fully data-driven approach have made significant progress in visual recognition [LeCun et al., 2015]. These systems are known as deep learning models and consist of fully supervised high capacity machine learning models that can leverage huge volumes of human annotated data in order to perform accurate visual recognition directly from pixels. The level of improvement brought by deep learning models can be illustrated with the results of the ImageNet Challenge [Russakovsky et al., 2015] along the years. As shown in Figure 1.1, initially, the existent algorithms had an error rate above to 25%, until 2012 when Krizhevsky et al. [2012] proposed a deep neural network approach, named AlexNet, and dropped the error rate by approximately 10%. In the subsequent years, we saw an increasing number of participants using deep nets which resulted in a steady rate of improvement that eventually surpassed human level accuracy by 2% in 2015 with the ResNet model [He et al., 2016]. Similar trend can be observed in other visual recognition problems like visual question answering [Goyal et al., 2017] and object detection [Everingham et al., 2015]. These accomplishments provoked the dissemination of deep learning techniques in computer vision and now most of the vision systems in academia and industry are based on deep learning. For instance, Google has increased the presence of deep learning algorithms in their products exponentially [Dean, 2017].

The major requirement for applying deep learning models in a diverse range of visual recognition applications is the existence of large scale human annotated datasets. Consequently, researchers and practitioners have been spending most of their time curating large datasets and training even larger deep learning models in order to succeed in their desired application. However, having in mind the human

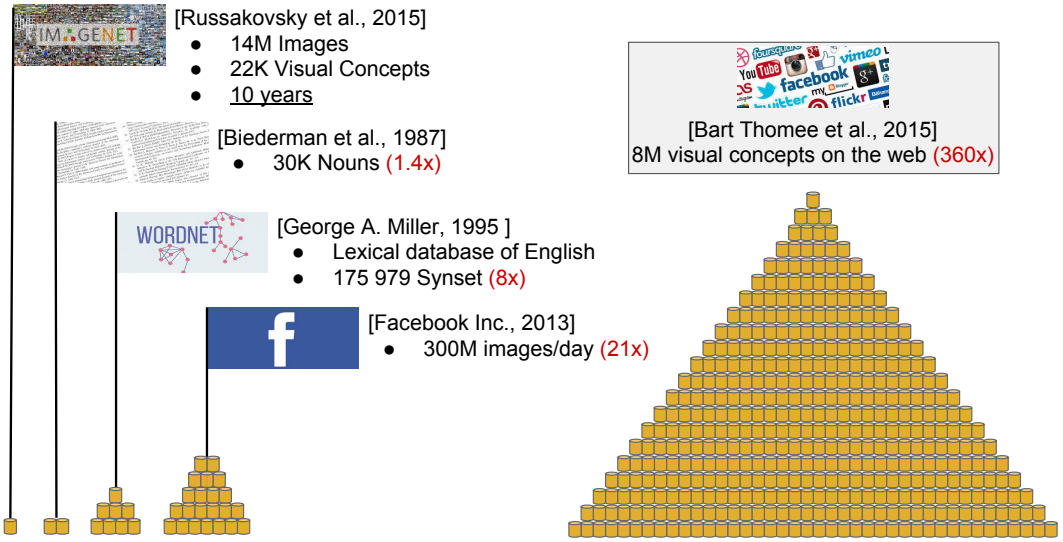


Figure 1.2: Comparing the Image-Net, one of the largest publicly available dataset for visual recognition, with conservative estimates of the amount of data we produce and visual concepts that we can recognize found in the literature. As you can see, it is not feasible to collect and annotate data for all concepts that a human can recognize.

visual system capabilities, is the way to solve visual recognition to collect data for all the things we want to recognize?

The enthusiasts of deep learning may argue that data collection is not a problem nowadays, due to the popularization of crowdsourcing marketplaces like Amazon Mechanical Turk (AMT) [Sorokin and Forsyth, 2008] and the dissemination of data collection companies around the world. They can even state that the human performance on visual recognition tasks can be always achieved by scaling out the data curation and training very large models since deep learning models seem to improve as more annotated data is provided [Sun et al., 2017]. Such statements may be true for applications in stationary environments where the number of concepts and the appearance variation of their instances are limited like the visual recognition challenges that attract a lot of attention in the major computer vision conferences. However, it is not feasible to collect and annotate data for all concepts that a human can recognize.

As you can observe in Figure 1.2, the ImageNet dataset that is an effort with almost 10 years only contemplate annotated images for approximately 12% of WordNet [Miller, 1995], which is just a very modest lower bound for the concepts that we can recognize. This discrepancy becomes even worse if we compare to the amount of data we generate in the social media. For instance, the Facebook image collection increases 300 million images per day which is more than 21 ImageNets daily [Facebook Inc., 2013]. In addition, some types of annotations are very laborious to collect like pixel-level labelling or can just be performed by a domain specialist like medical images. Therefore, even aided by the nowadays data curation technologies, we are not able to generate human annotated datasets in scale compatible with the richness

(a) (b) (c)

Figure 1.3: Examples of structural information in the outputs of visual recognition models that can help a learner to perform accurate predictions. In the human pose estimation problem shown in Figure 1.3(a), the prediction of the head position should not be very far from the prediction of the shoulders position. In the segmentation map shown in Figure 1.3(b) the labels colored as green and gray should be "grass" and "sky" since they often appear in the bottom and top regions of the images, respectively. Figure 1.3(c) shows the geometric structure of permutation and doubly stochastic matrices that can be used to prune infeasible solutions (e.g., red dot) for ranking problems as shown in Chapter 3.

depicted in the real world in which humans perform visual recognition with mastery.

In addition, the labelling process is problematic by itself being subject to artificial bias, inconsistencies, and ambiguities [Torralba and Efros, 2011]. For instance, we tend to curate datasets with artificial distributions in order to make it convenient for us to train and test recognition algorithms. These distributions end up biasing our model and degrading its performance in the real world distribution. This problem is even more alarming with deep learning models that are known to be easily fooled by out-of-distribution samples [Alcorn et al., 2018; Tian et al., 2018]. For instance, this generalization problem may be the cause of the accident involving a Tesla autopilot car that failed to recognize a white truck against a bright-lit sky – an unusual view that might be out of the training distribution – it crashed into the truck, killing the driver [Lambert, 2016; Tesla Motors, Reuters, 2016]. Similar tragedy also happened with the self-driving Uber car that killed a pedestrian [Grabar, 2018]. Another problem with the labelling process is the inconsistencies and ambiguities generated by different annotators. For instance, the exact moment an action starts and ends in a video clip is subject to the annotators which ends up producing datasets with a lot of ambiguities that make the learning ineffective. Therefore, even if we were able to set up an army of annotators to produce a dataset for the entire visual world, we still would have problems to apply the deep learning approach or any other fully supervised machine learning algorithm.

Therefore, the reliance on extensive human supervision is indeed the key obstacle to have visual recognition systems operating as well as humans. We, along with many other contemporary researchers, [Doersch, 2016; Vondrick, 2017; Misra, 2018], acknowledge the importance of these fully supervised approaches, but argue that we should explore other sources of supervision in order to perform visual recognition

(a) (b)

Figure 1.4: Examples of structural information in visual inputs of visual recognition models. Figure 1.4(a), which is a courtesy of Zhang et al. [2016], shows that the color of images can be predicted by the general context of the depicted scene which can be used to learn image representations. In similar fashion, in Chapter 4, we demonstrate that the proposed visual permutation learning model, can also be used to learn image representations by solving jigsaws like the one in Figure 1.4(b).

in difficult scenarios where annotated data is scarce and the visual concepts are innumerable or ambiguous. In this thesis, we propose methods that reduce the need for extensive human supervision by leveraging the structure in the visual world. We call this approach visual recognition from structured supervision and explore the inherent structure that exists in the outputs, inputs, and models for visual recognition.

Let us start by analyzing the outputs of visual recognition systems like class labels, bounding boxes predictions, human joints locations and segmentation masks. These outputs are highly structured and provide useful priors that can help a learner to perform accurate predictions as shown in Figure 1.3. For instance, the prediction of the head position should not be very far from the prediction of the shoulders position in human pose estimation problems due to deformations allowed by our body. Some visual concepts often appear in certain regions of the image in image segmentation problems. In Chapter 3, we follow these ideas and leverage the structure on outputs to learn visual permutations proposing the *visual permutation learning* framework. More specifically, we encode the ground-truth of ranking tasks as permutation matrices and make use of their geometric structure to prune infeasible solutions (e.g., the red dot in the Figure 1.3(c)) for our learning and inference algorithms. Such an approach provides more accurate rankers using the same amount of annotated data than state-of-the-art algorithms for image ranking.

Another rich source of structure and visual priors that can be exploited in order to better solve challenging computer vision problems is the input visual data itself. For example, a large collection of unlabelled images depict contextual information about its visual content, while unlabelled videos exhibit temporal coherence on the development of an action and the deformation of the human body. Inspired by these ideas and encouraged by the generality of the proposed visual permutation learning framework, Chapter 4 presents a self-supervised approach to learn transferable image features by leveraging the spatial structure and other visual priors existent in unlabelled images. Towards this end, we define tasks resembling image jigsaws

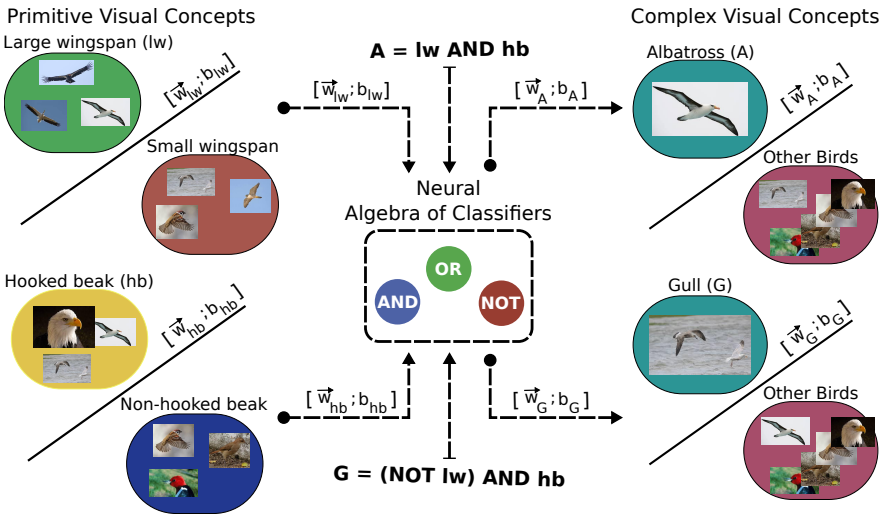


Figure 1.5: Illustration of the proposed *neural algebra of classifiers*. Given classifiers for primitive visual concepts such as hooked beak and large wingspan, we can compose classifiers for complex concepts such as gull and albatross that are represented by boolean expressions of these primitives.

(see Figure 1.4(b)) and demonstrate that the proposed visual permutation learning framework trained to solve these puzzles also learns to produce useful image representations for object recognition without human supervision. We evaluate this hypothesis on transfer learning experiments using well known object classification, detection and segmentation benchmarks. Note that the proposed approach can mitigate the need for large-scale human annotated datasets for some applications by pre-training deep models on the proposed self-supervised task. The proposed approach outperforms other contemporary self-supervised representation learning techniques like image colorization shown in Figure 1.4(a).

In the same fashion of visual input and output spaces, the model space is also highly structured. Intuitively, a dog's classifier should be more similar to a fox's classifier than an elephant's classifier since dogs and foxes are more visually similar than dogs and elephants as described by Misra et al. [2017]. Chapter 5 leverages this similarity and other visual priors in classifier space and develop an algebra for combining concept classifiers, named *neural algebra of classifiers*. More specifically, we first see complex visual concepts as compositions of simple visual concepts according to well-defined rules. Then, we develop neural network modules which can learn to compose classifiers according to these composition rules. This approach allows us to produce classifiers for any complex concept expressed as boolean expression of primitive concepts even without a single training sample of such a concept. As illustrated in Figure 1.5, using a classifiers for hooked beak and large wingspan, we can compose a classifier for albatrosses without having images of such a bird in our training data. Likewise, we can compose a classifier for gulls without training data by expressing such a concept as birds with hooked beak that does not have large wings. Therefore, such a models allows us to recognize a huge number of visual concepts without additional annotation effort.

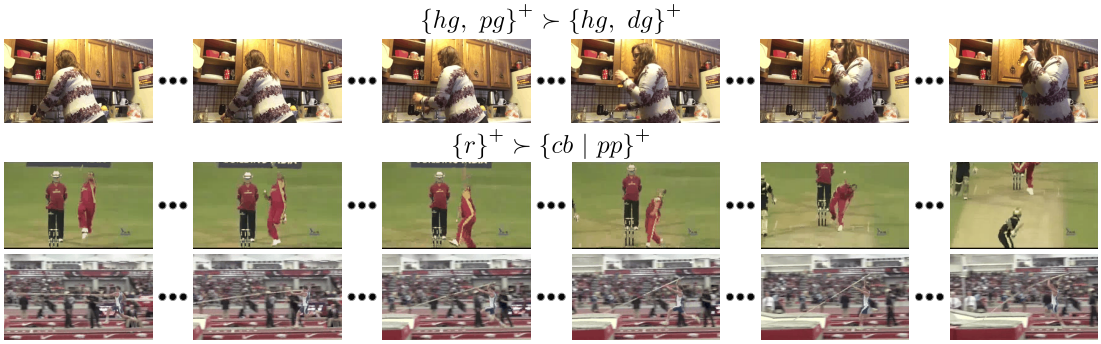


Figure 1.6: Examples of *action patterns* and corresponding ground-truth videos. In the first row, we see examples of concurrent ($\{\ldots\}$), sequential ($\succ$), and recursive ($+$) actions where the woman depicted is holding a glass (hg) and pouring water into the glass (pg) simultaneously, and then she drinks from the glass (dg) while holding the glass. In the last two rows, we see an example of alternated ($|$) actions where the desired action pattern starts with running (r) and finishes with someone either bowling (cb) or pole vault planting (pp).

In addition to the structure existent in the model space, we can also leverage existing models to perform more expressive tasks which would require an infeasible amount of well-trained annotators in order to apply any supervised approach. For instance, recognizing activities in videos using a supervised deep learning model would involve curating a dataset for the very log tailed distribution of activities like cooking meals and group activities which is prohibitive. Chapter 6 instead presents a probabilistic framework to unambiguously describe and accurately infer these complex activities in videos using only existing simple action classifiers without annotating additional data or training new machine learning models. Similar to the neural algebra of classifier framework, we first describe complex activities as regular expressions of simple primitive actions named *action patterns*. Then, we develop a probabilistic framework that can recognize these regular expressions in videos. Figure 1.6 shows examples of such an approach. It is important to emphasize that both this proposed inference procedure and the neural algebra of classifiers framework can scale-up recognition systems to a very huge number of visual concepts without any additional annotation effort like humans do.

In summary, the methods proposed in this thesis provide more accurate, extensible, and interpretable vision models using much less human supervision than black-box fully supervised deep learning approaches. We also tackle visual recognition in a more realistic scenario where the visual concepts are not defined a priori and we can not annotate large volumes of data for them. Therefore, this thesis presents a more feasible direction towards the development of visual recognition algorithms with the capabilities of the human visual system. The following sections in the current chapter provide a summary of our main contributions, outline the remainder of this dissertation and enumerate our relevant previously published papers.

1.1 Thesis Contributions

This thesis contributes to visual recognition proposing methods that reduce the need for human supervision by leveraging the structure in the visual data. We also focus on visual recognition in difficult settings where annotated data is scarce or the number of visual concepts is innumerable. Our main contributions can be described as follows:

1. Visual Permutation Learning. Sorting sequences of images according to a predefined criterion is an important part of many computer vision problems such as image search, person re-identification, and active learning. We propose to cast this problem as predicting the permutation that recovers the correct order for a shuffled sequence of images. Towards this end, we propose to represent the orderings by permutation matrices and develop a learning framework that can explore the geometry of these matrices and its surrogates. Incorporating the inherent structure of permutation matrices can avoid the learner searching over impossible solutions, thereby leading to faster convergence and accurate predictions.

2. Self-supervised Learning By Permuting Image Regions. We argue that human annotators are not the only source of supervision that can guide the learning of visual representations as in standard deep learning models. In fact, unlabeled visual data itself encompasses rich spatial (and temporal) structure that can be explored in order to learn representations useful for visual recognition tasks. In contrast to human annotators, this form of self-supervision is cheap and abundant. Therefore, using the proposed visual permutation learning framework, we formulate a pretext task similar to image jigsaws and show that a model trained to solve such a self-supervised task learns image representations useful for object recognition tasks such as image classification, object detection, and object segmentation.

3. Neural Algebra of Classifiers. We build on the insight that visual concepts are fundamentally compositional and propose an algebra for combining concept classifiers according to boolean algebra operators. More specifically, we develop neural network modules which can learn to compose classifiers according logical operators leveraging visual priors such as correlations, co-occurrences and contextuality between visual primitives. The proposed framework is able to produce classifiers for any complex concept expressed as a boolean expression of primitive concepts. Different from existing works where new concepts require annotating data and retraining machine learning models, the proposed model can predict unseen, subcategories and specific instances of complex visual concepts without any additional annotation effort or retraining.

4. Inferring Action Patterns in Videos. Existing algorithms for action recognition either recognize singleton actions from a fixed vocabulary of actions or retrieve videos using natural language sentences which are often incomplete, vague, and ambiguous descriptions of the activity of interest. We instead build on the insight that complex activities are fundamentally action patterns and develop a probabilistic inference

framework to unambiguously describe and efficiently recognize activities in videos exploring existing primitive action classifiers. The proposed approach allows us to unambiguously distinguish between fine-grained actions, retrieve very specific activity instances, and recognize complex composites of actions that may not have a single training sample.

1.2 Thesis Organization

To facilitate the presentation of this material, we organize this thesis in seven chapters including this introduction. The summary of the remaining chapters as well as their relevant publications are described bellow:

Chapter 2: Background. This chapter is organized in two parts. In the the first part, we review the current trend on visual recognition focusing on deep learning models. More specifically, we formulate a generic visual recognition problem, discuss a data-driven approach for such a problem, and present the current state-of-the-art models for the visual recognition tasks relevant for this thesis. On the other hand, the second part provides a concise literature review contrasting our research with existing methods to reduce the exhaustive human supervision required by these state-of-the-art models.

Chapter 3: Image Ranking by Predicting Permutations. This chapter, focusing on image ranking applications, describes a framework to learn permutations of images exploring the structure of permutation matrices. First, we review related works on attribute-based image ranking and its applications. Second, we review background topics important to the derivation of our model like principled algorithms to approximate doubly stochastic matrices. Third, we formulate our model and describe in detail the proposed learning and inference algorithms. Last, we evaluate our model on image ranking applications. **Relevant Publication:** [Gould et al., 2016; Santa Cruz et al., 2017; Santa Cruz et al., 2018b].

Chapter 4: Learning Image Representations by Permuting Image Regions. This chapter describes how to use the spatial structure in images to learn image representations in a self-supervised way. We start by describing how image jigsaws and their solutions can be represented by permutations of image regions and permutation matrices, respectively. Then, we learn to solve these jigsaws using the visual permutation learning framework presented in Chapter 3. Finally, we demonstrate that this approach is a good pretext task to learn useful image representations for objects which we evaluate on object recognition tasks such as object classification, detection and segmentation. We also discuss related works on visual representation learning and how to properly set up the image jigsaws in order to avoid the learning of uninformative representations. **Relevant Publication:** [Gould et al., 2016; Santa Cruz et al., 2017; Santa Cruz et al., 2018b].

Chapter 5: Compositional Algebra of Classifiers. This chapter describes how to explore regularities in classifier space in order to synthesize classifiers for new visual

concepts. In order to accomplish such a goal, we start by reviewing vision systems inspired by the principle of compositionality which also inspires the proposed framework. Next, we propose to represent complex visual concepts as boolean expressions of simple visual concepts. Then, we formulate our problem as an algebra of classifiers which is learned from data using compositional neural network modules. We also demonstrate that such a model can be simplified by the well known De Morgan's laws. Finally, we evaluate the proposed approach by synthesizing classifiers for boolean expressions of attributes and categories for birds and other animals. **Relevant Publication:** [Santa Cruz et al., 2018a].

Chapter 6: Activity Recognition as Inferring Action Patterns. This chapter describes how to leverage existing action classifiers to infer complex activities in videos. In the same spirit of Chapter 5, we first propose to describe complex activities as regular expressions of simple actions. Then, we develop a probabilistic model that can recognize instances of these expressions in videos. Last, we demonstrate the effectiveness of our approach on activity classification in trimmed and untrimmed videos. This chapter also reviews existing works on action recognition that try to circumvent the need of human supervision by leveraging textual data, highlighting their limitations on precisely describing complex activities. **Relevant Publication:** [Santa Cruz et al., 2019].

Chapter 7: Conclusion and Future Directions. We conclude the thesis with a summary of our main contributions and discussion of future directions for improving our work.

1.3 Publications

Much of the work described in this thesis has been previously published in conference proceedings, journals and technical reports as listed below.

- Gould, S.; Fernando, B.; Cherian, A.; Anderson, P.; Santa Cruz, R.; and Guo, E., 2016. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447, (2016).*

- Santa Cruz, R.; Fernando, B.; Cherian, A.; and Gould, S., 2017. Deeppermnet: Visual permutation learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

- Santa Cruz, R.; Fernando, B.; Cherian, A.; and Gould, S., 2018. Neural algebra of classifiers. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV).*

- Santa Cruz, R.; Fernando, B.; Cherian, A.; and Gould, S., 2018. Visual permutation learning. In *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI).*

- Santa Cruz, R.; Campbell, D.; Fernando, B.; Cherian, A.; and Gould, S., 2019. Inferring rich compositional activities in videos. Under review.

Chapter 2

Background

"Artificial intelligence is the science of making machines do things that would require intelligence if done by men."

Marvin Minsky, 1963

The purpose of this chapter is to provide an introduction to visual recognition from the computer vision and machine learning perspective. We first provide an overview of large scale visual recognition in Section 2.1. We start by defining visual recognition as the problem of interpreting the visual world, evolve to an explanation of the current data-driven approach for visual recognition and finish with a brief presentation of the state-of-the-art models for different visual recognition problems. Once the visual recognition problem and its current solutions are presented, we shift our focus to learn visual recognition models using minimal human supervision in Section 2.2. We provide a concise literature review on different strategies and methods to achieve such a goal. It is also important to emphasize that the current chapter is not intended to be a comprehensive treatment of either computer vision or machine learning. For an in depth coverage, the reader should consult the excellent textbooks on machine learning (e.g., [Bishop, 2006; Friedman et al., 2001; Murphy, 2012]) or computer vision (e.g., [Szeliski, 2010; Prince, 2012]).

2.1 Large-Scale Visual Recognition

Without bells and whistles, visual recognition consists of extracting semantic meaningful interpretations from visual data like humans do. Taking Figure 2.1 as an example, humans can recognize the nature scene from the blurry green background, localize the birds, note the water splash at the bottom of the picture, and even infer that the image is depicting birds drinking water from some water source like a river or a lake. This impressive skill is the result of many years of biological evolution of one of our most important sensing devices, the eyes, and interpreting device, the brain. Likewise, visual recognition focus on developing computational tools to mimic such an impressive skill using a camera as sensing device and computers as interpreting device.

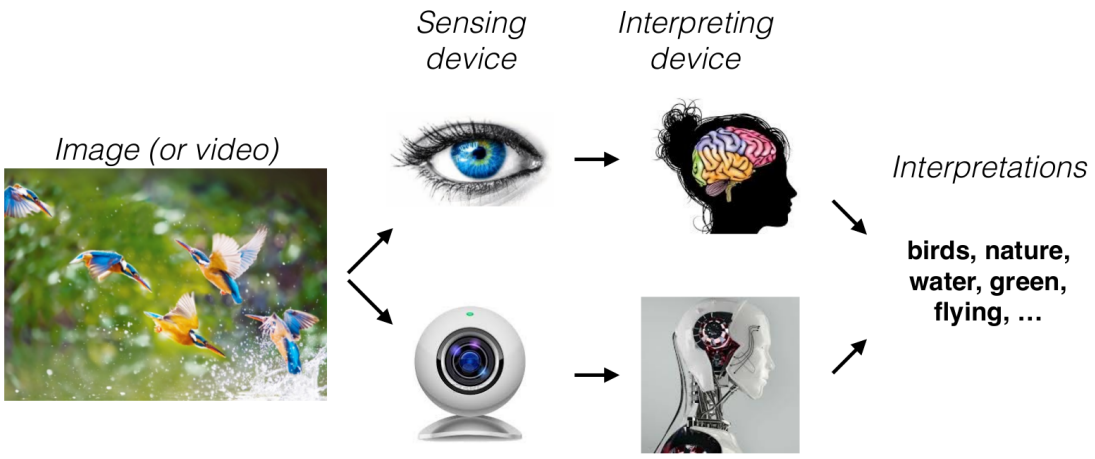


Figure 2.1: Visual recognition consists of inferring semantic entities in the visual world using computational tools. Image courtesy of Ballan [2018].

We can summarize this problem as modelling a function $f(\cdot)$ that maps from the visual data x to the semantic outputs y,

$$f : x \rightarrow y. \tag{2.1}$$

While x can be visual inputs like an image or a video, y can vary from simple image-level visual concept like "nature scene", passing through precise localization of semantic entities like the birds positions, to more abstract scene interpretations like the description, "the birds are drinking water from the river". Therefore, the exact form and representation of x and y depends on the application. For instance, image classification models infer discrete image labels, object detectors predict continuous bounding-box locations, and video captioning algorithms produce textual descriptions from videos.

While the task of inferring semantic outputs y from visual data x is trivial for humans, it is very hard for computer vision algorithms. It is not obvious how one might write an algorithm for identifying birds in images due to all possible appearance variations that they may present in the visual world. Figure 2.2 shows examples of common appearance variations that you may encounter in the visual world. Therefore, instead of try to model the function $f(\cdot)$ by a conventional algorithm, as we would do to sort a list of numbers, we collect many examples of such a visual concept and then develop learning algorithms to learn such a concept and recognize this concept in new visual instances. This approach is named as data-driven approach and consists of the most common approach for visual recognition.

For most of the visual recognition problems, the learning algorithm to be developed follows the supervised learning paradigm. More specifically, we collect a dataset of visual data and its respective outputs $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, parametrize the function $f(\cdot)$ in terms of learnable parameters θ (denoted from now on as $f_\theta(\cdot)$), and define a loss function $\Delta(\cdot, \cdot)$ that measures how wrong is our prediction $f_\theta(x_i)$ from

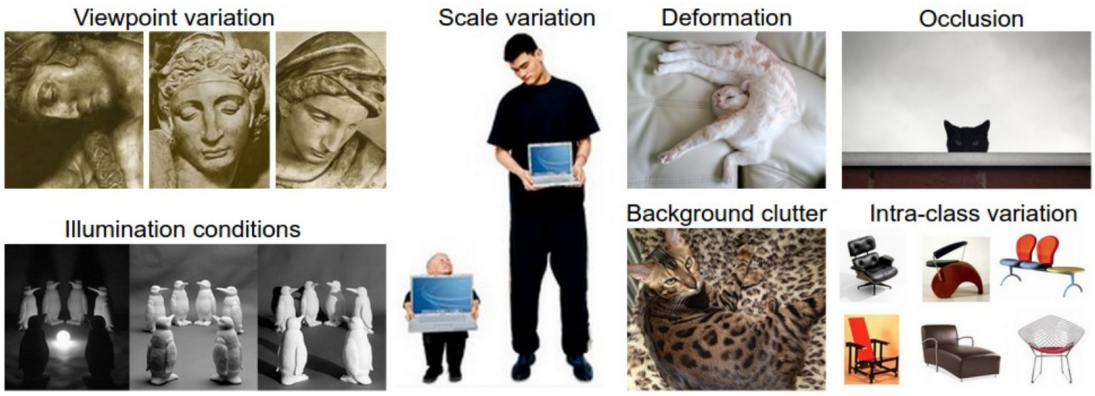


Figure 2.2: Variations in the visual appearance of semantic concepts. Image courtesy of Li et al. [2019].

the expected output y_i given the visual data x_i. Then, the parameters θ are estimated by minimizing the loss function $\Delta(\cdot,\cdot)$ over the training set $\mathcal{D}$. Mathematically, our learning algorithm consists of the following optimization problem,

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} \Delta\left(f_\theta(x_i), y_i\right) + R(\theta), \tag{2.2}$$

where the form of the loss function $\Delta(\cdot,\cdot)$ depends on the type of the output y and the application of interest. For instance, regressing bounding boxes coordinates requires continuous outputs which are a good fit for the euclidean loss $\sum_{i=1}^{N} \|y_i - f_\theta(x_i)\|_2^2$, while distinguishing between cats, dogs, and people images requires discrete outputs which can be easily handled by the log cross-entropy loss $\sum_{i=1}^{N} -logP(y_i|x_i;\ \theta)$ where the probability $P(y_i|x_i;\theta)$ is computed by the softmax function over the outputs of our learnable function $f_\theta(\cdot)$. Furthermore, the above minimization problem is often solved by gradient based methods [Boyd and Vandenberghe, 2004] where the gradients are computed using the back-propagation algorithm [Rumelhart et al., 1988] which requires the loss function and our model to be differentiable with respect to the learnable parameters θ. The function $R(\cdot)$ is some regularization function to avoid over-fitting providing also good predictions for visual data instances that are not in the training set.

We now have all the ingredients to solve a generic visual recognition problem, with the exception of the accurate formulation of the function $f_\theta(\cdot)$ also called model, in a machine learning perspective. Initially, computer vision researchers proposed to decompose this function in two other functions: a feature extractor and a machine learning algorithm. The former involves defining what visual features are relevant to a given task and designing data processing pipelines to extract and encode those characteristics in an format amenable to the latter learn from and produce the desired semantics outputs. Following these ideas, features extractors like SIFT [Lowe, 2004] and HOG [Dalal and Triggs, 2005] have been employed with machine learning models like support vector machines [Cortes and Vapnik, 1995] and boosted classifier [Freund et al., 1999] in a wide range of visual recognition tasks [Dollár et al., 2014;

Viola and Jones, 2001]. However, this strategy requires huge efforts in engineering and domain knowledge imposing difficulties for many large scale visual recognition applications.

Recently, aided by the developments in hardware platforms [Nickolls et al., 2008] and the availability of large scale human annotated datasets [Deng et al., 2009; Lin et al., 2014; Caba et al., 2015], enormous progress has been made by deep learning models which has drastically boosted the state-of-the-art performance in many visual recognition tasks. Basically, these models consists of a cascade of multiple layers of nonlinear processing units that jointly performs feature extraction and model learning from pixels to semantic outputs which is also known as end-to-end learning. Since deep learning models can vary a lot depending on the target application, we will focus on the explanation of one of the simplest deep learning model for now, i.e., the multi-layer perceptron neural network (MLP) [Rosenblatt, 1961], and later present the state-of-the-art models in the applications that are relevant for this thesis.

As shown in Figure 2.3, a standard 2-layers MLP consists of the computation of two linear transformations followed by non-linear functions that maps the input visual data to the semantic outputs. Then our model $f_\theta(\cdot)$ can be described as,

$$f_\theta(x) = h_2\left(W_2^\intercal \;\; h_1\left(W_1^\intercal x + b_1\right) + b_2\right) \tag{2.3}$$

where W_1, W_2, b_1, and b_2 together form the set of learnable parameters θ which should be estimated as described in Equation 2.2. Likewise, h_1 and h_2 are called activation functions. h_1 is usually implemented by a sigmoid function, while h_2 will depend again on the desired semantic outputs y and the target application. For instance, regressing continuous bounding-boxes coordinates can be accomplished by a linear transformation, while predicting discrete labels can be achieved by a softmax function. According to the universal approximator theorem, these models can approximate any continuous function on a compact input domain to arbitrary accuracy provided the network has a sufficiently large number of learnable parameters, sufficient amount of training data and appropriated learning algorithm [Hornik, 1991; Cybenko, 1989]. However, the assumptions of this theorem are often violated in practice.

In summary, current visual recognition models following the deep learning approach interpret the visual world by learning a sequence of over-parametrized non-linear transformations that maps from visual inputs to semantic outputs. Such an approach is very convenient compared to the two steps formulation used before by computer vision practitioners since it requires less domain knowledge. However, this data-driven approach relays heavily on the abundance of human annotated data which is problematic and expensive as discussed in Chapter 1. In the next sections, we discuss details of these supervised deep learning models for the visual recognition applications relevant for this thesis, while we delay the presentation of existing strategies to overcome these problems to Section 2.2.

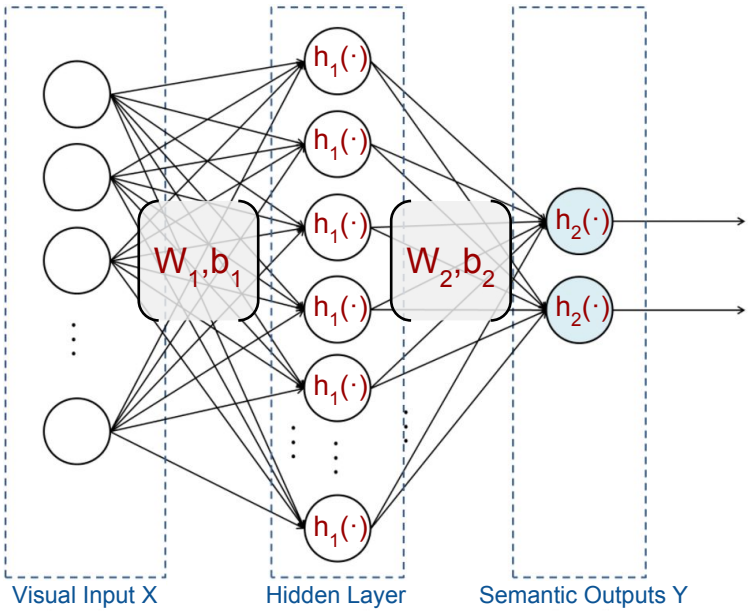


Figure 2.3: The Multi-layer perceptron neural network (MLP) consists of the computation of two linear transformations followed by non-linear functions ($h_1(\cdot)$ and $h_2(\cdot)$) that maps the input visual data to the semantic outputs. In this model, the learnable parameters θ are W_1, W_2, b_1, and b_2.

2.1.1 Object Recognition

We start our discussion about current deep learning models for visual recognition applications by considering object recognition which is one of the most famous computer vision problems. Typically, it encompasses three different tasks: Object Classification, object detection and object segmentation. The first consists of assigning one or more labels to a given image, the second aims to localize objects using bounding boxes in addition to classify them, and the third refers to the task of predicting a label for every pixel of the image, producing the localization as well as marking the extent of every object in the image. Figure 2.4 illustrates these tasks.

In the case of object classification, the most used deep learning models are the Convolutional Neural Networks (CNNs). Like MLPs, CNNs are feed-forward models consisting of a sequence of non-linear transformations also called layers. The main type of these layers is the convolutional layer which applies a linear transformation (convolution) to its input followed by a non-linear function (e.g., REctified Linear Unit (ReLU) or sigmoid). The network may also contain other types of layers, such as pooling layers that are used to downsample the input, dropout layers that try to prevent overfitting by randomly dropping intermediate outputs, or fully-connected layers which are essentially the MLP's layers. The models are usually trained by stochastic gradient descent and back-propagation as described in previous section.

The exact sequence of layers and their parameters defines the CNN's architecture. The first architecture to earn notoriety was the AlexNet proposed by Krizhevsky et al.

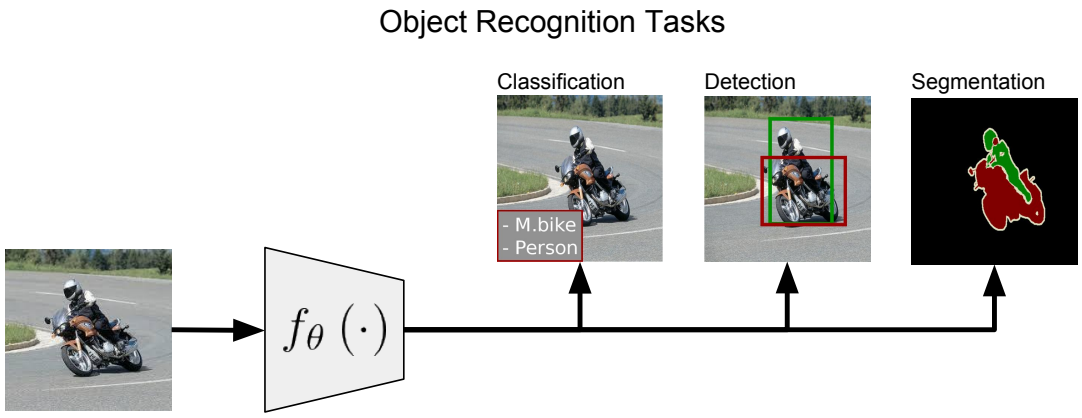
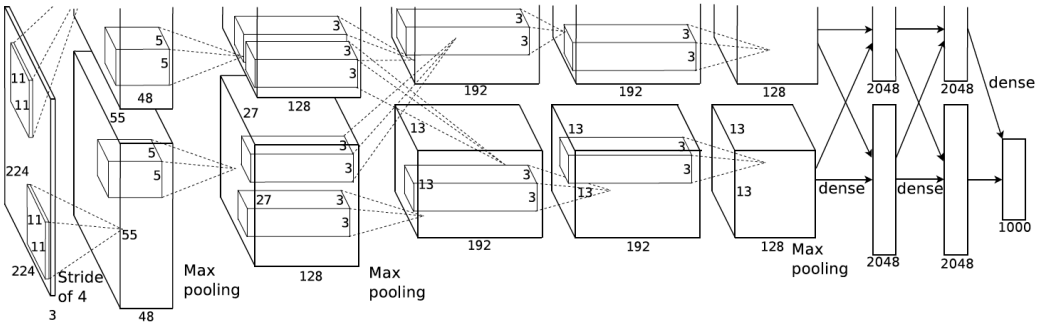


Figure 2.4: Illustration of the traditional object recognition tasks: Object Classification, object detection and object segmentation. Object classification consists of assigning one or more labels to a given image, object detection aims to localize objects using bounding boxes in addition to classify them, and object segmentation refers to the task of predicting a label for every pixel of the image.

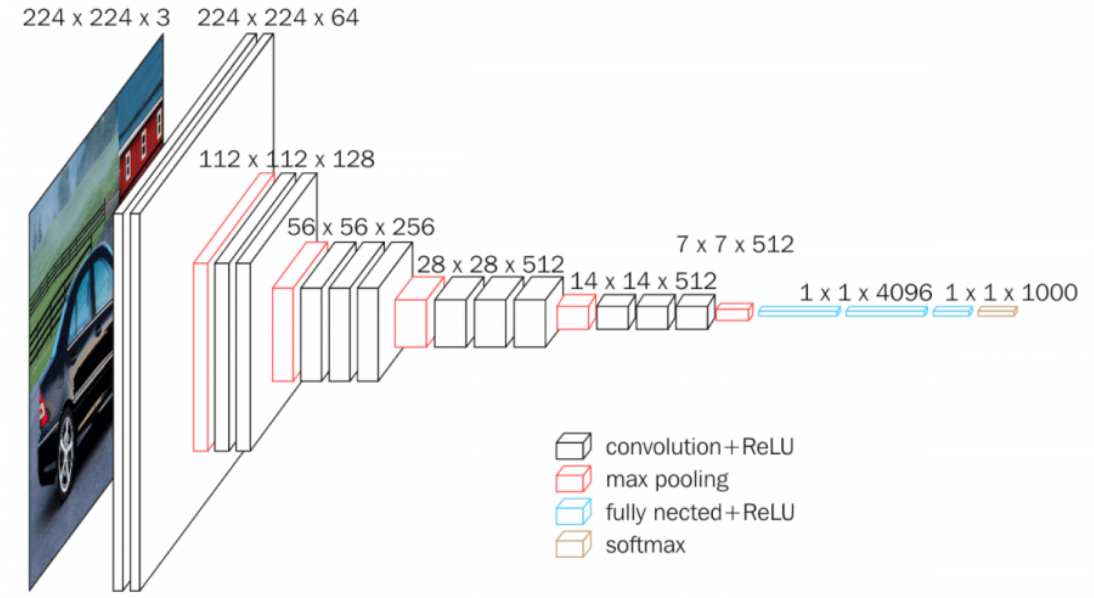
[2012]. The authors won the ImageNet Challenge in 2012 by a large margin starting the deep learning era in visual recognition as discussed in Chapter 1. As illustrated in Figure 2.5(a), the AlexNet is composed by five convolutional layers followed by three fully-connected layers. It uses Rectified Linear Units (ReLU) as non-linearity instead the standard sigmoid function. To prevent overfitting, it uses two dropout layers and data augmentation techniques – during training, the data set is augmented with random translations, reflections, and patch extractions from the training images.

An important aspect of CNN's architectures is the number of consecutive layers also called depth. There is the notion that deeper networks can take more advantage of the hierarchical representation of the visual data allowing to learn more effective models for visual recognition. Following these ideas, deeper and even deeper networks have been proposed like VGG [Simonyan and Zisserman, 2014b] (Figure 2.5(b)) and GoogleNet [Szegedy et al., 2015] which has 16 and 22 layers, respectively. The GoogleNet also introduced the inception module (Figure 2.5(c)), a set of small convolutions to drastically reduce the number of learnable parameters attempting to avoid overfitting.

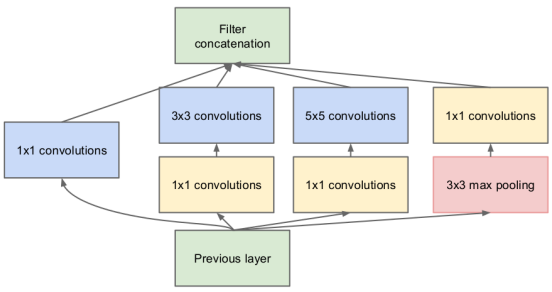
While these deep networks can learn a more rich feature hierarchy and complex functions, they are harder to train due to the well known vanishing gradient problem [Hochreiter et al., 2001]. During training, the gradients in the first layers of the network approach zero as the network architecture gets deeper which slows down the learning of the earlier layers. In order to circumvent such a problem, skip connections (Figure 2.5(d)) have been used by CNNs architectures such as Highway Network [Srivastava et al., 2015b], ResNet [He et al., 2016], and DenseNet [Huang et al., 2017]. These CNN architectures can have 1,000 layers and produce impressive results. For instance, the ResNet outperformed human experts in the ImageNet



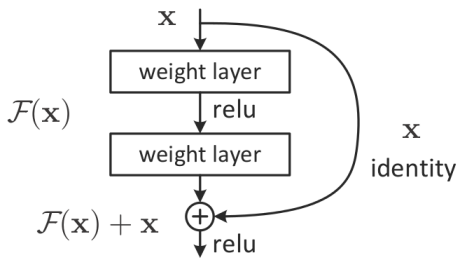
(a) AlexNet Architecture.



(b) VGG Architecture.



(c) Inception module.



(d) Skip Connection.

Figure 2.5: Most common CNN architectures used in visual recognition applications. Figures 2.5(a), 2.5(b), 2.5(c), and 2.5(d) are courtesy of Krizhevsky et al. [2012], Simonyan and Zisserman [2014b], Szegedy et al. [2015], and He et al. [2016], respectively.

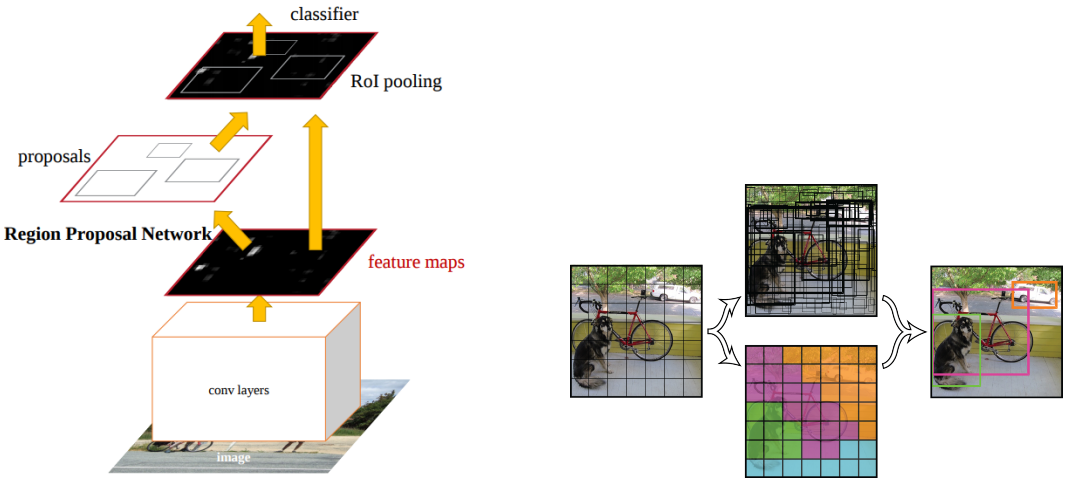
Challenge 2015. Another CNN component that is worth to mention is batch normalization [Ioffe and Szegedy, 2015] which normalizes the inputs of intermediate layers by adjusting and scaling the activations using batch statistics. This component speeds up the network training, in addition to slightly reduce the overfitting.

The methods developed in this thesis are based on CNN models following the aforementioned architectures. For instance, the visual permutation learning model described in Chapter 3 uses a Siamese architecture [Bromley et al., 1994; Chopra et al., 2005] whose each subnetwork follows the AlexNet, while the neural algebra of classifiers described in Chapter 5 uses a feature extractor network based on the VGG variant.

While object classification only focus on classifying images according to different object classes, object detection aims to localize in addition to classify each instance of object in a image using a tight rectangle called bounding box. Due to the similarity between these two problems, object detection models are usually implemented as the classification of multiple locations and scales of the input image [Felzenszwalb et al., 2010; Dollár et al., 2014]. Following these ideas, the Region Convolutional Neural Network (R-CNN) detector [Girshick et al., 2014] combines the selective search region proposal method [Uijlings et al., 2013], AlexNet feature extraction and linear SVM classifiers. It also uses a per-class bounding box regression mechanism which refines the detections to tightly enclose the object instances. In order to train such a model, the authors first pretrain the CNN feature extractor in the ImageNet dataset for object classification, then they fine-tune the CNN and train the SVM classifiers in a detection dataset. The R-CNN detector earned notoriety by improving the state-of-the-art performance significantly on many modern object detection benchmarks.

Despite the impressive results, the R-CNN detector presents many inefficiencies at training and test time. It requires to run the CNN's forward-pass on about 2000 regions proposals per image which is very computationally expensive. In addition, the multi-stage training is not convenient. Inspired by the SPPNet [He et al., 2015], Girshick [2015] proposes the Fast R-CNN detector introducing the Region of Interest pooling layer (RoIPool) which extracts a fixed-length feature vector for each proposal directly from the corresponding region of the convolutional map reducing the number of CNN's forward-pass per image for only one. It also simplifies the training process by replacing the SVM classifiers by fully connected layers forming a two head structure with the bounding box regressor. Such a structure allows the model to be trained for classification and localization jointly using a Multi-Task loss. The transfer learning experiments described in the Section 4.3 of Chapter 4 uses the Fast R-CNN model to perform object detection.

Inspired by the success of end-to-end learning approaches in related problems, Ren et al. [2015] propose the Faster R-CNN detector by introducing the Region Proposal Network (RPN) into the Fast R-CNN model as shown in Figure 2.6(a). More specifically, the object proposal method is replaced by the Region Proposal Network which is another CNN that slides over the last feature map to determine whether a region is an object or not. It predicts "objectness" scores, bounding boxes coordinates, and bounding boxes dimensions for K anchor boxes which are then used to



(a) Faster R-CNN - Two-stage approach (b) Yolo - One-stage approach

Figure 2.6: Fast R-CNN [Girshick, 2015] and YOLO [Redmon et al., 2016] deep learning frameworks for object detection. Figures 2.6(a) and 2.6(b) are courtesy of Girshick [2015] and Redmon et al. [2016], respectively.

train the rest of the model which is essentially its predecessor the Fast R-CNN model. The Faster R-CNN framework is an end-to-end deep learning framework for object detection and presents slightly better performance than the Fast R-CNN.

The aforementioned deep learning frameworks for object detection follow a two-stage strategy where regions of interest are first extracted and then classified. Despite the accuracy of these methods, they are extremely slow for real-time applications processing only about 6 frames per second (FPS) on a GPU. In order to speed up deep learning-based object detectors the one-stage strategy was developed concurrently by Liu et al. [2016] and Redmon et al. [2016]. These algorithms predict bounding boxes and class probabilities all at once from a given input image. More specifically, the Single-Shot Detector (SSD) [Liu et al., 2016] predicts a large number of bounding boxes and class probabilities, like the anchor boxes in the Faster R-CNN, using feature maps from different convolutional layers of the backbone CNN. Then, the Non-Maximum Suppression method is used to keep the most relevant detections. Differently, as shown in Figure 2.6(b), the You Only Look Once (YOLO) [Redmon et al., 2016] divides the input image into a regular grid, predicts different bounding boxes and confidence scores within each grid cell, selects the most relevant bounding boxes using their intersection over union (IoU) and predicted confidence scores, and assigns class probabilities for these most relevant detections. Just to have an idea of the level of speed up brought by these models, YOLO runs at 45 FPS on a GPU and their light version which uses a smaller CNN architecture can run at impressive 155 FPS on a GPU. Other versions of this model was developed later to deal with larger number of object classes [Redmon and Farhadi, 2017] and to make the predictions more accurate [Redmon and Farhadi, 2018], but always keeping the real-time performance.

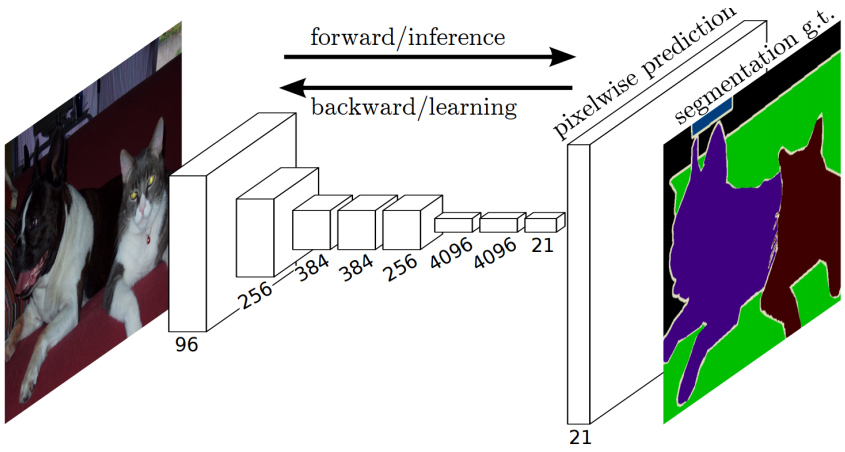


Figure 2.7: Architecture of Fully Convolutional Networks (FCN) for image segmentation. Image courtesy of Long et al. [2015].

As mentioned earlier, object segmentation aims to interpret images at pixel level, i.e., the objective is to assign a class label for every pixel of the input image. Like in detection problems, the initial deep learning approaches for segmentation were just adaptations of classification models [Ciresan et al., 2012]. More specifically, they assign labels to pixels by classifying the image patch around it. The main reason to adopt this approach was because CNN's with fully connected layers can only handle fixed size inputs and outputs. In order to circumvent such a limitation and provide an efficient model, Long et al. [2015] propose the Fully Convolutional Network (FCN), a deep learning model for dense prediction tasks like object segmentation. This model does not have fully-connected layers and it is able to process images of any size in a single shot producing segmentation maps in a more efficient way than the patch classification approach. In addition, the authors introduced deconvolution layers to upsampling feature maps and skip connections to refine the predicted segmentation maps. The architecture details of the FCN is shown in Figure 2.7 and this model is used in the segmentation experiments described in the Section 4.3 of Chapter 4.

Focusing on biomedical image segmentation, Ronneberger et al. [2015] extends the FCN architecture aiming to reduce the amount of training images and to improve the fine details in the predicted segmentation maps. The authors propose the U-Net, a encoder-decoder architecture for image segmentation. The encoder part, named contracting network, reduces the dimensions of the input image by extracting many feature maps of decreasing resolution using convolutional layers. On the other hand, the decoder part, named expanding network, consumes these low resolution feature maps producing fewer high resolution feature maps using deconvolution layers. Finally, a 1x1 convolutional layer processes these high resolution feature maps to predict the final segmentation result. The U-Net also employ skip connections to allow the decoder leverage the high resolution feature maps from earlier layers of the encoder network producing more precise segmentation results.

The U-Net architecture is well accepted by the community and has motivated many other works recently. As examples, Lin et al. [2017] propose the the Feature Pyramid Network (FPN) which uses similar architecture than U-Net, but predicts segmentation maps at different layers of the decoder network denoted by the authors as the top-down pathway. Zhao et al. [2017] develop the Pyramid Scene Parsing Network (PSPNet) which aims to better learn the global context of a scene by introducing dilated convolutions [Yu and Koltun, 2015] and the Pyramid Pooling Module in the encoder part. While the dilated convolutions allow to increase the size of the receptive field without decreasing the spatial dimensions of the output feature maps, the Pyramid Pooling Module allows to analyse the image at different scales by processing feature maps pooled at different scales. Pursuing similar goals, different versions of the DeepLab model [Chen et al., 2018, 2017] make use of the fully connected pairwise CRF by Krähenbühl and Koltun [2011] as a separated post-processing step to capture long term dependencies between pixels in order to produce refined segmentation results. These models also use a multi-scale approach similar to the PSPNet's Pyramid Pooling Module but exploring dilated convolutions.

Up to the present moment, we have discussed object classification, detection and segmentation in isolation. However, it is evident that these tasks have a lot in common. For instance, a good segmentation algorithm has to be able to localize and classify every object in a image in order to produce a good segmentation map. Following these ideas, He et al. [2017] propose the Mask R-CNN, a deep learning model for instance segmentation which consists of predicting segmentation masks for every instance of objects depicted in a image. Note that such a problem differs from traditional object segmentation because it differentiate object instances. In order to tackle the instance segmentation problem, the Mask R-CNN extends the Faster R-CNN object detector by adding a branch for predicting an object mask in parallel with the existing branches for predicting bounding box coordinates and object class probabilities. These branches and the whole model are trained using a multi-task loss which tries to solve these complementary tasks jointly leading to better models on each individual task.

2.1.2 Action Recognition

Like object recognition, action recognition is also a fundamental task in computer vision [Kang and Wildes, 2016; Herath et al., 2017]. It refers to the act of classifying or localizing the execution of complete actions in videos. However, different from object recognition, such a problem requires thorough analysis of the temporal evolution of semantic entities in addition to the understanding of the appearance of static images in isolation. Consider the two videos (visualized as sequences of frames) in Figure 2.8 as an example. While we can only predict that someone is swimming in both videos by analysing the frames in isolation due to the recognition of semantic entities like water and human body, we can distinguish front crawl from breaststroke swimming styles by analysing the temporal evolution of the frames and noting the differences between the periodic motion patterns of the arms. Therefore, action recognition

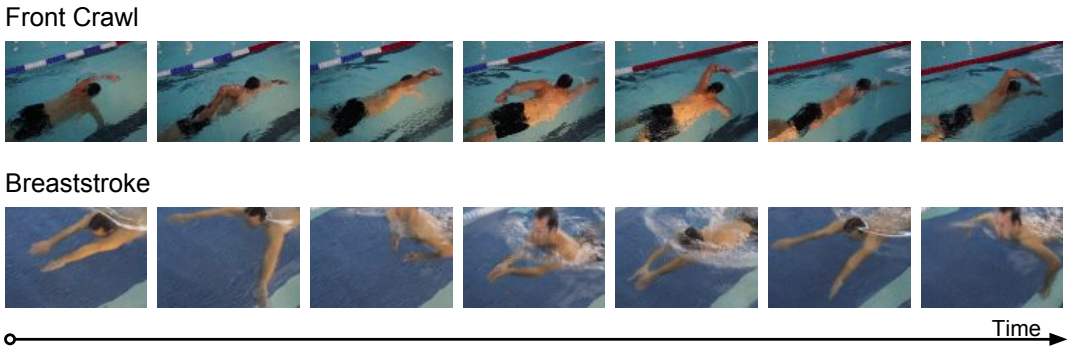


Figure 2.8: The influence of temporal information on action recognition tasks. While we can only predict that someone is swimming in both videos by analysing the frames in isolation due to the recognition of semantic entities like water and human body, we can distinguish front crawl from breaststroke swimming styles by analysing the temporal evolution of the frames and noting the differences between the periodic motion patterns of the arms. These images are courtesy of Ghosh [2018].

systems should reason about temporal information depicted in videos in order to accurate predict actions.

The most straightforward way to incorporate temporal information into deep learning models is to extend Convolutional Networks to the temporal domain by using 3D convolutions as basic building blocks instead of 2D convolutions. In this way, the 3D convolutions can extract both spatial and temporal features from adjacent frames. Figures 2.9(a) and 2.9(b) illustrate the difference between these operations. This approach also called space-time networks was first used for action recognition by Ji et al. [2010] and latter improved by Tran et al. [2015] using modern CNN architectures as backbone and large scale human annotated datasets for training. The main drawback of 3D CNNs is the large number of learnable parameters provoked by the extensive use of 3D convolutions making these models susceptible to overfitting and hard to train. In order to circumvent such a problem, Carreira and Zisserman [2017] propose to inflate very deep 2D CNNs for image classification into 3D CNNs by repeating 2D convolutional filters along the time dimension, allowing the network to reuse 2D filters pretrained on larger and richer static images datasets like ImageNet. Their model is known as I3D and we use it as primitive action classifiers in the action recognition experiments in Section 6.3.2 of Chapter 6. Other more elaborated approaches to transfer the knowledge between 2D CNNs and 3D CNNs have been propposed in [Qiu et al., 2017; Varol et al., 2018].

According to the two-stream hypothesis in visual perception [Goodale and Milner, 1992], object attributes such as appearance, color and identity are handled separately from its motion and location information by two different streams, the *Ventral Stream* and the *Dosaral Stream*, respectively. Inspired by these ideas, Simonyan and Zisserman [2014a] introduce another way to model visual appearance and temporal information in action recognition using convolutional neural networks named multiple stream networks. Their so called Two-Stream model, shown in Figure 2.9(c),

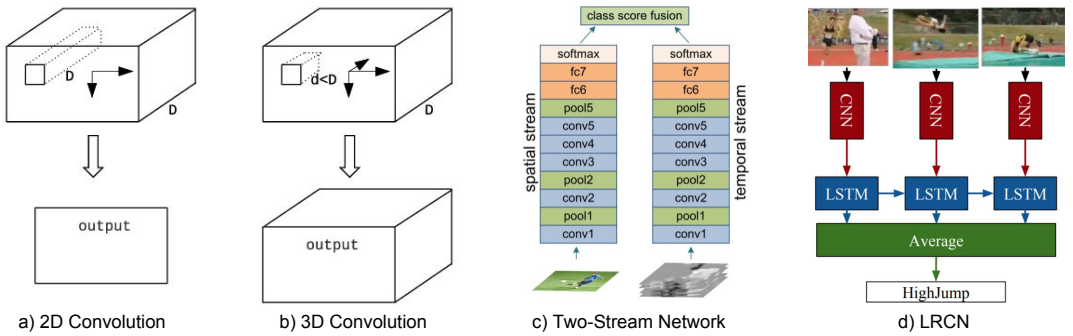


Figure 2.9: Deep learning approaches for action recognition. Figures 2.9(a) and 2.9(b) compares 2D and 3D convolution operation. Figure 2.9(c) shows the two-stream architecture proposed by Simonyan and Zisserman [2014a], and Figure 2.9(d) shows the temporal pooling approach using LSTM proposed by Donahue et al. [2015]. Figures 2.9(a) and 2.9(b) are courtesy of Tran et al. [2015], while Figures 2.9(c) and 2.9(d) are courtesy of Yue-Hei Ng et al. [2015] and Donahue et al. [2015], respectively.

consists of a two parallel CNNs for processing raw video frames and optical flow fields separately. These two streams are then fused together by averaging their softmax scores. Since such a fusion schema is not appropriate for learning long term temporal information, Feichtenhofer et al. [2016] propose to extend the two-stream model allowing fusion at an intermediate layer. Aiming at the same goal, Wang et al. [2015] propose to aggregate dense trajectories [Wang and Schmid, 2013] traced over convolutional feature maps of the two-stream, Yue-Hei Ng et al. [2015] investigate temporal feature pooling, and Girdhar et al. [2017] propose the ActionVLAD pooling layer that aggregates convolutional feature descriptors in different image portions and temporal spans. There are also works like [Wu et al., 2015a; Zolfaghari et al., 2017] which investigate other streams of data processing like audio signals and human body pose information, respectively.

Another approach is to use temporal pooling or aggregation to capture temporal information in a video. Donahue et al. [2015] extract visual features for every frame in a video using a 2D CNN and capture the temporal evolution of these features using a LSTM as shown in Figure 2.9(d). Yue-Hei Ng et al. [2015] investigate this approach in depth comparing different ways to perform temporal aggregation using LSTMs on top of 2D CNN features. Wu et al. [2015b] extend this CNN-LSTM schema by using a two-stream network to extract visual appearance and motion features and bidirectional LSTM to model long term temporal dependencies. Using other formulations different from LSTMs to capture and represent the temporal information in video, Fernando et al. [2015b] and its variants [Fernando and Gould, 2016; Fernando et al., 2016] use a learning-to-rank approach and Cherian et al. [2017] propose to represent sequences of frames as a subspace. In order to constraint the video representation to capture useful information, in addition to the temporal information, Wang et al. [2018] propose to use the decision boundaries of a SVM classifier that separates data features from independently sampled noise, and Wang and Cherian

[2018] extend such an approach using adversarial perturbations to model the data dependent noise generation.

The aforementioned models are designed solely for predicting action labels in videos. However, temporal action localization which consists of predicting the start and end frame of complete action instances have also been studied by the computer vision community. Like object detection models, early approaches address this task by applying temporal sliding window followed by classifiers to classify the action within each window [Ni et al., 2016; Yuan et al., 2016], then these solutions evolved to region proposal approaches where a hand-crafted algorithm is used to generate generic action proposal that are subsequently classified into different action categories [Caba Heilbron et al., 2016; Escorcia et al., 2016], and recently deep learning end-to-end approaches have been developed for this problem. In this direction, we would like to highlight the works of Xu et al. [2017b] which adapted the Faster R-CNN object detector for temporal action localization, Dai et al. [2017] introduce temporal context in the prediction of action proposals, and Chao et al. [2018] propose modifications on these previous approach to handle the large variation in action instances duration. Moving forward, there also exists a large body of work on spatio-temporal action localization which focus on localizing spatially and temporally complete action instances in videos, in addition to classify them [Gkioxari and Malik, 2015; Kalogeiton et al., 2017]. Since we only make use of action classifiers in the development of the inference algorithm for compositional activity recognition in Section 6.2 of Chapter 6, a detailed presentation of these localization methods in beyond the scope of this thesis.

2.1.3 Image ranking

In order to finish our brief presentation of current deep learning models for different visual recognition applications, we now discuss image ranking. The goal of image ranking is to order a collection of images according to some predefined criterion which can range from visual attributes to natural language queries. Figure 2.10 shows examples of this task. This topic has been explored by the scientific community with applications in information retrieval [Yang and Hanjalic, 2010], active learning [Liang and Grauman, 2014], zero-shot learning [Parikh and Grauman, 2011] and person re-identification [Wang et al., 2016]. In order to solve such a task, supervised learning-to-rank algorithms are usually employed. As all machine learning methods that follow the supervised paradigm, supervised learning-to-rank methods learn to order new image instances from a training set of correctly ordered sequences of images according to some criterion.

Supervised learning-to-rank algorithms can be categorized by the way they process the training and testing sequences. Point-wise methods process each element of the sequences individually. They first use a classifier or regressor based algorithm to estimate how relevant a single element is for a given criterion, then the final ranking is obtained by sorting all elements by these scores. Following this approach, Crammer and Singer [2002] propose a rule based algorithm, Shashua and Levin [2003] use

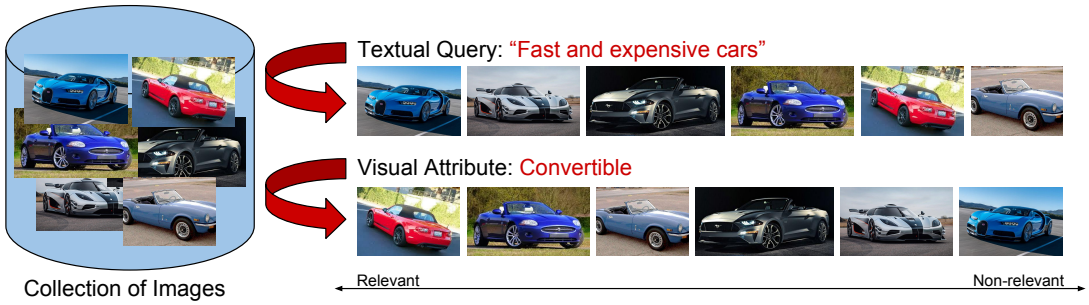


Figure 2.10: Example of image ranking problem. The goal of image ranking is to order a collection of images according to some predefined criterion which can range from visual attributes (e.g., Convertible) to natural language queries (e.g., "Fast and expensive cars").

multiple parallel hyperplanes, Cossock and Zhang [2006] explore regression errors, and Li et al. [2008] use a gradient boosting tree algorithm to produce these relevance scores. Point-wise methods are simple, easy to train, but prone to over-fitting.

Pair-wise methods process pairs of elements in the sequences at a time during training and testing. They essentially formulate the ranking task as classification of pairs into correctly and incorrectly ordered pairs. The final ranking is obtained by performing pair-wise comparisons between the elements in a given input sequence. As good examples of these methods, Herbrich et al. [2000] propose the large margin formulation for ranking problems known as RankSVM, Burges et al. [2005] develop a neural network model and training schema named RankNet, Souri et al. [2016] propose a Siamese CNN architecture and ranking loss that takes ties between elements into account, Singh and Lee [2016] design a CNN based model able to localize, in addition to compare images according to visual attributes, and Li et al. [2018] provide a more interpretable ranking model using a probabilistic framework. Pair-wise methods work better in practice than point-wise methods, because predicting relative order is closer to the nature of ranking than predicting relevance scores. However, they are limited to only explore the information depicted in training pairs.

List-wise methods process entire sequences at once by optimizing thoroughly designed objective functions over them. They essentially optimize ranking quality metrics like normalized discounted cumulative gain (NDCG), mean average precision (MAP), and mean reciprocal rank (MRR) using some sort of surrogate objective or derivative-free optimization method, since these metrics are not differentiable. Taylor et al. [2008] propose a smooth approximation for NDCG, Xu and Li [2007] develop an adaptive boosting algorithm to optimize NDCG and MAP using an ensemble of "weak rankers", Mohapatra et al. [2018] propose a quick-sort flavored optimization algorithm, and Engilberge et al. [2019] propose to use an additional neural network to perform the rank step of the computation of these metrics making them differentiable. List-wise methods are more computationally expensive, yet very effective because they are able to explore all the structural information present in the training sequences.

There is still a fourth category of supervised learning-to-rank methods positioned between pair-wise and list-wise methods that explores subsequences in order to produce a global ranking function. For instance, Fernando et al. [2015a] propose the MidRank model which explores multiple pair-wise relations within subsequences at the same time optimizes a list-wise ranking loss. Likewise, the visual permutation learning model proposed in Chapter 3 belongs to this family of rankers, however, our method is CNN based and is able to learn image representations and ranking function jointly from the pixel data.

2.2 Visual Recognition With Minimal Supervision

As discussed in Chapter 1, learning with minimal human supervision is a fundamental step towards the development of computer vision algorithms as capable as the human visual system. Due to its importance, there exists a rich literature on methods and practices aiming to achieve this goal. We dedicate the rest of this chapter to review existing works in the literature that like us aim to learn visual recognition systems with minimal human supervision. We start by enumerating existing learning paradigms alternative to the dominating fully supervised approach which is described in Section 2.1. Then, we focus our discussion on the weakly supervised learning approach, since it guides most of the methods developed in this thesis. In order to provide a complete literature review, we finish this section by discussing other forms of human free supervision that have been extensively used recently.

2.2.1 Standard Non-Supervised Learning Paradigms

As discussed before, the supervised learning paradigm is the mainstream in visual recognition despite its dependency on the availability of human annotated datasets. However, in the literature, there are other learning paradigms that aim to overcome such a limitation. The current section reviews these learning paradigms, as well as their applications in visual recognition.

Unsupervised Learning. As widely stated in machine learning books, unsupervised learning techniques explore the underlying structure of unlabelled data to learn mappings between inputs and outputs of a system [Bishop, 2006]. Besides not requiring human annotated datasets which are very expensive to collect and maintain, these techniques have many other advantages over fully supervised approaches, e.g., they are less sensitive to data bias and inconsistencies originated by the labelling process. Due to these reasons, it has been widely used in computer vision applications involving data clustering, compression and representation. For instance, the classic Autoencoder framework [Baldi and Hornik, 1989; Hinton and Zemel, 1994] has been used for image representation [Vincent et al., 2008, 2010], incremental learning [Aljundi et al., 2017], 3D orientation learning [Sundermeyer et al., 2018] and image super resolution [Yu and Porikli, 2017].

In spite of being a very generic approach, unsupervised learning methods imposes several challenges for visual recognition applications caused by the following *"chicken-and-egg problem"*: How to search for a object before knowing what it looks like? How to represent an activity without knowing how it happens? How to estimate a pose if we do not know what are the articulations? As a consequence, these techniques are discouraged in many real world visual recognition problems and usually exchanged by semi-supervised or weakly supervised methods.

Semi-Supervised Learning. Semi-supervised learning techniques were initially proposed as a way to provide strong generalization for supervised models by utilizing abundant unlabelled data [Chapelle et al., 2006]. However, we can also say that these techniques attempt to overcome the aforementioned *"chicken-and-egg problem"* by introducing a small labelled dataset which provides guidance to the learning process without incurring in excessive data curation cost or other supervised learning inherent problems. For instance, Lee [2013] first learns classifiers for the concepts of interest guided by the labelled data and then explores regularities in the unlabelled data to refine such a model using pseudo-labels. Likewise, Haeusser et al. [2017] perform similar refinement by learning associations between labelled and unlabelled samples in the feature space, while Sajjadi et al. [2016] regularize the initial model by introducing an unsupervised regularization term to push the decision boundaries to less dense areas of decision space and to enforce mutual exclusivity of classes.

Along the years, semi-supervised learning methods have achieved good performance in different visual recognition tasks [Rasmus et al., 2015; Tarvainen and Valpola, 2017], but there were always concerns whether the initial supervised model could bias the exploration of the unsupervised data to a wrong direction in which mistakes are reinforced instead of fixed. Very recently, Oliver et al. [2018] endorse these concerns by showing that traditional techniques for semi-supervised learning can be outperformed by supervised baselines using well-tuned hyper-parameters or transfer learning techniques. The authors also pointed out that the performance of these methods can degrade substantially when the unlabelled dataset contains out-of-distribution examples.

Weakly Supervised Learning. On the other hand, weakly-supervised learning methods focus on applications that have been successfully tackled by supervised learning approaches, but they only explore a "weaker" form of supervision. Such a "weak supervision" should exist in abundance, be easily collected, or be computed on-the-fly in order to these approaches be useful. These methods allow us to learn object detection models without the laborious bounding-box annotations but using only image tags [Shi et al., 2017] and to derive object segmentation models using the same image-level labels instead of the even laborious pixel-wise annotations [Pathak et al., 2014]. In the video domain, these techniques are even more important since it is much convenient to learn temporal action segmentation models from sequences [Richard et al., 2018b] or sets [Richard et al., 2018a] of actions that can be extracted from video

transcripts than from dense frame-wise annotations used by the supervised state-of-the-art methods [Carreira and Zisserman, 2017]. Therefore, these techniques are very convenient as well as effective in practice.

In this dissertation, we also follow the weakly-supervised learning paradigm by proposing methods that explore weak forms of supervision obtained from the structure and priors existent in the visual world. As examples, we exploit the spatial structure and context depicted in images to learn image representations without annotated data in Chapter 4, while we perform zero-shot image classification by using only visual primitive labels in Chapter 5. However, it is also common to perceive a mix of these approaches in our applications as well as in other works in the literature. For instance, we use additional supervised methods and human annotated datasets in the transfer learning experiments in Chapter 4 resulting in a semi-supervised approach. In the literature, Hu et al. [2018] propose an object instance detection and segmentation model able to segment 3000 visual concepts by training the Mask R-CNN framework using a mix of human annotated bounding-boxes, human annotated segmentation masks and induced masks from object class latent representations, resulting in a mix of supervised, semi-supervised and weakly-supervised approaches. Therefore, despite this very explanatory learning taxonomy, mixing these forms of supervision seems the most promising research direction to minimize the amount of human supervision required by visual recognition systems.

2.2.2 Variants of Weakly Supervised Learning

Weakly supervised learning is also an umbrella term covering a variety of approaches that attempt to construct predictive models by learning with weak supervision. Just to have a taste of the coverage of this term, some authors consider semi-supervised learning a form of weakly supervised learning with incomplete labels [Zhou, 2017]. In this section, we highlight the main variants in weakly supervised learning methods emphasizing the difference between the weak form of supervision used by them.

"Weaker" Annotations. As observed by Bearman et al. [2016a], related visual recognition problems like object classification, detection and segmentation require related annotations with an increasing level of detail and production cost like image-level labels, object bounding boxes and pixel-wise segmentation masks, respectively. According to the authors, human annotators take 1 second per instance on average to assign object labels to images, while they take 10 times more to provide objects precise localization and 78 times more to provide objects extent. Therefore, it is appealing to use less expensive annotations at training time to perform a more complex but related problem at test times. Following these ideas, Shi et al. [2017] and Pathak et al. [2014] propose to perform object detection and segmentation, respectively, using only image-level labels. Likewise, in the video domain, Richard et al. proposed to perform temporal action segmentation by assigning labels to frames using a model trained on sequences [Richard et al., 2018b] or unordered sets [Richard et al., 2018a] of actions which are easily extracted from the video transcripts. De-

spite the convenience of these approaches, their performance is considerably lower than the performance of their fully supervised counterparts, although this gap has gradually reduced over the years.

Self-Supervised Learning. In addition to cheap forms of human supervision, we can also leverage the visual data itself to perform weakly supervised learning. More specifically, the visual world depicted in images and videos have structural information that can be used to train visual recognition systems without human supervision. For instance, scene context [Doersch et al., 2015], regularities between shapes and colors [Noroozi and Favaro, 2016; Zhang et al., 2016; Larsson et al., 2016], and low-level motion cues [Wang and Gupta, 2015; Jayaraman and Grauman, 2015; Pathak et al., 2017] have been used to learn unsupervised image representations. In Chapter 4, we employ the visual permutation learning framework developed in Chapter 3 in the same task and we provide a more in-depth discussion about these related methods in Section 4.1. The temporal coherency of colors in videos have been used to learn object tracking models [Vondrick et al., 2018]. Simple robot and object interactions have been used to learn the physics behind robotic manipulation [Agrawal et al., 2016]. Despite there exist similar strategy on the literature [Mikolov et al., 2013; de Sa and Ballard, 1993], this learning paradigm has been renamed as self-supervised learning and attracted a lot of attention recently. The techniques following this paradigm usually define auxiliary tasks whose the solutions rely on the same visual cues than the solutions of a target task. Consequently, a model trained on the auxiliary task can also solve the target task with minimal modifications. A major drawback of such an approach is the required domain knowledge to propose and setup these auxiliary tasks avoiding solutions that are not useful for the target task.

Webly Supervised Learning. In a similar fashion, the web data can be seen as human-free form of weak supervision. Researchers have pushed hard to be able to learn visual recognition systems from the enormous amount of visual data online. Early works focused on building large datasets with minimal supervision by exploiting image search engines [Fergus et al., 2010; Schroff et al., 2011; Li and Fei-Fei, 2010], while more recent ones propose algorithms to handle the noise and bias existent in the web data by employing clustering and outlier detection techniques [Golge and Duygulu, 2014], curriculum learning [Chen and Gupta, 2015], and attention mechanisms [Zhuang et al., 2017], just to name a few. Since the web data are constantly increasing and changing, there are also initiatives to develop autonomous learning systems that continuously increase and update their knowledge base [Divvala et al., 2014; Chen et al., 2013]. A good example of what can be accomplished by such an approach, the system named NEIL collected an ontology of 1152 object categories, 1034 scene categories and 87 visual attributes after being continuously running for 2.5 months [Chen et al., 2013]. However, despite these webly supervised systems have seen orders of magnitudes larger number of images, their performance has never matched up against contemporary methods that receive extensive supervision from humans. On the other hand, they have shown very promising results when used as a

large-scale unsupervised pretraining and transfer learning approach [Mahajan et al., 2018].

2.2.3 Data Generation Approaches

Due to the rich literature, simplicity and better performance presented by supervised methods, researchers have investigated ways to efficiently curate visual recognition datasets in order to train supervised models for a vast range of visual recognition application. Consequently, in the last years, we have seen the data curation process evolve from a human-driven operation (first performed by domain experts and later by crowd-sourcing mechanisms), passing through web data automatic collection as discussed before, to a research field aiming at automatically producing very large datasets spanning a representative amount of the visual world for the task of interest. In such a research field, there are three main approaches: Synthetic data generation, data augmentation techniques, and data automatic annotation.

Learning From Synthetic Data. Learning from synthetic data consists on generating images or videos depicting the visual concepts of interest and their variations to enable training of supervised machine learning models that will be used at test time to recognize new instances of these concepts in real images and videos. Aided by developments in computer graphics, synthetic data can be cheaply and efficiently generated for a vast range of application. For instance, Dosovitskiy et al. [2015] trained an optical flow estimation model using synthetically generated images of moving chairs, Peng et al. [2015] tackle object detection by rendering 3D objects with different object/background texture and color features, Fanello et al. [2014] render synthetic infrared images of hands and faces to predict depth, Gaidon et al. [2016] have released the Virtual KITTI dataset which allows studies in multi-object tracking by leveraging synthetically generated videos of cars, Rahmani and Mian [2016] strive for 3D action recognition by leveraging synthetically generated videos in novel viewpoints, and Tokmakov et al. [2019] learn a neural network based model for segmenting objects in videos using synthetic data of moving objects. In summary, this direction seems to be very promising, especially in conjunction with deep architectures which can leverage large amounts of data to perform accurate prediction.

Although there exists a vast amount of works produced in this direction, using machine learning models trained only on synthetic data, the majority of them can not accomplish results comparable to supervised models trained on real data [Movshovitz-Attias et al., 2016]. The main reason for that is the well known reality-gap: Neural networks trained on only synthetic data often fail to generalize to real images. In order to bridge this reality gap, researchers have used auxiliary real images and other techniques to improve its performance on real images or videos. For instance, Shrivastava et al. [2017] and Bousmalis et al. [2017] propose to use generative adversarial models to generate realistic images from synthetics ones during training, while Tobin et al. [2017] and Barbosa et al. [2018] use domain adaptation strategies to adapt the model predictions from synthetic to the realistic domain at

testing time. These works have provided significant performance gains [Su et al., 2015; Sadeghi and Levine, 2016] and made synthetic data generation a feasible data source for some visual recognition problems.

Data Augmentation. The process of data generation of photo-realistic images or videos can be very computationally expensive for some tasks, e.g., action recognition may require to run heavy game simulators as described in [Roberto de Souza et al., 2017]. Furthermore, we already have large scale datasets of real images and videos for many visual recognition applications like object detection [Lin et al., 2014] and action recognition [Caba et al., 2015]. Therefore, it is plausible to use synthetic data generation procedures to augment the existing datasets allowing to train more capable models for visual recognition. Following these ideas, Varol et al. [2017] render 3D humans in different real scenes and use this data for pose estimation, Gupta et al. [2016] automatically add text to natural scenes in a manner compatible with the scene geometry in order to learn an efficient text detector, while Dwibedi et al. [2017] use real images of both objects and backgrounds to compose new scenes from the existing ones to train object detectors. These approaches combining real and synthetic data can achieve very good results and improve the performance of existing models on real data.

Automatic Annotation. As discussed before, the annotation process to curate visual recognition datasets can be very time consuming like drawing a tight bounding box around every object of interest in a image for object detection or labeling every pixel of a image for semantic segmentation. In order to speed up this process, researchers have also proposed efficient ways to produce these annotations by exploring different computational tools or even machine learning models trained on smaller and simpler data. As examples, Papadopoulos et al. [2014] propose to track the eyes movement of annotators to automatic produce rough bounding boxes to train object detectors, Papadopoulos et al. [2016] use the human feedback to improve the quality of weak detectors, Bearman et al. [2016b] and Papadopoulos et al. [2017] reduce the annotation effort from large image regions to only points of interest to train object detection and segmentation models, Castrejón et al. [2017] and Acuna et al. [2018] provide an annotation tool that uses a machine learning model to infer the vertices of the polygon outlining the object in a given image crop which is subsequently used for training object segmentation models, and Xiong et al. [2019] and Croitoru et al. [2019] propose algorithms able to jointly learn foreground object segmentation models and automatic data annotation procedures from unlabeled videos. In summary, these approaches still rely on supervised models for visual recognition, but provides an efficient way to produce human annotated data to feed them.

2.2.4 Exploring External Sources of Supervision

In the era of big data, we should not restrict ourselves only to vision datasets. We instead should explore the plethora of information existent in different domains and

modalities to reduce the amount of human supervision used to train visual recognition systems. Following these ideas low-shot, cross-modal and transfer learning techniques have attracted a lot of attention of the research community recently. They present good results on reducing the amount of labelled data and increasing the number of visual concepts that can be recognized by visual recognition models.

Few, One, And Zero-Shot Learning. Few, one or zero shot learning methods focus on solving a given task using few, one or even any example of that task at training phase. Taking zero-shot learning as example, Lampert et al. [2009] recognize new object categories, Gan et al. [2016] predict novel human actions, and Wang et al. [2019] generate out-of-domain video captions without training instances supporting these tasks. As suggested before, these models accomplish such a challenging task by exploring some external source of information like object–attributes relationships [Bucher et al., 2016; Zhang et al., 2017a], verb-attribute induction [Zellers and Choi, 2017], knowledge bases [Lei Ba et al., 2015; Wang et al., 2019], word embedding learned on a large corpus [Socher et al., 2013; Xu et al., 2017c], and textual description from web data [Niu et al., 2018; Habibian et al., 2017]. Therefore, these models attempt to overcome the closed world assumption made by the fully supervised approaches, allowing the visual recognition of new visual concepts without additional data annotation, but relying on some external source of information.

Aiming at the same goal, in this thesis, Chapters 5 and 6 propose a compositional model for recognizing unseen objects and activities in visual data, respectively. However, these unseen visual concepts are expressed by a a far more expressive language than simple labels and a far less ambiguous language than natural language queries. We also do not use any external source of information, since we explore the compositionality in the visual domain, such as co-occurrences and dependence of visual attributes.

Cross-Modal Learning. Likewise, cross-modal learning refers to any kind of learning that involves information obtained from more than one modality. In the context of visual recognition, we have seen works exploring thermal images [Xu et al., 2017a] for robust pedestrian detection, depth map for object recognition [Hoffman et al., 2016], and audio signals for representation learning [Owens et al., 2016]. These works focus on transfer the knowledge contained in non-visual modalities like audio signals to the visual recognition models providing robust predictions, in addition to reduce the need for large human annotated datasets. Another form or cross-modal learning that has attracted a lot of attention recently are vision–language models. They allow to recognize visual concepts described by textual sentences, consequently they scale up the number of visual concepts that can be recognized to the richness of our natural language. For instance, Hu et al. [2016a] can segment objects, Li et al. [2017] can track visual concepts , Hendricks et al. [2017] can localize actions, and Shao et al. [2018] can retrieve movies from natural language queries.

Transfer Learning. Transfer learning consists of improving the learning of a given target task through the knowledge acquired from a previously learned and related source task. In the deep learning community, it has been heavily used as an strategy to avoid over-fitting when training large models on relatively small datasets [Yosinski et al., 2014]. For instance, the state-of-the-art models for object detection [Ren et al., 2015], object segmentation [Zhao et al., 2018] and action recognition [Carreira and Zisserman, 2017] are built from neural networks models pretrained on large human annotated datasets for object classification and only fine-tuned in their respective target task. However, such a strategy tends to become inefficient as the tasks differ restricting its applicability in some applications. We also make use of this strategy in the experiments of Chapter 4, where we first train deep learning models in the proposed self-supervised task and then transfer the knowledge acquired to target tasks using relatively small datasets.

2.2.5 Active Learning

In addition to costly, large human annotated datasets may contain redundancies or uninformative samples which if removed would reduce significantly the amount of data and time necessary to train visual recognition models. The active learning framework [Settles, 2010] pursue this objective by enabling the learner to query the user or an oracle for labels for the most important/informative samples. Standard selection criteria include entropy [Joshi et al., 2009], boosting the margin for classifiers [Collins et al., 2008] and expected informativeness [Houlsby et al., 2011]. Focusing on a more realistic scenario, Vijayanarasimhan and Grauman [2011] present a live learning approach that autonomously refines its object detection models by actively requesting crowd-sourced annotations on images crawled from the Web. Recently, language models have also been used to request more information, like answer for specific questions about the content of the images, than a single image-level label [Misra et al., 2018].

In a more exploratory scenario, reinforcement learning techniques have been applied to acquire supervision direct from the environment by simulating computer games [Kulkarni et al., 2016], inverse kinematics [Baranes and Oudeyer, 2013], and motion planning for humanoids [Frank et al., 2013]. In summary, these models attempt to make fully supervised models more sample efficient when learning by labelling the most informative samples, extracting information from the images, or generating relevant data through simulations.

2.3 Chapter Summary

In this chapter, we reviewed some important background material in visual recognition. We started by discussing visual recognition problems, challenges, and applications. Next, we described a generic data-driven approach for visual recognition that is followed by most of the state-of-the-art methods in different applications. In

addition, we briefly presented the most important models for the visual recognition problems that are relevant for this thesis. Due to the limitations of the described approach, we shifted our discussion to learning visual recognition models using minimal supervision by discussing different unsupervised learning paradigms, different forms of weak supervision, and other ways to achieve such a goal.

Chapter 3

Image Ranking by Predicting Permutations

"The critical act in formulating computational theories turns out to be the discovery of valid constraints on the way the world is structured – constraints that provide sufficient information to allow the processing to succeed."

David Marr and Herbert Nishihara, 1978

Machine learning algorithms often use the structure of data in order to provide accurate and efficient solutions to difficult problems. For instance, in supervised learning-to-rank, list-wise methods exploit structural information beyond pairs of samples in order to learn better rankers [Cao et al., 2007]. Structured prediction models such as CRFs [Lafferty et al., 2001] and Structured SVMs [Tsochantaridis et al., 2004] explicitly model what structural information should be exploited by the learning algorithm. Therefore, we can say that the structural information implicit in data is crucial to machine learning applications.

Following these ideas, this chapter presents a learning framework that uses the inherent structure in data and leverages the geometry of the output space to solve image ranking tasks. As an example, consider the task of assigning a meaningful order (with respect to some visually salient attribute) to the images shown in Figure 3.1. Indeed, it is difficult to solve this task by just processing a single image or even a pair of images at a time where extracting visual cues related to the attributes is limited. The task becomes feasible, however, if one exploits the structure and the broader context by considering the entire set of images jointly. Only then do we start to recognize shared patterns that could guide the algorithm towards a solution.

The aforementioned task essentially involve learning a function that can recover the order, i.e., infer the shuffling permutation matrix (see Figure 3.1). However, such a learning problem presents many challenges. First, enumerating every possible permutation for a given set is usually infeasible since the number of permutations scales factorially with the cardinality of the set. As such, naively learning discriminative functions by enumerating all permutations is prohibitive. Second, a large amount of

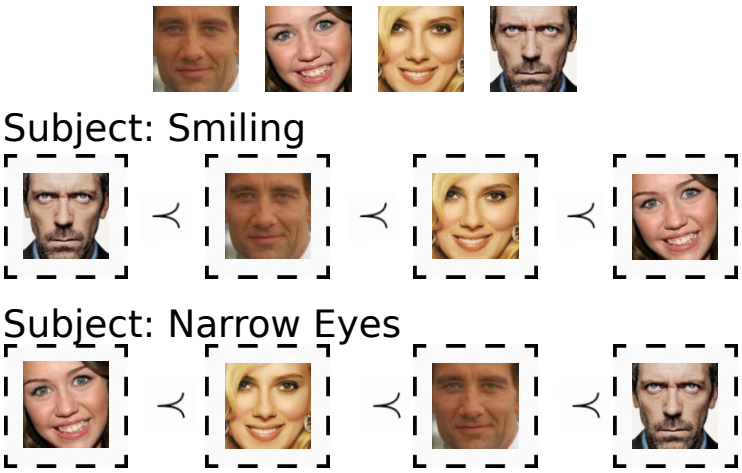


Figure 3.1: Illustration of the proposed visual permutation learning task. The goal of our method is to jointly learn visual features and the predictors to solve the visual permutation problem which consists of recovering the correct order of image sequences.

data is required to effectively learn the variations of a permutation problem, which requires more computational resources and efficient methods.

In this chapter, we address the problem of learning to predict visual permutations by leveraging the geometry of permutation matrices. Towards this end, we propose a novel permutation prediction formulation and a model based on convolutional neural networks that can be trained end-to-end. This allows us to learn image representations suitable for predicting permutations and to exploit the structure existent in the data. Moreover, our formulation admits an efficient solution and allows our method to be applied to a range of important computer vision problems. In summary, our method can be used in any problem that can be stated as a learning-to-rank problem. For instance, the list-wise learning-to-rank problem [Xia et al., 2008] can be seen as a scheme to predict the correct permutation of a random set of samples given some criteria. Recommendation problems can be cast as selecting a subset of items permuted according to the users' profiles. In archeology, broken relics may be re-assembled by permuting fragments [Brown et al., 2008]. Therefore, the proposed learning framework for such a task would benefit different applications.

Our contributions are threefold. First, we propose the *Visual Permutation Learning* problem as a generic task to learn structural concepts in ordered image sequences. Second, we formulate such a problem as the prediction of the permutation matrix that recovers the structure of the data from shuffled samples of it. Since permutation matrices are discrete, we extend our formulation to their nearest convex surrogate, doubly-stochastic matrices. From this proposed formulation, we develop an exact solution by deriving and solving a *bi-level optimization* problem and an approximated solution by using the iterative procedure *Sinkhorn normalization*. Last, we propose the *DeepPermNet* model, a end-to-end learning framework to solve the visual permutation problem using convolutional neural networks. Since our approaches are defined over continuous matrices and differentiable functions, the proposed model can be

efficiently learned via backpropagation and stochastic gradient descent.

Initially, we evaluate how well our model can learn to predict permutation matrices from shuffled image sequences. We observe that our model can leverage the structure of large sequences and the geometry of permutations matrices to infer the shuffling permutation, while naive approaches are only able to work with small sequences. With our proposed approach validated, we apply our DeepPermNet model to two different image ranking applications: relative attributes and supervised learning-to-rank. We also extend the proposed model for learning self-supervised image representations in Chapter 4.

In Section 3.4.2, we demonstrate that our proposed approach can be used to compare images according to visual attributes by predicting permutations of unordered sets of images. We evaluate this strategy on the relative attributes task where we outperform state-of-the-art methods on the Public Figures and OSR datasets [Parikh and Grauman, 2011]. We also notice that our model is able to localize attributes without any explicit supervision.

In Section 3.4.3, we extend our inference for image sequences of arbitrary length by predicting permutations of fixed-length subsequences and aggregating the results with a sorting algorithm. Using this approach, we evaluate our model on learning-to-rank applications such as ranking scenes according to how interesting they look [Gygli et al., 2013] and ranking cars according to their manufacturing date [Lee et al., 2013]. In both applications, we outperform the state-of-the-art methods in all utilized ranking metrics.

It is important to emphasize that other tasks in different scientific communities can be cast as visual permutation learning. For instance, the jigsaw puzzle problem in computer graphics [Cho et al., 2010; Sholomon et al., 2013], DNA or RNA modeling in biology [Marande and Burger, 2007] and re-assembling relics in archeology [Brown et al., 2008]. However, we limit our scope to computer vision. As we have just described, we focus on image ranking applications in the current chapter, while we tackle self-supervised representation learning in Chapter 4. More specifically, we show that our visual permutation learning formulation can be used to learn features in a self-supervised manner by exploring the structure of natural images. Using our formulation as a self-supervised representation learning method, we achieve performance similar to the state-of-the-art methods on object classification, detection and segmentation on the Pascal VOC dataset [Everingham et al., 2007, 2012].

3.1 Visual Attributes and Their Applications

In this section, we review topics that are relevant to the applications considered in this chapter. We start by describing visual attributes and different treatments given by the computer vision community. We then describe their vast range of applications and finish with a brief presentation of existing models for these applications.

Visual attributes are human understandable visual properties shared among images. They may range from simple visual features (such as “narrow eyes” and “bushy

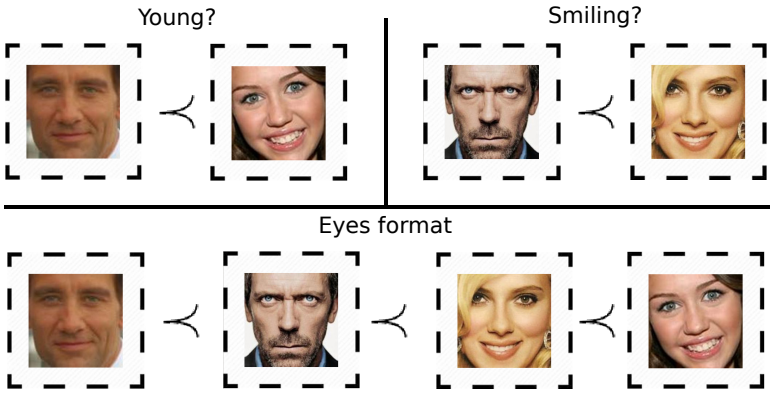


Figure 3.2: Imagine you want to learn a ranking function. The pair of face images on the top row may suggest that the criterion used to rank these faces is the age, the level of "smiling" or the eyes format. On the other hand, the image sequence in the bottom row makes clear that the format of the eyes is the most plausible ranking criterion since the level of "smiling" and age would flip the order of the first two pictures, according to the ground-truth information. Therefore, list-wise annotations are more complete and less ambiguous which simplifies the learning of ranking functions.

eyebrows" in faces) to semantic concepts (like "natural" and "urban" scenes), or subjective concepts (such as "memorability" and "interestingness" of images). Due to the expressiveness of visual attributes, researchers have successfully used them for many applications, including image search [Kovashka et al., 2012], fine-grained recognition [Branson et al., 2013], and zero-shot learning [Parikh and Grauman, 2011; Lampert et al., 2014].

Visual attributes are traditionally treated as binary predicates indicating the presence or absence of certain properties in an image. From this perspective, most of the existing methods use supervised machine learning, whose goal is to provide mid-level cues for object and scene recognition [Farhadi et al., 2010], or to perform zero-shot transfer learning [Lampert et al., 2014].

However, there are also methods that can discover binary visual attributes in an unsupervised way [Shankar et al., 2015; Huang et al., 2016]. Huang et al. [2016] use unsupervised discriminative clustering and cluster membership as a soft form of supervision to discover shared attributes. In contrast, our formulation directly learns the properties of visual attributes in a data driven manner using a single end-to-end trainable network able to explore the entire structure of the data.

A more natural view on visual attributes is to measure their strength in visual entities. For instance, Parikh and Grauman [2011] introduced the problem of relative attributes, in which pairs of visual entities are compared with respect to their relative strength for any specific attribute. This problem is usually cast as a learning-to-rank problem using pair-wise constraints. Following this idea, Parikh and Grauman [2011] propose a linear relative comparison function based on the well-known Rank-SVM [Joachims, 2006], while Yu and Grauman [2014] uses a local learning strategy.

With the recent success of deep learning methods in computer vision, CNN-based methods to tackle the relative attributes problem have been developed. Souri

et al. [2016] jointly learn image representation and ranking network to perform pair-wise comparisons according to a certain attribute. Similarly, Singh and Lee [2016] propose to combine spatial transformer networks [Jaderberg et al., 2015] and rank networks to localize, in addition to compare visual attributes. Differently from our proposed approach, the aforementioned methods use only pair-wise relationships. As explained in Section 2.1.3, these pair-wise learning-to-rank approaches are often less computationally expensive, but not very accurate when compared to list-wise methods which can leverage structure within longer image sequences. For instance, imagine you want to learn a ranking function from the pairs of face images in the top row of Figure 3.2. The pair of face images on the top row may suggest that the criterion used to rank these faces is the age, the level of "smiling" or the eyes format. On the other hand, the image sequence in the bottom row makes clear that the format of the eyes is the most plausible ranking criterion since the level of "smiling" and age would flip the order of the first two pictures, according to the ground-truth information. In summary, list-wise annotations are more complete and less ambiguous which simplifies the learning of ranking functions. Therefore, we propose a method that can leverage structure within longer image sequences to learn accurate image rankers.

3.2 Preliminaries

In this section, we review the following background topics that we use in the subsequent sections for deriving our model for permutation learning: permutation matrices, doubly stochastic matrices and bi-level optimization.

3.2.1 Permutation Matrices

In matrix theory, a permutation matrix is a binary square matrix that has exactly a single unit value in every row and column, and zeros elsewhere. These matrices are used to compactly represent permutations of elements in an ordered sequence. For instance, given an ordered sequence $S = \langle a_1, \ldots, a_n \rangle$ of n elements any permutation $\pi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ can be uniquely represented by a permutation matrix P_π. Furthermore, if we describe the original ordered sequence as a column vector, then any desired permutation π can be obtained by a simple matrix-vector multiplication,

$$S_\pi = P_\pi \, S \tag{3.1}$$

where P_π is formed by swapping the rows of the identity matrix according to the desired permutation π.

The set of $n \times n$ permutation matrices is a subgroup in the group of nonsingular matrices in $\mathbb{R}^{n \times n}$ with cardinality $n!$. These matrices have very interesting and useful properties. For instance, permutation matrices are closed under multiplication, that is, the product of two permutation matrices is again a permutation matrix representing the combined permutation. Likewise, the inverse of a permutation matrix is the

inverse permutation, i.e., the permutation that recovers the original sequence from the permuted sequence, that can be efficiently computed by $P^{-1} = P^T$ (orthogonality).

3.2.2 Doubly Stochastic Matrices

A nonnegative matrix with the property that all its rows sum to one, is said to be a row stochastic matrix. Likewise, its transpose is said to be column stochastic matrix, since all its columns sum to one. A matrix that is simultaneously row and column stochastic is said to be a doubly stochastic matrix (DSM). Mathematically, the requirements for a matrix $A \in \mathbb{R}^{n \times n}$ to be doubly stochastic are,

$$A_{ij} \geq 0, \qquad A\,\mathbf{1} = \mathbf{1}, \qquad A^T\,\mathbf{1} = \mathbf{1}, \tag{3.2}$$

where $\mathbf{1}$ is an n-dimensional column vector of ones.

Permutation matrices are doubly stochastic matrices. In fact, according to the Birkhoff-von Neumann theorem [Birkhoff, 1946; Von Neumann, 1953], any doubly stochastic matrix is a convex combination of finitely many permutation matrices. Thus, the set of $n \times n$ DSMs forms a convex hull for the set of $n \times n$ permutation matrices, known as the Birkhoff polytope $\mathcal{B}^n$. Consequently, it is natural to think of DSMs as convex relaxations of permutation matrices. Figure 3.3(a) illustrates the geometry of the Birkhoff polytope.

Doubly stochastic matrices, as well as permutation matrices, have a prominent history in engineering ranging from cryptography to topics in communication theory [Brualdi, 1988]. And approximating doubly stochastic matrices is a key problem in many applications. Here, we briefly demonstrate two efficient and principled approaches to fulfill such tasks. In later sections, we explain how these approaches can be applied in gradient based learners to solve our proposed permutation learning problem.

3.2.2.1 The Sinkhorn-Knopp algorithm

The Sinkhorn-Knopp [Sinkhorn and Knopp, 1967] theorem states that if A is a real nonnegative squared matrix and has total support, then there exists a doubly stochastic matrix Q of the form,

$$Q = D_l A D_r \tag{3.3}$$

where D_l and D_r are diagonal matrices with positive main diagonals. Furthermore, there is a simple iterative procedure known as Sinkhorn Normalization, which can find D_l and D_r by repeatedly rescaling the rows and columns of a given matrix.

Knight [2008] analyzes the convergence guarantees of Sinkhorn-Knopp algorithm. The author states that for a matrix A with entries in $[1, V]$, $\mathcal{O}(V\,|\log \epsilon|)$ iterations suffice to reach ϵ-near double stochasticity. However, we noticed empirically that only a few iterations are sufficient to reach acceptable approximations for most of the problems that we consider. Figure 3.3 shows empirical results for approximating DSMs

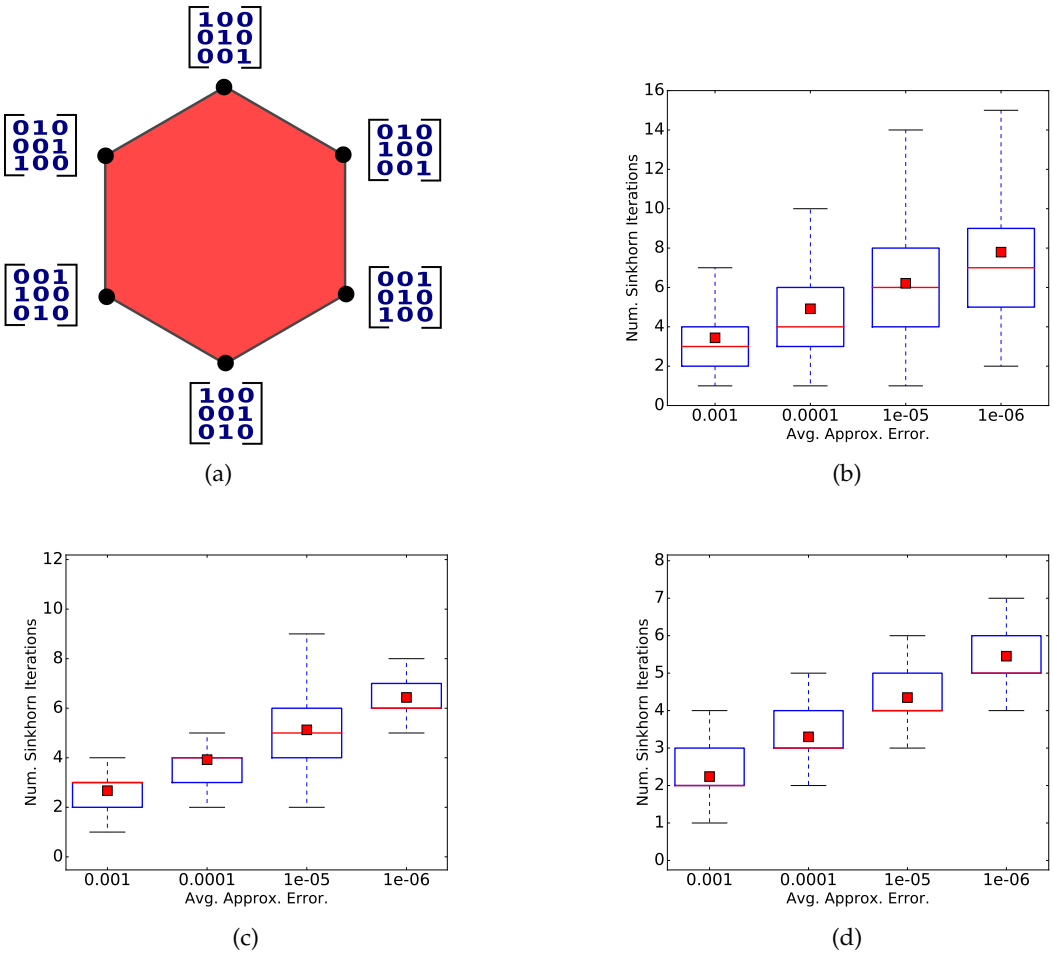


Figure 3.3: Figure 3.3(a) Illustrates the Birkhoff polytope for $n \times n$ permutation matrices. Figures 3.3(b), 3.3(c), and 3.3(d) shows boxplots of the approximation error for the Sinkhron-Knopp algorithm applied on nonnegative random matrices of size 3x3, 6x6 and 9x9, respectively.

from nonnegative random matrices of sizes 3×3, 6×6 and 9×9 using the Sinkhorn-Knopp algorithm. We find it to converge to an acceptable accuracy in approximately 4–6 iterations in most cases.

3.2.2.2 Norm Approximation

Norm approximation is a well known problem in the field of convex optimization [Boyd and Vandenberghe, 2004]. The goal of the norm approximation problem is to approximate a vector, matrix, or space, as closely as possible, with deviation measured in the norm $\|\cdot\|$. We can cast the doubly stochastic approximation problem as a norm approximation problem. Formally, given an arbitrary matrix $A \in \mathbb{R}^{n \times n}$, its closest doubly stochastic matrix $Q \in \mathcal{B}^n$ can be obtained by solving the following

problem,

$$\begin{aligned} \underset{Q \in \mathbb{R}_{+}^{n \times n}}{\text{minimize}} \quad & \|Q - A\| \\ \text{subject to} \quad & Q\,\mathbf{1} = \mathbf{1} \\ & Q^T\,\mathbf{1} = \mathbf{1} \end{aligned} \tag{3.4}$$

which is a convex optimization problem. Thus, the solution is globally optimal. Moreover, when the norm $\|\cdot\|$ is the frobenius norm, this problem can be stated as a quadratic program (QP) which can be solved efficiently by most publicly available solvers [Gurobi Optimization, 2016].

3.2.3 Bi-level Optimization

Given our interest in learning end-to-end models and solving DSMs approximation problems optimally in the derivation of our visual permutation learning framework, we need to deal with a bi-level optimization problem. We describe our specific problem in details in Section 3.3.3.1. Here we present a generic formulation for bi-level optimization problems and discuss how to solve them.

A bi-level optimization problem consists of an upper problem and a lower problem, whose objectives (and constraints) share a set of variables. More specifically, the former defines an objective over two sets of variables, say $\boldsymbol{x}$ and $\boldsymbol{y}$, and the latter binds $\boldsymbol{y}$ as an optimization problem parametrized by $\boldsymbol{x}$. We can state the problem mathematically as,

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f\left(\boldsymbol{x}, \boldsymbol{y}\right) \\ \text{subject to} \quad & \boldsymbol{y} \in \underset{\boldsymbol{y}'}{\text{argmin}}\; h\left(\boldsymbol{x}, \boldsymbol{y}'\right) \end{aligned} \tag{3.5}$$

where f and h are the upper and lower level objectives, respectively. Recently, bi-level optimization problems have found applications in machine learning and computer vision where they have been applied to hyper-parameter learning [Wohlhart et al., 2015], image denoising [Ochs et al., 2015], and video activity recognition [Fernando and Gould, 2016].

We can solve such a problem by rewriting it as an equivalent single-level problem. This can be done by replacing the lower problem with an analytical solution (e.g., normal equations for a least-square problem) or a set of sufficient conditions for optimality (e.g., the KKT for convex problems). Then, the bi-level problem can be solved using the resulting single-level problem. However, for many lower problems either an analytical solution does not exist or the optimality conditions are hard to express. Furthermore, the resulting problem may be hard to solve.

However, if the lower problem can be solved efficiently, and there exists a method for finding the gradient at the current solution, we can solve the bi-level optimization problem via gradient descent. The main idea is to compute the gradient of the solution to the lower problem with respect to variables in the upper problem and

perform updates of the form,

$$x \leftarrow x - \alpha \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial x} \right) \Bigg|_{(x, y^\star)} \tag{3.6}$$

Note that the partial derivative $\frac{\partial y}{\partial x}$ may be difficult to compute, since it typically involves a parametrized argmin or argmax optimization problem. For a detailed explanation about procedures to differentiating such problems, we refer the readers to Faugeras [1993] and Gould et al. [2016].

3.3 Visual Permutation Learning

In this section, we describe our method for learning visual permutations. We start by formalizing the visual permutation learning task. Then, we describe our end-to-end learning algorithm, deep learning model, and inference procedure. We finish this section by discussing alternative approaches to our method.

3.3.1 Task Formulation

Let us start by considering the task illustrated in Figure 3.4. Given a sequence of images ordered by a pre-decided visual criterion, for instance “smiling”, we generate shuffled sequences by applying randomly sampled permutation matrices to the original sequences. Similarly, we can recover the original sequence from the shuffled ones by “un-permuting" them using the inverse of the respective permutation matrices. In this context, we define the *visual permutation learning* task as one that takes as input a permuted sequence and produces as output the permutation matrix that shuffled the original sequence.

Formally, let us define X to be an ordered sequence of l images in which the order explicitly encodes the strength of some predetermined criterion c. For example, c may be the degree of “smilingness” in each image. In addition, consider an artificially permuted version $\tilde{X}$ where the images in the sequence X are permuted by a randomly generated $l \times l$ permutation matrix P. Formally, the permutation prediction task is to predict the permutation matrix P from a given shuffled image sequence $\tilde{X}$ such that $P^{-1} = P^T$ recovers the original ordered sequence X.

We also hypothesize that deep models trained to solve this task are able to capture high-level semantic concepts, structure, and shared patterns in visual data (In Chapter 4 and Section 3.4, we provide empirical evidence supporting this hypothesis). The ability to learn these concepts is important to perform well on the permutation prediction task, as well as to solve many other computer vision problems. Therefore, we posit that the features learned by our models are transferable to other related computer vision tasks as well.

Note that we describe our problem using only ordered sequences. This may seem a limitation, since structured information may be better represented by higher dimensional data. However, most of the time these higher order representations

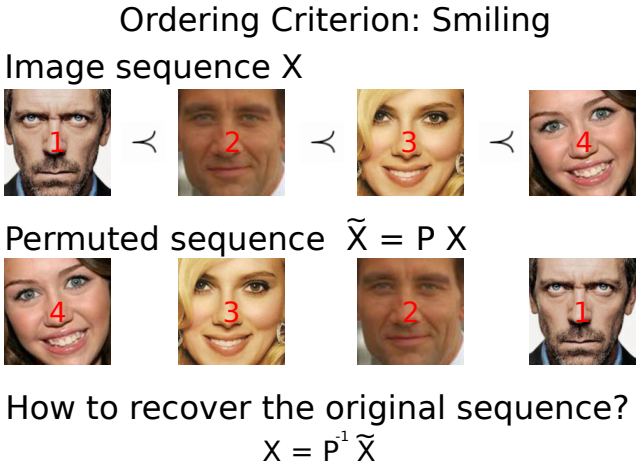


Figure 3.4: Illustration of Permutation learning task as the prediction of the permutation matrix P from a given permuted image sequence $\tilde{X}$ such that $P^{-1} = P^T$ recovers the original ordered image sequence X.

can be efficiently encoded as ordered sequences. For instance, the placement of 2-D image regions can be represented as an ordered sequence, where every position in the sequence is mapped to a distinct position in the 2D layout. Therefore, the proposed task can be used to solve different problems encoded in terms of ordered sequences.

3.3.2 Learning Objective

With the visual permutation learning task described, we now focus on how to solve such a problem. We define a training set $\mathcal{D} = \{(X, P) \mid X \in \mathcal{S}^c \text{ and } \forall P \in \mathcal{P}^l\}$ composed by tuples of ordered image sequences X and permutation matrices P. Here, $\mathcal{S}^c$ represents a dataset of ordered image sequences, orderings implied by a predetermined criterion c. Each $X \in \mathcal{S}^c$ is composed of $X = \langle I_1, I_2, \ldots, I_l \rangle$, an ordered sequence of images I_i. The notation $\mathcal{P}^l$ represents the set of all $l \times l$ permutation matrices. Accordingly, the training set $\mathcal{D}$ is composed of all shufflings of each X by all P. Note that given an ordered X, such a dataset can be generated on-the-fly by randomly permuting the order, and the size of such permuted sets scales factorially on the sequence length l, providing a huge amount of data with low processing and storage cost to train high capacity models.

Directly working with permutation matrices for deriving gradient-based optimization solvers is difficult as such solvers often start with an initial point and iteratively refine it using small steps (stochastic updates along gradient directions) towards an optimum. In this respect, working directly with discrete permutation matrices is not feasible. Thus, in this work, we propose to approximate inference over permutation matrices to inference over their nearest convex surrogate, the set of doubly-stochastic matrices. As discussed in Section 3.2, it is natural to think of DSMs as relaxations of permutation matrices.

Following these ideas, we propose to learn a parametrized function $f_\theta : \mathcal{S}^c \rightarrow \mathcal{B}^l$

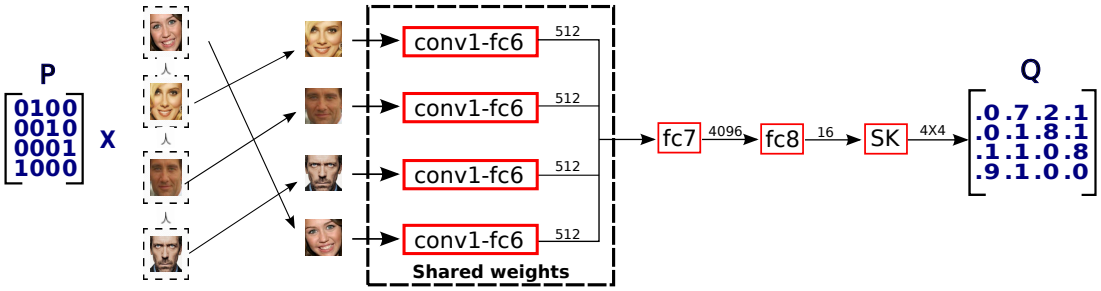


Figure 3.5: DeepPermNet Architecture. It receives a permuted sequence of images as input. Each image in the sequence goes trough a different branch that follows the AlexNet [Krizhevsky et al., 2012] architecture from *conv1* up to *fc6*. Then, the outputs of *fc6* are concatenated and passed as input to *fc7*. Finally, the model predictions are obtained by applying the Sinkhorn Layer on the outputs of *fc8* layer.

that maps a fixed length image sequence (of length l) denoted by $\tilde{X}$ to an $l \times l$ doubly stochastic matrix Q. In the ideal case, the matrix Q should be equal to P. Then, our permutation learning problem can be described as,

$$\underset{\theta}{\text{minimize}} \quad \sum_{(X,P)\in\mathcal{D}} \Delta\left(P, f_\theta(\tilde{X})\right) + R\left(\theta\right), \tag{3.7}$$

where $\tilde{X}$ is the image sequence X permuted by the permutation matrix P, $\Delta(\cdot,\cdot)$ is a loss function, θ captures the parameters of the permutation learning function, and $R(\theta)$ regularizers these parameters to avoid overfitting. The exact implementation of these components will be presented in the following sections.

3.3.3 Model Details

Having the task and learning objective defined, here we focus on the parametrization of the function $f_\theta(\cdot)$. Note that we wish to learn the image representation that captures the structure behind our sequences and also solves the permutation problem jointly. As such, the function $f_\theta(\cdot)$ should learn intermediate feature representations which encode semantic concepts about the input data. We propose to implement the function $f_\theta(\cdot)$ as a convolutional neural network (CNN), which is able to exploit large datasets and learn valuable visual features, that can be used as intermediate representations, while jointly learning the required mapping.

More specifically, we use a Siamese type of convolutional neural network in which each branch receives an image from a permuted sequence $\tilde{X}$ (see Figure 3.5). Each branch up to the first fully connected layer *fc6* uses the AlexNet architecture [Krizhevsky et al., 2012]. The outputs of *fc6* layers are concatenated and given as input to *fc7*. All layers up to *fc6* share the same set of weights. We refer to our proposed model as **DeepPermNet**.

Note that, if we ignore the structure of permutation matrices, this problem can

have many different naive and infective solutions which we discuss later in Section 3.3.5. However, incorporating the inherent structure of permutation matrices can avoid the optimizer from searching over impossible solutions, thereby leading to faster convergence and better solutions. Thus, in the sequel, we discuss approaches for the permutation learning problem that explore the geometry of permutation matrices (using doubly-stochastic approximations).

3.3.3.1 Bi-level Optimization

Note that we wish to provide the closest doubly stochastic matrix $\hat{Q} \in \mathcal{B}^l$ from an arbitrary matrix $Q \in \mathbb{R}_+^{l \times l}$, e.g., CNN outputs. A principled way to achieve such a objective is to define and solve a convex quadratic program (QP). In this way, we can restate our learning problem in Equation 3.7 as,

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \sum_{(X,P)\in\mathcal{D}} \Delta\left(P, \hat{Q}\right) + R\left(\theta\right) \\ \text{subject to} \quad & \hat{Q} \in \underset{Q\in\mathcal{B}^l}{\text{argmin}} \ \left\|Q - f_\theta(\tilde{X})\right\|_F \end{aligned} \tag{3.8}$$

where $\mathcal{B}^l$ is the Birkhoff polytope.

This formulation is an instance of a bi-level optimization problem discussed in Section 3.2. Here, the loss minimization is the upper problem and the doubly stochastic approximation is the lower problem. Furthermore, this formulation is well suited for gradient-based optimization methods, provided that we can compute the gradient of the argmin function in our lower level problem as other authors observed [Domke, 2012; Ochs et al., 2015; Fernando and Gould, 2016].

In order to simplify the gradient computation, we can approximate the lower level problem in Equation 3.8 by the following function $\boldsymbol{h}\left(\cdot\right)$,

$$\begin{aligned} \boldsymbol{h}\left(\boldsymbol{q}\right) \ = \ & \underset{\hat{\boldsymbol{q}}\in\mathbb{R}^n}{\text{argmin}} \quad \tfrac{1}{2}\left\|\hat{\boldsymbol{q}} - \boldsymbol{q}\right\|_2^2 - \mu \textstyle\sum_{i=1}^n log\left(\hat{q}_i\right) \\ & \text{subject to} \quad A\hat{\boldsymbol{q}} = \mathbf{1} \end{aligned} \tag{3.9}$$

where $\boldsymbol{q}, \hat{\boldsymbol{q}} \in \mathbb{R}^n$ are the vectorized versions of Q and $\hat{Q}$ respectively $\left(n = l^2\right)$. The equality constraints defined by $A \in \mathbb{R}^{(2l)\times n}$ and the log-barrier function approximates the Birkhoff polytope. The hyper-parameter $\mu \geq 0$ controls the quality of the approximation. As $\mu \to 0$ the solution to the problem in Equation 3.9 converge to the solution to the lower problem in Equation 3.8.

Recently, Gould et al. [2016] reviewed an earlier work of Faugeras [1993] and collected some results on differentiating argmin and argmax optimization problems. Here, we will make use of their Lemma 4.2 for linearly constrained argmin problems that is restated in Lemma 3.3.1 below.

Lemma 3.3.1: Let $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ be a continuous function with first and second derivatives. Let $A \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$ with $\text{rank}(A) = m$. Let $\boldsymbol{g}(x) =$

$\text{argmin}_{\boldsymbol{y}:A\boldsymbol{y}=\boldsymbol{b}} f(x,\boldsymbol{y})$. Let $H = f_{YY}(x,\boldsymbol{g}(x))$. Then,

$$\boldsymbol{g}'(x) = \left(H^{-1}A^T\left(AH^{-1}A^T\right)^{-1}AH^{-1} - H^{-1}\right) f_{XY}(x,\boldsymbol{g}(x)) \tag{3.10}$$

where $f_{YY} \doteq \nabla^2_{\boldsymbol{yy}} f(x,\boldsymbol{y}) \in \mathbb{R}^{n\times n}$ and $f_{XY} \doteq \frac{\partial}{\partial x}\nabla_{\boldsymbol{y}} f(x,\boldsymbol{y}) \in \mathbb{R}^n$.

However, note that $\boldsymbol{h}(\cdot)$ is a vector-valued function on vector domain. As such, we compute the derivative with respect to each input entry separately $\nabla_{q_i}\boldsymbol{h}(\boldsymbol{q}) \in \mathbb{R}^n$ and aggregate the results to compose the gradient $\nabla\boldsymbol{h}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$. Finally, $H = \nabla^2_{\hat{\boldsymbol{q}}\hat{\boldsymbol{q}}} f(\boldsymbol{q},\hat{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ and $\frac{\partial}{\partial q_k}\nabla_{\hat{\boldsymbol{q}}} f(\boldsymbol{q},\hat{\boldsymbol{q}}) \in \mathbb{R}^n$ where $f(\boldsymbol{q},\hat{\boldsymbol{q}}) \in \mathbb{R}$ is the objective in Equation 3.9, can be obtained using the following partial derivatives,

$$H_{i,j} = \frac{\partial f^2(\boldsymbol{q},\hat{\boldsymbol{q}})}{\partial\hat{q}_i\partial\hat{q}_j} = [\![i=j]\!]\left(1+\mu\hat{q}_i^{-2}\right) \tag{3.11}$$

$$\frac{\partial}{\partial q_k}\nabla_{\hat{\boldsymbol{q}}} f(\boldsymbol{q},\hat{\boldsymbol{q}}) = -[\![i=k]\!], \tag{3.12}$$

where $[\![\cdot]\!]$ is the indicator function, evaluating to one if its argument is true and zero otherwise. Note the gradient $\nabla_{q_i}\mathbf{h}(\boldsymbol{q})$ can be efficiently computed because H is a diagonal matrix and the derivative $\frac{\partial}{\partial q_k}\nabla_{\hat{\boldsymbol{q}}} f(\boldsymbol{q},\hat{\boldsymbol{q}})$ does not depend on $\boldsymbol{q}$. Finally, the gradient of the loss with respect to the inputs can be easily obtained by applying the chain rule.

3.3.3.2 Sinkhorn Normalization

Despite providing the optimal solution for the DSM approximation, the bi-level optimization approach may be computationally expensive, since we need to solve an optimization problem for every sample in the training batches. Alternatively, we can resort to an approximate solution based on the Sinkhorn-Knopp algorithm discussed in Section 3.2.

Inspired by Adams and Zemel [2011], here we propose a CNN layer that performs Sinkhorn normalization. Consider a matrix $Q \in \mathbb{R}_+^{l\times l}$, which can be converted to a doubly stochastic matrix by repeatedly performing row and column normalizations. Define row $R(\cdot)$ and column $C(\cdot)$ normalizations as follows,

$$R_{i,j}(Q) = \frac{Q_{i,j}}{\sum_{k=1}^{l} Q_{i,k}}; \quad C_{i,j}(Q) = \frac{Q_{i,j}}{\sum_{k=1}^{l} Q_{k,j}} \tag{3.13}$$

Then, the Sinkhorn normalization for the n-th iteration can be defined recursively as:

$$S^n(Q) = \begin{cases} Q, & \text{if } n = 0 \\ C\left(R\left(S^{n-1}(Q)\right)\right), & \text{otherwise.} \end{cases} \tag{3.14}$$

The Sinkhorn normalization function $S^n(\cdot)$ is differentiable and we can compute its gradient w.r.t. the inputs by unrolling the normalization operations and propa-

gating the gradient through the sequence of row and column normalizations. For instance, the partial derivatives of the row normalizations can be defined as,

$$\frac{\partial \Delta}{\partial Q_{p,q}} = \sum_{j=1}^{l} \frac{\partial \Delta}{\partial R_{p,j}} \left[\frac{[\![j = q]\!]}{\sum_{k=1}^{l} Q_{p,k}} - \frac{Q_{p,j}}{\left(\sum_{k=1}^{l} Q_{p,k} \right)^2} \right] \tag{3.15}$$

where Q and R are the inputs and outputs of the row normalization function. The derivative of the column normalization can be obtained by transposing indexes appropriately. In practice, before applying the Sinkhorn normalization, we add a small value ($\approx 10^{-3}$) to each entry of Q as a regularization term to avoid numerical instability.

Despite being a principled and efficient approach, the Sinkhorn normalization layer may have a notorious drawback from the CNN optimization point of view – the problem of vanishing gradients in deep networks [Glorot and Bengio, 2010]. This may happen because each normalization can be seen as an extra layer to the network which makes the network deeper. However, as observed for random matrices in Figure 3.3, a small number of normalizations are sufficient to approximate the doubly stochastic matrix from CNN's raw outputs, and consequently the vanishing gradients problem is avoided.

3.3.4 Inference Algorithm

Finally, we describe the last component of our approach, the inference procedure. Our main goal is to recover the original image sequence from a permuted sequence. Thus, our inference consists of approximating the closest permutation matrix $\hat{P}$ from the predicted doubly stochastic matrix Q. This problem can be described as,

$$\begin{aligned} \hat{P} \in \underset{P}{\operatorname{argmin}} \quad & \|P - Q\|_F \\ \text{subject to} \quad & P \cdot \mathbf{1} = \mathbf{1} \\ & \mathbf{1}^T \cdot P = \mathbf{1} \\ & P \in \{0,1\}^{l \times l} \end{aligned} \tag{3.16}$$

where $\hat{P}$ is our approximated permutation matrix from Q.

This problem is an instance of a mixed-boolean program and can be efficiently solved by branch-and-bound methods available in public solvers [Diamond and Boyd, 2016]. These methods begin by finding the optimal solution to the convex "relaxation" of the problem without the boolean constraints. If the optimal solution has any non-boolean variables, it creates new subproblems where the variables are more tightly constrained and this process is repeated until a solution that satisfies all boolean constraints is found.

After solving this problem to obtain $\hat{P}$, we transpose it to compute the inverse permutation matrix since $\hat{P}^T = \hat{P}^{-1}$. Then we can recover the original sequence X

Algorithm 3.1 Simple bubble sort style algorithm for ordering longer sequence.

Input: Shuffled image sequence $\tilde{X}$.

Let L and l be the length of $\tilde{X}$ and the input size of model $f_\theta(\cdot)$, respectively.
while $L \geq 1$ **do**
  for $i = 0, \ \ldots, \ \max(0, L - l)$ **do**
    $\hat{Q} \leftarrow f_\theta\left(\tilde{X}\left[i : i + l - 1\right]\right)$. ▷ Predicting the DSM for a subsequence
    Approximate $\hat{P}$ from $\hat{Q}$ by solving the opt. problem in Equation 3.16.
    $\tilde{X}\left[i : i + l - 1\right] \leftarrow \hat{P}^T \tilde{X}\left[i : i + l - 1\right]$. ▷ ordering a subsequence
  end for
  $L \leftarrow L - l + 1$
end while

from the permuted sequence $\tilde{X}$ as,

$$X = \hat{P}^T \tilde{X}. \tag{3.17}$$

3.3.4.1 Ranking Long Sequences

In some applications like multimedia retrieval, the sequence of images to be ordered according to a given criterion may be very long. Therefore, we also want to be able to recover the original image sequence from a permuted sequence of arbitrary length. Since a sequence of length L is correctly ordered if and only if all of its subsequences of length l are correctly ordered, we extend our inference for image sequences of arbitrary length by predicting permutations of fixed-length subsequences and aggregating the results with a sorting algorithm. For simplicity, we describe in Algorithm 3.1 a bubble sort style algorithm for ordering long sequences using the visual permutation learning framework. Figure 3.6 illustrates an example of the execution of such an algorithm and Section 3.4.3 evaluates this approach on image ranking applications.

3.3.5 Alternative Approaches

In this section, we describe alternative approaches to solve the visual permutation learning problem. Overlooking all nice properties of permutation matrices, we can reformulate the permutation learning problem as a regression on the correct ordered sequence. More specifically, we can explicitly lean to predict the correct position of each item in a given shuffled input sequence by minimizing the euclidean loss between the correct sequence and the predictions. However, this solution may focus on correcting outliers, like swapping the first element by the last, which generates most part of the overall loss leading to a suboptimal solution.

Likewise, we can follow a discriminative setup (as done in [Noroozi and Favaro, 2016; Lee et al., 2017]) and cast our problem as a multi-class classification problem which we enumerate every possible permutation as a independent class. However, this solution is not feasible in practice since the number of parameter and predictors

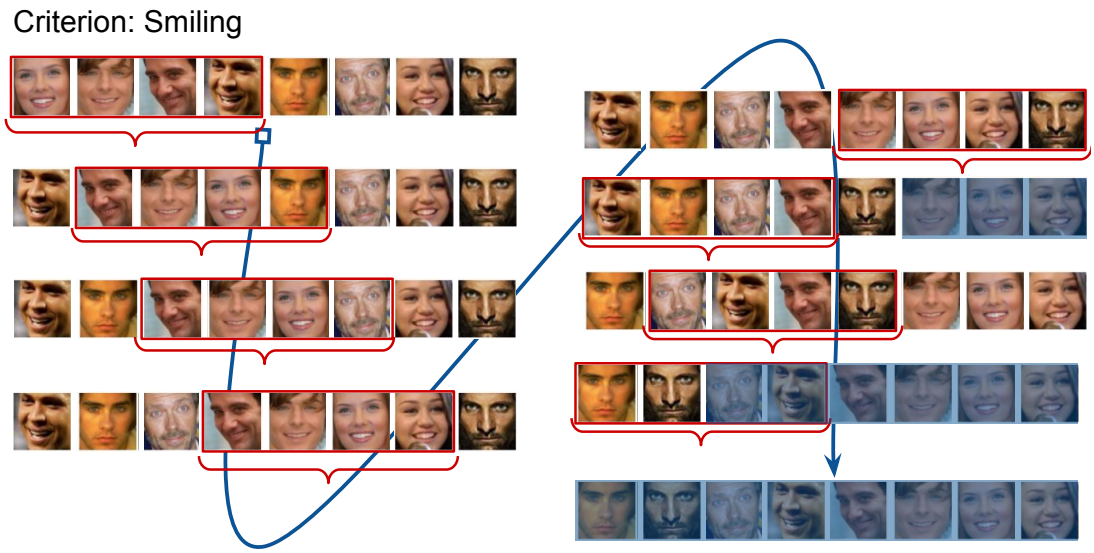


Figure 3.6: Example of the execution of the proposed algorithm to sort long sequences. We use the proposed DeepPermNet model trained on fixed-size sequences of length l and the sorting algorithm shown in Algorithm 3.1 to order long sequences of length L where $L \geq l$.

in the model scales factorial with the the input length. For instance, for a sequence of length 8 we need 40320 classes, which is intractable even for deep models.

On the other hand, we can use the permutation matrices formulation only to avoid the aforementioned enumeration problem and cast our problem as an l^2 binary classification problem by optimizing the combination of sigmoid outputs and cross-entropy loss,

$$\Delta(P, Q) = -\frac{1}{l^2}\sum_{ij}^{l\times l}\Big[P_{i,j}\log\left(Q_{i,j}\right) + \left(1 - P_{i,j}\right)\log\left(1 - Q_{i,j}\right)\Big], \tag{3.18}$$

where each entry $P_{i,j}$ is a binary entry in the target permutation matrix P and $Q_{i,j}$ is an arbitrary prediction outputted by the function $f_\theta(\tilde{X})$. We refer to this solution as naive approach since it is more related to our proposed model. Note this solution does not explore the geometry of permutation matrices and has a series of inefficiencies which will be demonstrated in our experiments.

3.4 Experiments

We now describe how our model can tackle different computer vision problems and measure our models performance on well established benchmarks. First, we give some details of the datasets used in our experiments. Second, in Section 3.4.1, we analyze how effectively our proposed model solves the permutation prediction problem under different settings. Third, in Section 3.4.2, we evaluate our model on the relative attributes task. Last, in Section 3.4.3, we evaluate our model for long sequences

using image ranking applications.

We evaluate our proposed model using the following datasets (see Figure 3.7):

Public Figures (PubFig) [Parikh and Grauman, 2011]. This dataset consists of 800 facial images of eight public celebrities annotated with eleven physical attributes, such as big lips, white, and young. This is a *relative attribute dataset* with category level annotation, i.e., all images in a specific category may be ranked higher, equal, or lower than all images in another category, with respect to an attribute. Our goal is to rank subsets of images according to these visual attributes.

Outdoor Scene Recognition (OSR) [Parikh and Grauman, 2011]. This is another *relative attribute dataset* with category level annotation. It consists of 2688 images of eight different types of outdoor scenes such as Mountain, Forest, and Coast, annotated with six different visual attributes such as natural and open. This dataset has more ties between pair of images than PubFig, which may impose some difficulties to our model as discussed in later sections.

Historical Car (CarDb) [Lee et al., 2013]. This dataset consists of 12k images of cars annotated with manufacturing information such as model and manufacturing year. In this work, we are interested in ranking the cars according to their manufacturing date. Different from the PubFig and OSR datasets, CarDb has instance-level annotations, i.e., each image may be ranked higher, equal or lower than other image. This is a harder problem, since fine-grained comparisons have to be made in order to correctly rank the images.

Interestingness Annotations. This dataset comes from an investigation of human interest in photos by Gygli et al. [2013]. Using psychophysical experiments on Mechanical Turk, they annotate the images from OSR dataset with an interestingness score which measures the degree of interestingness of an image. Our goal is to rank images according to how interesting they are. Similar to CarDb, this dataset is instance-level annotated and we use the OSR train/test splits in our experiments.

ImageNet [Krizhevsky et al., 2012]. This is a large scale dataset for object recognition. It consists of approximately 1.3M images of 1k different object categories. In our experiments, we use the training set images of this dataset discarding the labels to learn image representations in a self-supervised fashion (See Chapter 4).

Pascal VOC [Everingham et al., 2007, 2012]. This is a fine-grained object recognition dataset. It has 9,963 images containing 24,640 annotated objects of 20 different classes. This dataset provides image, bounding boxes and pixel level annotations and it is widely used in the literature. In this work, we evaluate our self-supervised image representations in this dataset for object classification, detection and segmentation.

3.4.1 Permutation Prediction

In this experiment, we evaluate our proposed method on the permutation prediction task and compare with a naïve approach which combines sigmoid outputs and cross-entropy loss by casting the permutation prediction problem as a multi-label classification problem. In this experiment, we use the Public Figures dataset [Parikh and Grauman, 2011] and its default train and test splits.

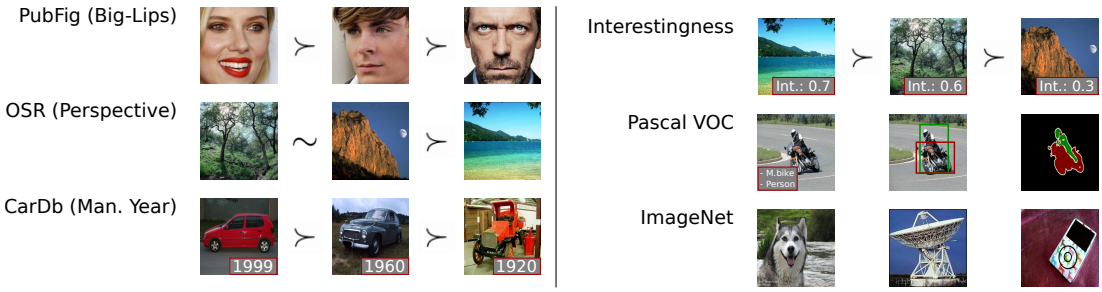


Figure 3.7: Datasets used in our experiments: PubFig and OSR [Parikh and Grauman, 2011], CarDb [Lee et al., 2013], Interestingness [Lee et al., 2013], Pacal VOC [Everingham et al., 2007, 2012] and ImageNet [Krizhevsky et al., 2012].

In our implementation, we use stochastic gradient descent with mini-batches of 32 image sequences, images of 256×256 pixels and different sequence lengths. During preprocessing, we subtract the mean and randomly crop each image to size 227×227. We initialize our network from *conv1* to *fc6* layers using an AlexNet model pre-trained on the ILSVRC 2012 [Krizhevsky et al., 2012] dataset for the task of image classification, while other layers are randomly initialized from a Gaussian distribution. We set the learning rate to 10^{-5} and fine-tune our model for permutation prediction over 25k iterations using the multi-class cross entropy loss. These hyper-parameters and implementation details are used throughout the experiments in this chapter.

As performance metrics for the permutation prediction task, we use Kendall-tau and Hamming similarity. Kendall tau is defined as $KT = \frac{c^+ - c^-}{0.5l(l-1)}$, where c^+ and c^- denote the number of all pairs in the sequence that are correctly and incorrectly ordered, respectively. It captures how close we are to the perfect rank. The Hamming similarity measures the number of equal entries in two vectors or matrices normalized by the total number of elements. It indicates how similar our prediction is to the ground truth permutation matrix. In addition, we measure the averaged ℓ_1 normalization error of rows and columns of the predicted doubly stochastic matrices.

We train a CNN model for each attribute in the Public Figures dataset by sampling 30K ordered image sequences from the training images. We then evaluate the trained models on 20K image sequences generated from the test set by sampling correctly ordered sequences and randomly permuting them. We averaged the results over the 11 attributes and repeat the experiment for image sequences of length 4 , 6 and 8. Figure 3.8 presents the results for our proposed methods and the naïve approach.

We observe the naïve approach works well for small sequences and is able to learn the normalization by itself. As the sequence length increases, however, the performance of the naïve approach degenerates and the ℓ_1 normalization error increases. On the other hand, the Sinkhorn Normalization and Bi-level optimization approaches reach better results in both *Kendall-Tau* and Hamming similarity while keeping the normalization error almost unchangeable even for longer sequences. This fact suggests that exploring the geometrical structure of the space of doubly-stochastic matrices (and thereby the permutation matrices) is useful.

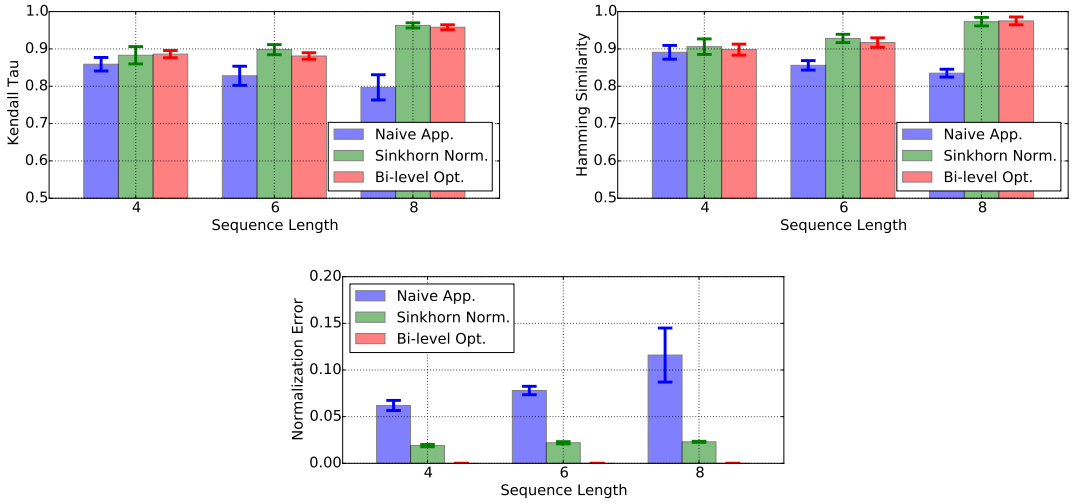


Figure 3.8: Evaluating and comparing naive approach, Sinkhorn normalization and bi-level optimization variants of the proposed model on the permutation prediction task using the Public Figures Dataset [Parikh and Grauman, 2011]. The models are trained and tested for each attribute separately. We report the mean and standard deviation of the the performance metrics (Kendall Tau, Hamming similarity, and normalization error) across the attributes.

It is worth noting that we could train our model for all attributes jointly by sharing the convolution layers and adding as many fully connected layers as the number of attributes. Such an approach is well known in multi-task CNNs [Abdulnabi et al., 2015] and usually provides more generalizable models. However, this approach requires more memory resources which would slow down our experiments.

3.4.2 Relative Attributes

In this experiment, we use DeepPermNet to compare images in a given sequence according to a certain attribute by predicting permutations and applying their inverse. This procedure can be used to solve the relative attributes task, the goal of which is to compare pairs or sets of images according to the "strength" of a given attribute. In this context, we compare our proposed approach to state-of-the-art methods for relative attributes.

For this application, we use the OSR scene dataset [Parikh and Grauman, 2011], the Public Figures Dataset [Parikh and Grauman, 2011], and the implementation details and hyper-parameters described in the previous section. We train our model for each attribute with 30k ordered image sequences of length 8 generated from the training set. Then, we report our models performance in terms of pairwise accuracy measured on the predicted ordering for 20k image sequences of length 8 generated from the test set using stratified sampling.

Different from the existing methods [Souri et al., 2016; Singh and Lee, 2016] which also use deep features and pre-trained models, we directly predict the order of sequences of images instead of pairs. Our scheme allows us to make use of the struc-

Table 3.1: Evaluating the proposed model on the Public Figures Dataset. We report the pairwise accuracy as well as its mean across the attributes.

Method	Lips	Eyebrows	Chubby	Male	Eyes	Nose	Face	Smiling	Forehead	White	Young	Mean
Parikh and Grauman [2011]	79.17	79.87	76.27	81.80	81.67	77.40	82.33	79.90	87.60	76.97	83.20	80.56
Li et al. [2012]	81.87	81.84	79.97	85.33	83.15	80.43	86.31	83.36	88.83	82.59	84.41	83.37
Yu and Grauman [2014]	90.43	89.83	87.37	91.77	91.40	89.07	86.70	87.00	94.00	87.43	91.87	89.72
Souri et al. [2016]	93.62	94.53	92.32	95.50	93.19	94.24	94.76	95.36	97.28	94.60	94.33	94.52
DeepPermNet (Sinkhorn Norm.)	**99.55**	**97.21**	97.66	**99.44**	96.54	**96.21**	99.11	97.88	**99.00**	**97.99**	**99.00**	**98.14**
DeepPermNet (Bi-level Opt.)	99.53	96.65	**98.54**	98.99	**97.21**	94.72	**99.44**	**98.55**	98.77	95.66	98.77	97.89

Table 3.2: Evaluating the proposed model on the OSR dataset. We report the pairwise accuracy as well as its mean across the attributes.

Method	Depth-Close	Diagonal-Plane	Natural	Open	Perspective	Size-Large	Mean
Parikh and Grauman [2011]	87.53	86.5	95.03	90.77	86.73	86.23	88.80
Li et al. [2012]	89.54	89.34	95.24	92.39	87.58	88.34	90.41
Yu and Grauman [2014]	90.47	92.43	95.7	94.1	90.43	91.1	92.37
Singh and Lee [2016]	96.1	97.64	98.89	97.2	96.31	95.98	97.02
Souri et al. [2016]	97.65	98.43	**99.4**	97.44	96.88	96.79	97.77
DeepPermNet (Sinkhorn Norm.)	96.09	94.53	97.21	96.65	96.46	98.77	96.62
DeepPermNet (Bi-level Opt.)	97.99	98.21	97.76	97.10	97.21	96.65	97.49
DeepPermNet (Sinkhorn Norm. + VGG16)	96.87	97.99	96.87	**99.79**	**99.82**	**99.55**	**98.48**
DeepPermNet (Bi-level Opt. + VGG16)	**98.12**	**99.92**	98.13	97.78	98.72	97.87	98.42

ture in the sequences as a whole, which is more informative than pairs providing better performance. For a fair comparison to prior methods, we measure our performance by computing the pairwise accuracy for all pairs in each sequence. Tables 3.1 and 3.2 present our results.

On the Public Figures dataset, DeepPermNet outperforms the state-of-the-art models by a margin of 3% in pairwise accuracy. It is a substantial margin, consistently observed across all attributes. Note that, we outperform the recent method Souri et al. [2016], which uses a pre-trained VGG16 model that has significantly more modeling capacity than the AlexNet [Krizhevsky et al., 2012] architecture that we use. On the other hand, our method works slightly worse than [Souri et al., 2016] on the OSR dataset. We also provide results by building our scheme on a VGG16 model. As is clear, using this variant, we demonstrate even better results outperforming the state-of-the-art methods. In addition, the bi-level variant of our model, despite providing optimal solution to the doubly stochastic approximation, works on par with the Sinkhorn layer, which shows that the Sinkhorn operator is a sufficient approximation for our problem.

It is worth noting that DeepPermNet works better when we use longer sequences for training, because they provide rich information that can be directly used in our method. For instance, the performance of DeepPermNet drops 7% in terms of average pairwise accuracy on the Public Figures dataset when we train our model using just pairs. In addition, the proposed model is not able to explicitly handle equality cases, since the permutation learning formulation assumes each permutation is unique, which is not true in the relative attributes task. Perhaps, this is the reason for the difference in performance between the Public Figures and OSR datasets. Nonetheless, DeepPermNet is able to learn very good attribute rankers from data as shown in our experiments.

We also compute the saliency maps of different attributes using the method proposed by Simonyan et al. [2014]. More specifically, we take the derivative of the esti-

mated permutation matrix with respect to the input, given a set of images. We perform max pooling across channels to generate the saliency maps. Figure 3.9 presents qualitative results and saliency maps generated by DeepPermNet for different attributes.

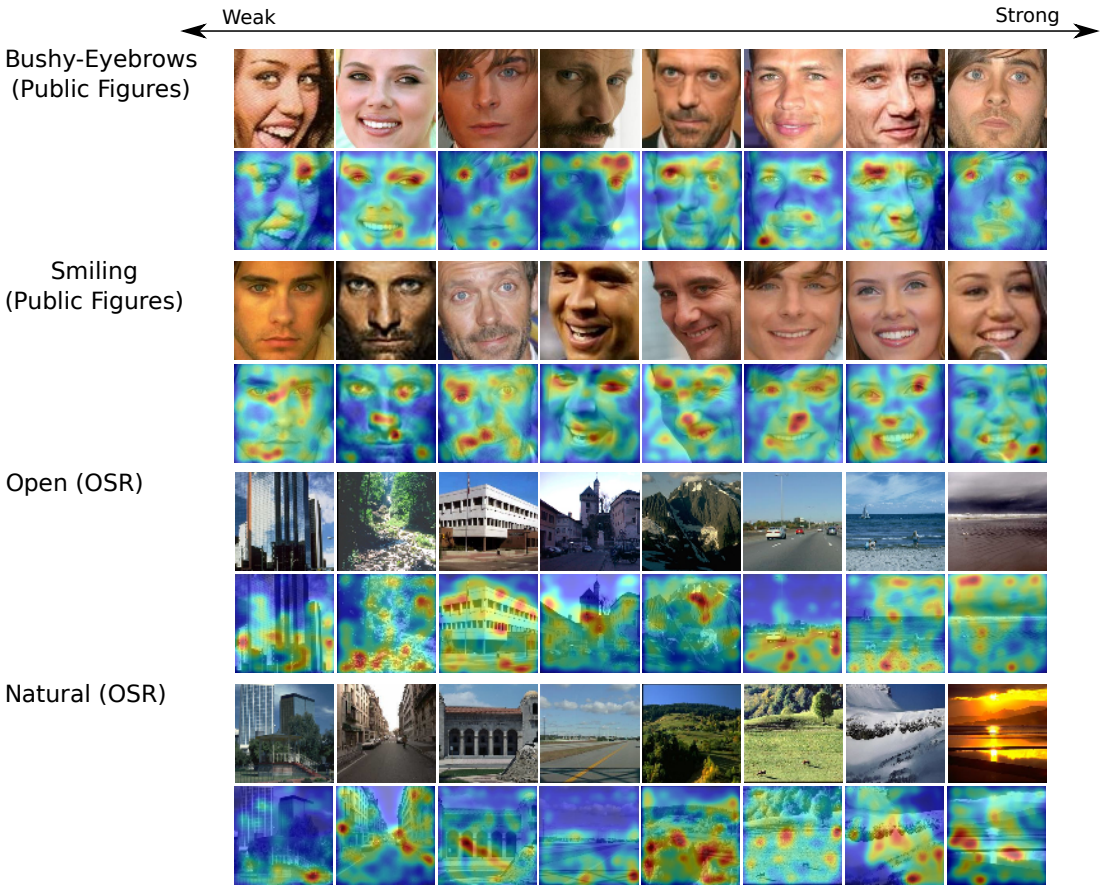


Figure 3.9: Qualitative results: Samples from the Public Figures and OSR test images are ordered according to different attributes. Saliency maps: Smoothed visualization of the derivative of the estimated permutation matrix w.r.t the input images. Regions with warmer color are more relevant to the predicted permutation for the specified attribute. Better viewed in color.

This map is a simplified way to visualize which pixels, regions, and features of a given image are more relevant to the respective permutation predicted by our method. For instance, the attribute “bushy eyebrows" is sensitive to the region of eyes, while the attribute “smiling” is more sensitive to the mouth region. An interesting observation is the possibility of localizing such features without any explicit supervision (e.g., bounding boxes), which could be used for unsupervised attribute localization.

Table 3.3: Evaluating the proposed model on ranking scenes according how interesting they look and ranking cars according to their manufacturing date. We report normalized discounted cumulative gain (NDCG), Kendall Tau (KT), and pairwise accuracy.

	Scene Interestigness			**Car Chronology**		
Method	NDCG	KT	Pair. Acc.	NDCG	KT	Pair. Acc.
Joachims [2006]	0.870	0.317	65.8	0.928	0.482	74.1
Xu and Li [2007]	0.745	-0.077	46.1	0.827	0.118	55.9
Wu et al. [2010]	0.860	0.315	64.3	0.935	0.409	70.6
Cao et al. [2007]	0.821	0.118	55.9	0.872	0.291	64.5
Xia et al. [2008]	0.862	0.282	64.1	0.854	0.278	63.9
Fernando et al. [2015a]	0.887	0.347	67.4	0.949	0.553	76.9
Ours (Sinkhorn Norm.)	0.922	0.360	68.0	**0.968**	**0.724**	**86.2**
Ours (Bi-level Opt.)	**0.923**	**0.363**	**68.2**	0.964	0.700	84.9

3.4.3 Supervised Learning to Rank

We select two supervised image ranking applications to compare our method with other supervised learning-to-rank algorithms, namely, ranking images based on interestingness and ordering car images by manufacturing date. For the former, we use the annotations provided by [Gygli et al., 2013] which assign an interestingness score for images of the OSR dataset. For the latter, we use the car dataset [Lee et al., 2013] which is composed by images of cars manufactured from 1920 to 1999. As implementation details, we use the same model hyper-parameters described in Section 3.4.1.

In this experiment, we train our model by sampling 30k sequences of length four from the training images, and use our learned model and the procedure described in Algorithm 3.1 to rank 20k sequences of length 20 sampled from the test images. Note that the test sequences are longer than the training sequence in this experiment. The final rank obtained is evaluated with rank metrics like NDCG (Normalized Discounted Cumulative Gain), Kendall-tau and Pairwise accuracy. Table 3.3 presents the results.

We observe that our method improves the accuracy of the ranking consistently for all evaluation criteria. It is worth pointing out that the proposed model work drastically better than other neural network models such as ListNet [Cao et al., 2007]. We argue that this improvement is caused by the image representation implicitly learned in a end-to-end fashion by our method. We again observe that the Sinkhorn normalization presents results as good as the exact solution provided by the bi-level optimization variant.

3.5 Chapter Summary

In this chapter, we tackled the problem of learning the structure of visual data by introducing the task of visual permutation learning. We formulated an optimization

problem for this task with the goal of recovering the permutation matrix responsible for generating a given randomly shuffled image sequence based on a pre-defined visual criteria. We proposed novel CNN layers that can convert standard CNN predictions to doubly-stochastic approximations of permutation matrices using Sinkhorn normalizations and bi-level optimization. Thus, the proposed CNN model can be trained in an end-to-end manner.

Through a variety of experiments, we assess the proposed method and demonstrate that permutation learning can be applied to different tasks. More specifically, we first validate the hypothesis of exploring the geometrical structure of doubly-stochastic matrices helps to learn visual permutations. As shown in Figure 3.8, both variants of the proposed DeepPermNet outperform the naïve approach. We then continued our evaluation for real-world applications and state-of-the-art methods such as relative attributes (Section 3.4.2) and supervised learning to rank (Section 3.4.3). In Chapter 4, we extend the pool of applications by testing the proposed approach on self-supervised representation learning. In all experiments, we present state-of-the-art results demonstrating the usefulness of the proposed permutation learning schema.

It is important to highlight the advantages and disadvantages of our two variants of the proposed approach. The bi-level optimization variant optimally solves the doubly-stochastic approximation problem, while the Sinkhorn normalization variant is an efficient and approximate solution for such a problem. However, in practice the Sinkhorn variant works slightly better than the bi-level variant in most of the cases which, perhaps, is a consequence of the quality of image representations learned as evidenced in Chapter 4. Even so, the bi-level variant is able to provide improvements in some cases, e.g, four attributes in Pubfig (Table 3.1), two attributes in OSR (Table 3.2), and Scene Interestingness (Table 3.3). Moreover, the bi-level variant can explore different norms which may further improve the results. However, it comes at the cost of solving an optimization problem for every input during training and inference.

Chapter 4

Learning Image Representations by Permuting Image Regions

"Computer science is an empirical discipline. [...] Each new machine that is built is an experiment. Actually constructing the machine poses a question to nature; and we listen for the answer by observing the machine in operation and analyzing it by all analytical and measurement means available."

Allen Newell and Herbert Simon, 1975

In the previous chapter, we leveraged the geometry on the output space to learn very accurate image rankers with the proposed visual permutation learning framework. Motivated by the generality of the proposed framework, the current chapter shows how to use the spatial structure and other visual priors existent in image data as self-supervision to train such a framework. We demonstrate that this form of supervision, which does not require a single human annotator, encourages the learning of transferable features for object recognition tasks such as object classification, detection and segmentation.

Visual data encompasses rich spatial (and temporal) structure, which is often useful for solving a variety of problems. For instance, surrounding background usually offers strong cues for object recognition, sky and ground usually appear at predictable locations in an image, and objects are made up of known parts at familiar relative locations. Such structural information within visual data has been used to solve several problems, such as object detection and semantic segmentation [Mottaghi et al., 2014; Saxena et al., 2009; Marszalek et al., 2009].

Following these ideas, consider the task shown in Figure 4.1. Here we ask the question "*given shuffled image patches like a jigsaw puzzle, can we recover the original image?*". Although this is a difficult task (even for a human), it becomes straightforward once we identify the object in the patches (e.g., a cat), and arrange the patches for the recognized object, thereby recovering the original image and solving the jigsaw. Therefore, we hypothesize that a machine learning model in order to do well on this

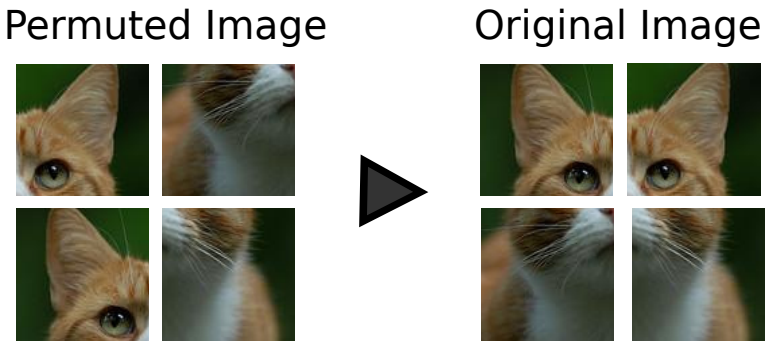


Figure 4.1: Illustration of the self-supervised pretext task. The goal is to learn visual features and solve image jigsaws jointly.

task also needs to understand scenes and objects, i.e., they need to implicitly build a good visual representation that extract objects and their parts in order to reason about their spatial location. As discussed before, this knowledge is also very useful for other object recognition tasks.

Furthermore, these jigsaws can be generated cheaply and in abundance from natural images. The problem of recovering the original image from shuffled ones can be cast in an unsupervised learning setting. Here the recovery task does not require any human annotations (and is thus unbiased given sufficient data [Torralba and Efros, 2011]). Instead it uses the spatial structure as a supervisory signal. Such a learning task is commonly known as self-supervised learning [Doersch et al., 2015; Fernando et al., 2017; Misra et al., 2016; Noroozi and Favaro, 2016; Noroozi et al., 2017; Gidaris et al., 2018], and is very useful to learn rich features, especially in the context of training deep learning models, which often require large amounts of annotated datasets.

In this self-supervised context, Doersch et al. [2015] show that the spatial layout of objects is a strong supervisory signal to learn transferable image representations, while others [Noroozi and Favaro, 2016; Misra et al., 2016; Lee et al., 2017] cast the problem of recovering the original image from shuffled ones as the prediction of a subset of permutations of image regions. More specifically, Misra et al. [2016] model the problem via binary classification and learn to discriminate between correct and incorrect permutations of a sequence. Noroozi and Favaro [2016] learn a multi-class classifier to distinguish between a few prototype permutations selected by a clustering procedure. Similarly, Lee et al. [2017] formulate a multi-class problem on pairwise features. However, these approaches fail to consider structural information beyond a small subset of jigsaws, since enumerating all possible permutations of image regions for a big collection of images is prohibitive.

In the same fashion as image ranking problems in the previous chapter, this task essentially involve learning a function that can recover the order. Therefore, in this chapter, we reformulate the "unshuffling" problem by encoding these jigsaws as sequences of image patches and solve them by predicting the permutation that recovers the original sequence using the proposed visual permutation learning framework.

Different from previous works, this approach allows us to explore the entire structure of natural image, all possible permutations of the image regions, and more complicated jigsaws using finer shuffling grids (e.g., 4×4, 5×5, and so on). In summary, this chapter contributes to our thesis by extending the proposed visual permutation learning framework to leverage the structure and visual priors of natural images in order to learn self-supervised image representations. We test the learned representations on object recognition tasks such as object classification, detection and segmentation on the Pascal VOC dataset [Everingham et al., 2007, 2012].

4.1 Self-Supervised Image Representation Learning

Due to the emergence and success of data hungry machine learning models like deep neural networks, the self-supervised learning paradigm has attracted a lot of attention from the computer vision community recently. The main idea of self-supervision is to exploit supervisory signals, intrinsically in the data, to guide the learning process. In this learning paradigm, a model is trained on an auxiliary task that provides an intermediate representation that can be used as generic features in other tasks. In deep learning, these approaches are well-suited as a pre-training procedure in situations when there is not sufficient data to support fully supervised learning [Girshick et al., 2014; Long et al., 2015]. In this section, we review image representation learning techniques focusing on the methods that follows such a learning paradigm.

The main objective of self-supervised representation learning methods is to learn visual representations without human supervision, and they differ greatly in the proposed pretext task and supervisory signal. For example, Doersch et al. [2015] use spatial co-location of patches in images, Wang and Gupta [2015] use object tracking in videos to provide similar representations for corresponding objects, Fernando et al. [2017] use odd-one-out question answering, Pathak et al. [2016] explore image context to recover missing parts in an image, Pathak et al. [2017] exploit low-level motion-based grouping cues, Noroozi et al. [2017] propose to count visual primitives in images, and Gidaris et al. [2018] explore geometric transformations of images like 2d rotations. In contrast, our proposed method is generic and can be used to solve a broader set of problems.

On the other hand, there are pretext tasks that can be useful themselves. Isola et al. [2016] learn to group visual entities based on their frequency of co-occurrence in space and time. Zhang et al. [2016] propose a model to provide plausible color versions for grayscale images. Donahue et al. [2017] build a generative model for natural images. Note, however, that these methods are highly engineered for their training task and they can not be easily extended to deal with other applications. On the other hand, our method is a general framework able to solve different problems.

A recent work closely related to ours is Noroozi and Favaro [2016] that also proposes to train CNNs for solving image-based jigsaw puzzles. However, different from us, they train a CNN to predict only a tiny subset of possible permutations generated from an image shuffling grid of size 3×3 (specifically, they use only 100

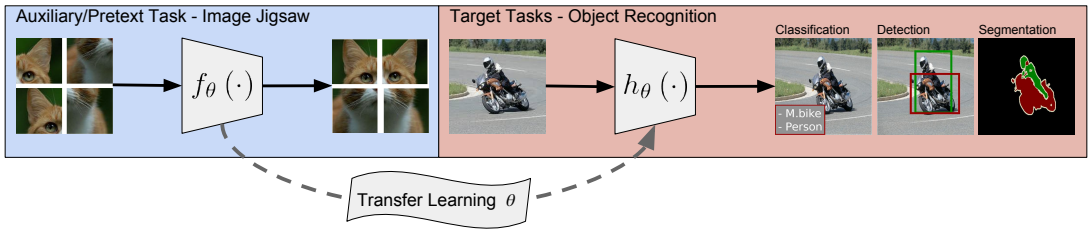


Figure 4.2: The proposed self-supervised representation learning approach. We first train a given deep learning model to solve image jigsaws and then we transfer the knowledge acquired for deep learning based object recognition models.

permutations from the set of 362k possible permutations). Lee et al. [2017] propose similar schema to order sequences of frames in videos. Our method, instead, can handle the full set of permutations and is scalable to finer shuffling grids (e.g., 4×4, 5×5, and so on). In addition, our scheme is generic and can also be used to solve different kinds of learning-to-rank problems.

4.2 Approach

In this section, we describe our method for learning visual representations without human supervision. We start by describing in detail the self-supervised image representation learning pipeline. Then, we describe the proposed pretext task and our solution based on the visual permutation learning framework described in Chapter 3. We finish this section by discussing implementation details important to learn meaningful visual features.

4.2.1 The Self-Supervised Paradigm

As discussed above, the self-supervised learning paradigm aims to learn machine learning models and visual representations without human supervision. In contrast to other unsupervised paradigms, these methods first train machine learning models in auxiliary tasks whose the labels can be easily obtained without human supervision and then transfer the knowledge acquired to a fully supervised target task which is the final goal of the system. These auxiliary tasks are usually motivated by prior knowledge and regularities intrinsically existent in the visual world that provides important clues to solve the target task. For instance, the colorization of objects in a collection of images requires to understanding geometric transformations and object deformation patterns which is also essential to build accurate object tracking systems [Vondrick et al., 2018], the prediction of the relative position of object parts also requires to recognize the object identity which is a common goal with object recognition systems [Doersch et al., 2015], and recognizing temporal coherent videos demands to learn motion dynamics which is also an important clue for action recognition [Misra et al., 2016].

In order to formalize these ideas, let us consider the auxiliary task S and the final task T whose ideal solutions explores similar clues and knowledge. Following

a machine learning approach for these tasks, let us also define two data sets $\mathcal{D}_S$ and $\mathcal{D}_T$ and two models f and h for the tasks S and T, respectively. The self-supervised learning approach can be summarized as first learning the model f for the auxiliary task S using the dataset $\mathcal{D}_S$ and then transfer this knowledge to the model h which is adapted for the task T using the dataset $\mathcal{D}_T$. Another essential requirement is that the data D_S is composed of a collection of visual inputs X (e.g., , videos and images) and their artificial labels Y which are generated from X itself using some hand-crafted procedure. In order for this approach be useful for real world applications, such a procedure has to be computationally cheap and generate artificial labels in abundance. In other words, it should provide a large dataset D_S allowing the dataset $\mathcal{D}_T$ for the target task T, which is usually human annotated, to be much smaller than the nowadays large scale datasets like ImageNet [Russakovsky et al., 2015] and MS-COCO [Lin et al., 2014].

In the context of deep learning models, self-supervised learning is a very useful approach to alleviate the needs for large scale human annotated dataset that has hampered the application of these models in the real world. Furthermore, deep models like neural networks and other parametrized approaches are well-suited for this learning paradigm since the knowledge acquired by these models can be easily transferred across models and tasks without complicated procedures [Yosinski et al., 2014]. Therefore, we propose to learn image representations for object recognition using deep learning models and the self-supervised paradigm. As shown in Figure 4.2, we train a given deep learning model f to solve image jigsaws like the one depicted in Figure 4.1. In addition to solve such a task, we also learn image features transferable to object recognition tasks like object classification, detection and segmentation. Then, we transfer the leaned features to the target model h by just undergoing small modifications and finetuning in a relatively small dataset.

4.2.2 Image Jigsaws And Visual Permutation Learning

Having our self-supervised approach described, we now focus on the explanation of the proposed auxiliary task and solver. Our goal is to learn image representations useful for object recognition tasks using the self-supervised paradigm. Towards this end, we first observe that objects are collections of salient parts which has certain spatial configurations or structure. Then, we explore this structural information and define an auxiliary task to train a deep learning model which has to learn such a information in order to perform well in this task. More specifically, we create a pretext task where the objective is to recover the object-parts configuration of an image given its artificially shuffled version. Note that in order to accurately solve such a task, the learner needs to learn what are objects, what are their parts and how those parts fit together. This knowledge is useful to discriminate between object categories and background which is also essential for object recognition tasks such as object detection, classification and segmentation.

Figure 4.3 formalizes the proposed pretext task. Let us define an image X as an ordered set of image patches $\{I_{1,1}, \ldots, I_{1,g}, I_{2,1}, \ldots, I_{2,g}, \ldots, I_{g,g}\}$ obtained by laying a

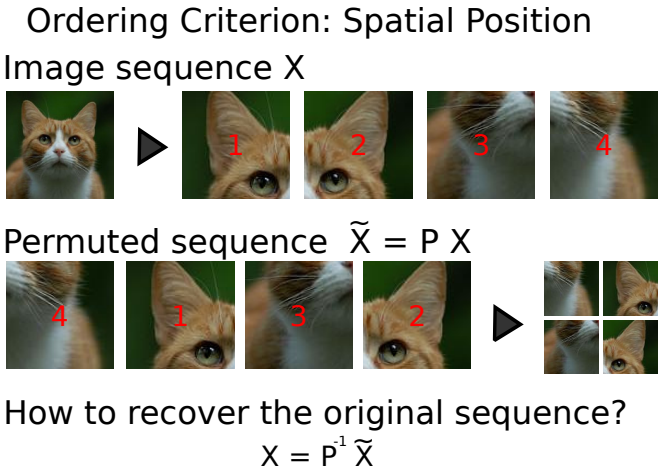


Figure 4.3: The proposed pretext task for image representations learning. Note that the proposed pretext task is an instance of the visual permutation learning problem (discussed in Chapter 3) where the input sequences are sequences of image regions and the ordering criterion is given by the spatial structure of objects in images.

$g \times g$ grid on top of an image and extracting random crops $I_{h,w}$ of size $c \times c$ pixels within each grid cell $h, w \in [1, g[$. In addition, let us define the artificially shuffled version of X as $\tilde{X}$ where the order of the patches $I_{h,w}$ were permuted by a random generated permutation matrix $P \in \{0,1\}^{g^2 \times g^2}$. Thus, our pretext task is in fact to predict the matrix P from a given shuffled collection of image patches $\tilde{X}$ such that $P^{-1} = P^T$ recovers the original image X. That is, the proposed pretext task is an instance of the visual permutation learning problem (discussed in Chapter 3) where the input sequences are sequences of image regions and the ordering criterion is given by the spatial structure of objects in images.

Therefore, we follow the proposed visual permutation learning framework and solve the proposed pretext task by learning a parametrized function $f_\theta(\cdot)$ that predicts the permutation matrix P which generated the shuffled image $\tilde{X}$ from a given input image X. Furthermore, we implement the function $f_\theta(\cdot)$ as the DeepPermNet model described in Section 3.3.3 in order to jointly learn transferable image representations. Such a model is well-suited to the problem since the learned representation can be easily transferred to the target tasks by performing small modifications in the network architecture and finetuning in a small dataset. Differently from existing self-supervised representation learning techniques which simplifies this kind of pretext task by just learning to discriminate subsets of possible permutations [Doersch, 2016; Misra et al., 2016; Lee et al., 2017], our approach can handle the full set of permutations and is scalable to finer shuffling grids (e.g., 4×4, 5×5, and so on). In addition, our scheme is generic and can also be used to solve different kinds of learning-to-rank problems as discussed in Section 3.4.

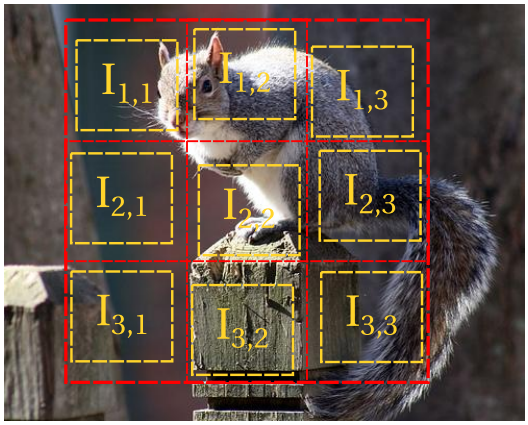


Figure 4.4: Image region sampling procedure used to avoid "shortcuts" when creating image jigsaws.

4.2.3 Avoiding "shortcuts"

It has been observed in the literature that self-supervised learning methods can exploit "shortcuts" involving information useful for solving the pretext task but not for a target task [Doersch et al., 2015; Noroozi and Favaro, 2016]. For instance, chromatic aberration and edge continuity are good cues for solving the visual permutation task, but are not useful for generic object detection or image classification. In order to avoid these "shortcuts", we follow image preprocessing procedures described by Doersch et al. [2015]. We first resize the images having the smallest side equal to 256 pixels. Then, we randomly crop a squared region of the image and resize to 225×225 pixels. Then we split the resized crop into a 3×3 grid cell, each with 75×75 pixels. Finally, we randomly select 64×64 pixels tiles from each cell and train our model as described above. Figure 4.4 presents an example of this preprocessing procedure. This allows us to have an 11 pixel gap between tiles. Noroozi and Favaro [2016] show improvements in the target task by using additional procedures such as augmenting the data with gray-scale images, jittering the color channels, and increasing the gap between sampled tiles. We did not investigate these additional procedures, but they can be easily added to our framework.

4.3 Transfer Learning Experiments

In this section, we evaluate the proposed model for self-supervised image representation learning. Following the literature on self-supervised pre-training [Doersch et al., 2015; Donahue et al., 2017; Pathak et al., 2016; Noroozi and Favaro, 2016; Larsson et al., 2017; Ren and Jae Lee, 2018], we test our models on the commonly used self-supervised benchmarks on the PASCAL Visual Object Challenge and compare against supervised and self-supervised procedures for pre-training. We first train our model on the proposed pretext task using the train split of the ImageNet dataset

Table 4.1: Classification and detection results on PASCAL VOC 2007 test set under the standard mean average precision (mAP), and segmentation results on the PASCAL VOC 2012 validation set under mean intersection over union (mIU) metric. *Noroozi and Favaro [2016] and our methods use a more computationally intensive ConvNet architecture with a finer stride at conv1 during the self-supervised training, but we use standard Alex-net architecture when finetune in the target task allowing a fair comparison with all competing methods.

Pre-training Method	Pretext task	Cls.	Det.	Seg.
ImageNet	Supervised	78.2	56.8	48.0
Random Gaussian	None	53.3	43.4	19.8
Agrawal et al. [2015]	Estimating Ego-motion	52.9	41.8	-
Doersch et al. [2015]	Context Prediction	55.3	46.6	-
Wang and Gupta [2015]	Visual tracking	58.4	44.0	-
Pathak et al. [2016]	Context autoencoder	56.5	44.5	29.7
Donahue et al. [2017]	Adversarial Learning	58.9	45.7	34.9
Zhang et al. [2016]	Image colorization	65.6	47.9	35.6
Noroozi and Favaro [2016]*	Image jigsaws	67.6	53.2	37.6
Owens et al. [2016]	Ambient sounds	61.3	44.0	-
Bojanowski and Joulin [2017]	Alignment with noisy targets	65.3	49.4	-
Noroozi et al. [2017]	Counting visual primitives	67.7	51.4	36.6
Lee et al. [2017]	Sorting sequences	63.8	46.9	-
Pathak et al. [2017]	Motion-based segmentation	61.0	52.2	-
Zhang et al. [2017b]	Cross-channel prediction	67.1	46.7	36.0
Larsson et al. [2017]	Image colorization	65.9	-	38.0
Jenni and Favaro [2018]	Predicting synthetic artifacts	69.8	52.5	38.1
Gidaris et al. [2018]	Predicting image rotation	**72.97**	54.4	39.1
Kim et al. [2018]	Damaged image jigsaws	69.2	52.4	39.3
Nathan Mundhenk et al. [2018]	Improved context prediction	69.6	**55.8**	**41.2**
Ren and Jae Lee [2018]	Multi-task	68.0	52.6	-
DeepPermNet (Sinkhorn Norm.)*	Visual Permutation Learning	69.4	49.5	37.9
DeepPermNet (Bi-level Opt.)*	Visual Permutation Learning	65.5	45.7	36.4

[Krizhevsky et al., 2012] as training set discarding its labels. Then, we transfer our learned weights to standard deep learning based recognition models for object classification, detection and segmentation which are finetuned and tested in the PASCAL Visual Object Challenge datasets. As evaluation metrics, we report the mean average precision (mAP) on PASCAL VOC 2007 [Everingham et al., 2007] for object classification and detection, while we report mean average intersection over union (mIU) on PASCAL VOC 2012 [Everingham et al., 2012] for object segmentation.

It is important to emphasize that we do not use any pre-trained models or human annotated labels when training our model in the pretext task. Instead, we train our CNN models from scratch using random initialization and self-supervised labels. The model is trained for 400k iterations using an initial learning rate of 0.001, which is dropped by one-tenth every 100k iterations. We use batches of 256 sequences each of 64×64 image patches. In order to evaluate how well the proposed models

can solve such a task, we use 50k images on the ImageNet validation set and apply random permutations using the 3×3 grid layout. In this self-supervised setting, DeepPermNet reaches a score of 0.72 on the Kendall-tau metric. After validating the self-supervised training, we transfer the learned weights to initialize from *Conv1* to *Conv5* layers of AlexNet [Krizhevsky et al., 2012], Fast-RCNN [Girshick, 2015] and Fully Convolutional Network [Long et al., 2015] models and fine-tune them for object classification, detection, and segmentation tasks respectively, using their default training parameters. In order to make the competing methods directly comparable, we use stride 2 in the first layer of our network during the training of visual permutation learning task, while we use a standard AlexNet (stride 4 on the first layer) when finetune the recognition models o the target tasks. Table 4.1 presents our results.

We observe that the self-supervised methods are still behind the supervised approach, but this performance gap is gradually reducing as self-supervised methods improve. Our DeepPermNet works as well as most of the self-supervised competitors. For instance, the tested classification, detection and segmentation neural network based models improve their performance in about 16%, 6%, and 18%, respectively, when pretrained in our self-supervised framework. While our proposal is marginally surpassed by very recent approaches, it outperforms its direct competitor [Noroozi and Favaro, 2016] in object classification and segmentation by exploiting our permutation prediction schema. In addition, DeepPermNet is a more generic method than the method proposed by Noroozi and Favaro [2016], since our method can be used to solve many different computer vision tasks as shown in our previous experiments. We also notice that the bi-level approach performs slightly worse than the Sinkhorn normalization approach in this self-supervised experiment. Perhaps, the reason for that is computation of the gradient which requires inverting a matrix, and can cause numerical issues.

Interestingly, when finer grid schemes are used (e.g., 4×4), we do not observe any improvement in the target tasks. This agrees with the ablation study presented in [Noroozi and Favaro, 2016], which shows that the performance in the target task increases with the total number of permutations, but decreases with the increasing of the similarity between these permutations in their jigsaw task. Therefore, we believe when we deal with all possible permutations and increase the grid partition, we end up increasing the general similarity between the target permutations, which is prejudicial for transfer learning. Perhaps, a solution to this issue is to weight the permutations according to their average similarity to the other permutations, which is a compelling direction for future work.

4.4 Chapter Summary

While Chapter 3 presents the visual permutation learning framework as an approach to learn structural information existent in data by exploring the structure of visual permutations, the current chapter applies such a framework to learn unsupervised image representations by exploring visual priors and regularities in the input visual data. Towards this end, we propose a self-supervised task resembling image jigsaws and show that a model trained to solve this task also learns image representations transferable to object recognition tasks such as object classification, detection and segmentation. In the context of deep learning models, this approach is very useful since it can mitigate the need for large scale human annotated datasets which has hampered the application of these models in complex problems.

Chapter 5

Compositional Algebra of Classifiers

"The moving power of mathematical invention is not reasoning, but imagination."

Augustus De Morgan, 1831

Chapters 3 and 4 exploited structural information in visual outputs and inputs in order to produce more accurate image rankers and unsupervised learned image representations, respectively. On the other hand, the current chapter leverages the structure in the model space to propose a compostional learning framework that resembles an algebra of visual classifiers. Aiming to overcome the closed world assumption made by fully supervised methods for visual recognition, the proposed framework can compose classifiers for new visual concepts without a single training data of these concepts.

In order to start our discussion, imagine a sea-faring bird with "hooked beak" and "large wingspan". Most people would be thinking of an albatross. Moreover, given a set of images of birds, the descriptive features "hooked beak" and "large wingspan" are key for someone to identify images of albatross versus other birds even if they had never seen an albatross before. These provide evidence that visual concepts are compositional and complex visual concepts like albatross are defined as a composition of simpler visual concepts such as "hooked beak" and "large wingspan". In addition, humans have very formal and structured ways of reasoning about compositions such as propositional logic, predicate logic, and boolean algebra. However, the current state-of-the-art models for recognition follow a laborious data-driven approach, where complex concepts are learned using thousands or millions of manually labeled examples instead of using composition. Such data greedy approach is unfeasible for many real world applications.

In this chapter, we build on the insight that visual concepts are fundamentally compositional and develop an algebra for combining concept classifiers. Towards this end, we propose a composition framework inspired by boolean algebra structures such as disjunction, conjunction, and negation. More specifically, we develop neural network modules which can learn to compose classifiers according these logi-

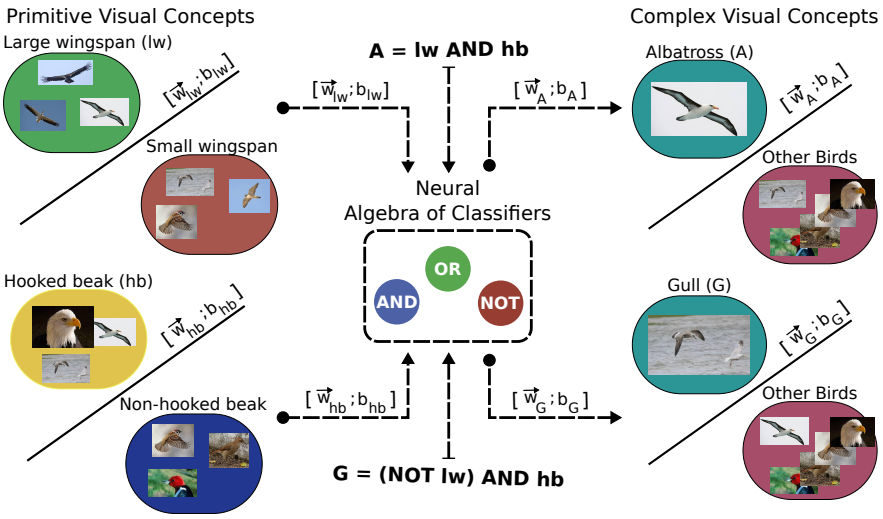


Figure 5.1: Illustration of the proposed neural algebra of classifiers. Given classifiers for primitive visual concepts such as hooked beak and large wingspan, we can compose classifiers for complex concepts such as gull and albatross that are represented by boolean expressions of these primitives.

cal operators allowing us to produce classifiers for any complex concept expressed as a boolean expression of primitive concepts. For instance, our approach can compose a classifier for albatross by combining classifiers for "large wingspan AND hooked beak". Likewise, gull's classifier can be expressed as "(NOT large wingspan) AND hooked beak" (Figure 5.1). Moreover, such a framework can predict unseen complex visual concepts like humans do. For example, it is possible to identify a car made of grass by composing a classifier for "grass AND car", even if such a concept does not have training data. It also allows us to recognize subclasses and specific instances of objects without any additional annotation effort. Therefore, we can scale-up recognition systems for complex and dynamic scenarios.

Learning how to compose classifiers for unseen complex concepts from simple visual primitives by developing a compositional algebra is a challenging task since there is no trivial mapping between primitives and their compositions. Naively, we can think of recognizing an albatross whenever the classifiers for large wingspan and hooked beak fire simultaneously. However, such an approach assumes strong independence between visual primitives and does not consider the imperfection of the primitive classifiers or reason about correlations and cooccurrences of visual primitives. Furthermore, as observed by Misra et al. [2017], the meaning of a composition depends on the context and the particular instance being composed. For instance, the visual appearance of "old" for bikes is completely different for people. In contrast, our approach is learned in the classifier space exploring correlations, cooccurrences, and contextuality between visual primitives in order to compose more accurate classifiers for complex visual concepts.

This chapter's contributions are threefold. First, we propose a learning framework for composition of classifiers. Such a framework resembles an algebra in which

we can synthesize classifiers for any visual concept described as boolean expression of visual primitives. Second, we develop a neural network based model which minimizes the classification error of a subset of visual compositions and generalizes for unseen compositions. Third, we show how these modules can be used recursively to produce classifiers for complex concepts expressed as boolean expressions of visual primitives.

We conduct several experiments to show the efficacy of our approach. We show that our method is able to synthesize classifiers according to simple composition rules by learning how to compose concepts from a subset of primitive compositions and generalizing for compositions not seen during training (Section 5.3.2). In addition, our approach naturally extends to complex compositions by recursively applying our learned neural network modules (Section 5.3.3). On all of these settings, our method outperforms standard baselines. Finally, we evaluate qualitatively some interesting properties of our method (Section 5.3.4).

5.1 Compositionality in Visual Recognition

The principle of compositionality says that the meaning of a complex concept is determined by the meanings of its constituent concepts and the rules used to combine them [Frege, 1948; Boole, 1854; Burnyeat et al., 1990]. For instance, written language is built of symbols which form syllables, words, sentences, and texts. Likewise, visual data can be decomposed into scene, objects, textures and pixels. The principle is pervasive in our world and have been studied extensively by different scientific communities ranging from mathematics to philosophy of language. In this chapter, we study compositionality in the context of visual recognition.

Viewing objects as collections of known parts at familiar relative locations may be the most common way to incorporate compositionality into visual recognition systems. For instance, deformable parts model [Felzenszwalb et al., 2010; Girshick et al., 2011], and-or graphs [Wu and Zhu, 2011; Si and Zhu, 2013; Zhu et al., 2008; Tang et al., 2017], dictionary learning [Tu et al., 2005; Zhu et al., 2010, 2007], and self-supervised representation learning [Doersch et al., 2015; Santa Cruz et al., 2017; Fernando et al., 2017] techniques are built over this intuition. Likewise, scenes can be seen as hierarchical compositions of concepts in different abstraction levels. Then, convolutional neural networks [Zeiler and Fergus, 2014; Simonyan and Zisserman, 2014b] and recurrent neural networks [Hochreiter and Schmidhuber, 1997; Chung et al., 2014; Socher et al., 2011] can also be seen as compositional models. Differently, we focus in composing classifiers for complex concepts that can be expressed as boolean expression of primitive visual concepts. For instance, our approach is able to classify a specific instance given its visual attributes even if such an instance is not present in the training set.

It is important to note that compositionality helps to reduce the complexity of some problems by decomposing them in subproblems which allow more tractable solutions. For instance, Andreas et al. [2016] and Hu et al. [2017] explore the structure

of natural language questions in order to define a set of simpler problems which can be solved by simple neural networks. Neelakantan et al. [2016], proposed a neural network to induce programs of simple operations to answer questions which involve logic and arithmetic reasoning. Faktor and Irani [2012, 2013] use the "similarity by composition" framework [Boiman and Irani, 2007] to perform clustering and object co-segmentation. Likewise, we decompose the problem of recognizing any specific instances of objects by the problem of composing a classifier according to simple rules from its individual visual primitives.

Closely related to our work, Misra et al. [2017] show the importance of context in composition of object and attributes. More specifically, they argue that the visual interpretation of attributes depends on the objects they are coupled with. For instance, an old bike has different visual features than an old computer. Building on this intuition, the authors propose a transformation function to map from object and attribute classifiers to the composition of classifiers. Thus, their scheme can only synthesize classifiers for visual concepts like "red wine", "large tv", and "small modern cellphone". In contrast, we develop a generic framework to combine any number of concept classifiers according to arbitrary boolean expressions. Such a framework provides richer expressiveness since we are able to compose classifiers for more complex concepts like "red or blue socks without holes".

The problem of classifying unseen visual concepts is also known as zero-shot classification [Palatucci et al., 2009; Lampert et al., 2009; Lei Ba et al., 2015; Frome et al., 2013]. However, zero-shot classifiers are only able to recognize unseen object classes, while our proposed framework is also able to recognize unseen groups, subgroups, and specific instances of objects. Furthermore, we do not make assumptions about the existence of an external source of knowledge such as class-attributes relationship [Lampert et al., 2009], text corpus [Lei Ba et al., 2015], or language models [Frome et al., 2013]. We explore compositionality in the visual domain and other visual priors, such as co-occurrence and dependence of visual attributes.

5.2 Neural Algebra Of Classifiers

In this section, we explain the proposed neural algebra of classifiers. We start by formalizing the problem of classifier composition in an algebraic perspective. Then, we describe our learning algorithm, model architecture, and inference pipeline.

5.2.1 Problem Formulation

Our problem consists of classifying images according to complex visual concepts expressed as boolean algebra of a set of primitives. Initially, let us assume we have a set of known visual concepts, named primitives, like socks (S), red (R), blue (B) and holes (H). In addition, consider basic composition rules inspired by boolean operators: ($\wedge$) that identifies whether two primitives are depicted in the image simultaneously, ($\vee$) which denotes if the image has at least one of the primitives, and ($\neg$) which accepts

all images which a primitive is not depicted. Then, what is the classifier for a complex visual concept expressed by multiple compositions of primitives and these rules. For instance, what is the classifier for "red or blue socks without holes" described by the expression "S $\wedge$ (B $\vee$ R) $\wedge$ ($\neg$ H)".

Formally, let us define a set of *primitives* $\mathcal{P} = \{p_i\}_{i=1}^{M}$. We can express complex concepts by forming arbitrary *expressions* recursively combining primitives with *composition rules* $\mathcal{O} = \{\neg, \wedge, \vee\}$. Note that this set of rules is a complete functional set, i.e., any propositional expression of primitives can be written in terms of these rules. Then, our objective can be summarized as learning a parametrized function, $f_\theta(\cdot) : \mathcal{E} \rightarrow \mathcal{C}$ that maps from the space of expressions $\mathcal{E}$ to a space of binary classifiers $\mathcal{C}$. In other words, we want the function $f_\theta(\cdot)$ be able to synthesize a classifier for any given expression.

Without loss of generality, we will explain the details of our approach for the case of linear classifiers, but the same formulation can be used to synthesize non-linear or kernelized classifiers. Thus, we define $f_\theta(\cdot)$ as,

$$\hat{w}_e = f_\theta(e) \tag{5.1}$$

where $\hat{w}_e \in \mathcal{C}$ is a linear classifier, i.e., separating hyperplane, that distinguishes positive and negative samples for an expression and θ are the function parameters.

5.2.2 Learning Objective

In order to efficiently learn the proposed mapping function, we need to represent the visual content of images and the semantic meaning of primitives in a compact way. Towards this end, we define $h_\phi \in \mathbb{R}^D$ as a parametrized feature extractor which computes a vector representation that summarizes all visual features of a given image and ϕ is the set of parameters. Likewise, we represent all primitives by classifiers trained to recognize images that depict them. Since we focus on linear classifiers in this chapter, we represent every primitive p by the separating hyperplane parameters $w_p \in \mathbb{R}^D$, e.g., obtained by training an one-vs-all linear SVM classifier on the feature representation of images.

Note that boolean expressions are evaluated by decomposing them into a sequence of simpler terms and evaluating these terms recursively. For instance, the expression $S \wedge (B \vee R) \wedge (\neg H)$ can be evaluated by recursively evaluating the sequence of simpler expressions $(B \vee R)$, $S \wedge (B \vee R)$, $\neg H$, $((S \wedge (B \vee R)) \wedge (\neg H))$. Such a decomposition can be computed efficiently by representing expressions as binary trees and parsing their nodes in post-order. Then, we propose to model the function $f_\theta(\cdot)$ as a set of composition functions $g^\wedge(\cdot)$, $g^\vee(\cdot)$, $g^\neg(\cdot)$. In other words, the function $f_\theta(\cdot)$ is computed by decomposing an expression in simple terms and applying the composition functions accordingly.

These composition functions $g^* : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$ are auto-regressive models which maps from and to the classifier space. For instance, the conjunctive composition function $g^\wedge(\cdot)$, given two concepts as input like "Socks" and "Red" represented in

the classifier space by $w_s, w_r \in \mathbb{R}^D$, should compute the classifier $w_{s \wedge r} \in \mathbb{R}^D$ that recognizes when both concepts are present in a image simultaneously. Similarly, the functions $g^{\vee}(\cdot)$ and $g^{\neg}(\cdot)$ should compute the disjunction and negation in classifier space, respectively.

We also observe that some of these composition functions can be defined analytically or in terms of other composition functions. More specifically, the negation consists of just inverting the separating hyperplane and the disjunction can be derived using De Morgan's laws. Then, we propose to implement these functions as

$$\begin{aligned} g_{\theta}^{\wedge}(w_a, w_b) &= \text{Neural Network}(w_a, w_b), \\ g^{\neg}(w) &= -w, \\ g^{\vee}(w_a, w_b) &= g^{\neg}\left(g^{\wedge}\left(g^{\neg}(w_a), g^{\neg}(w_b)\right)\right), \end{aligned} \tag{5.2}$$

where the conjunctive composition $g_{\theta}^{\wedge}(\cdot)$ is a neural network learned from data and θ are the learnable parameters.[1] Therefore, the learning of function $f_{\theta}(\cdot)$ is decomposed on the learning of these composition functions.

Following these ideas, let us define a subset of training expressions $\{e_k\}_{j=1}^{K} \subset \mathcal{E}$ composed by composition rules $\mathcal{O}$ and primitives $\mathcal{P}$. Note that such a subset is much smaller than all possible expressions that can be formed by composing these primitives. Likewise, we define a set of training images $\{(x_i, y_i) \mid x_i \in \mathcal{I}, y_i \in \{0,1\}^K\}_{i=1}^{N}$ with the ground-truth label y_{ij} denoting whether the image x_i is a positive example for the expression e_j. Then, learning the function $f_{\theta}(\cdot)$ can be defined as,

$$\underset{\theta,\phi}{\text{minimize}} \quad \frac{1}{KN} \sum_{j=1}^{K} \sum_{i=1}^{N} \quad \alpha_1 \Delta\left(f_{\theta}(e_j)^T h_{\phi}(x_i), y_{ij}\right) + \frac{\alpha_2}{2} \left\| f_{\theta}(e_j) \right\|_2^2 + \alpha_3 R(\theta), \tag{5.3}$$

where $\Delta(\cdot,\cdot)$ is a classification loss function, $R(\cdot)$ is some regularization function and $\{\theta, \phi\}$ is the set of learnable parameters. We also have the hyper-parameters $\alpha \in \mathbb{R}^3$ which controls how our model correctly fit the training data (α_1), regularizes for training expressions (α_2), and for unknown expressions (α_3). The idea is to learn how to synthesize classifiers that correctly classify images according to the input expressions even if the expressions had not been seen during training.

It is important to note that such a formulation aims to explore semantic similarity on classifiers space and the visual compositionality principle in order to make our learning problem easier to solve. We use a relative small subset of expressions to learn our proposed mapping function and rely on the classifier similarity to generalize for unknown expressions. Likewise, we explore visual compositionality by decomposing training expressions in simpler expressions and jointly learning the composition functions.

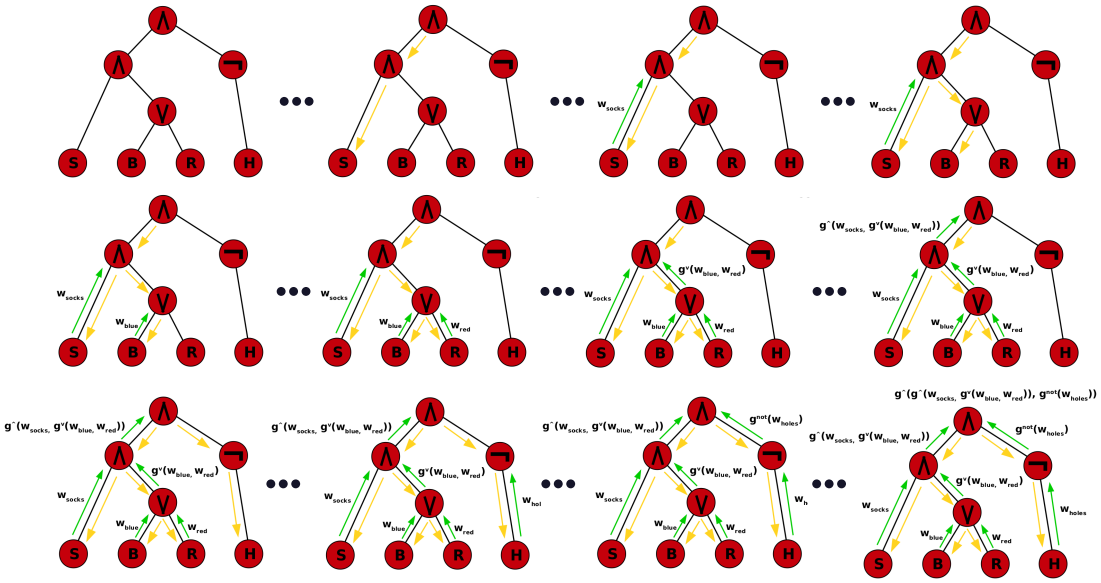


Figure 5.2: Inference steps for the visual concept "red or blue socks without holes" expressed as the boolean expression of primitives "S $\wedge$ (B $\vee$ R) $\wedge$ ($\neg$ H)". The resulting compositional function is: $f_\theta(e) = g^\wedge\left(g^\wedge\left(w_s, g^\vee\left(w_b, w_r\right)\right), g^\neg(w_h)\right)$. The images reads from top to bottom and from left to right.

5.2.3 Inference Algorithm

As alluded to above, our main goal is to produce classifiers for boolean expressions of primitives. These expressions can be represented by a tree where composition rules are nodes and primitives are leaves. Thus, our inference consists of parsing the expression tree in post-order and applying the composition functions accordingly in order to end up with the final classifier just after parsing the root (see the example in Figure 5.2).

Then, we can compute the classifier score for an image given an expression by:

$$s = f_\theta(e)^T h_\phi(x) \tag{5.4}$$

This score reflects the compatibility between the expression and the image. We want this score to be high only if the image contains the complex concept described by the expression e and low otherwise. As an example, for the expression "$S \wedge (B \vee R) \wedge \neg H$" we want the score s to be high only for images containing blue or red socks without holes and want it to be low for images containing any other concept.

5.2.4 Model and Implementation Details

We propose to implement the conjunctive composition function $g^\wedge_\theta(\cdot)$ and the feature extractor $h_\phi(\cdot)$ as a multilayer perceptron (MLP) [Haykin et al., 2009] network and VGG-16 convolutional neural network [Simonyan and Zisserman, 2014b] respectively. We represent images with 4096-dimensional feature vectors computed by the

[1] Equivalently, we could have defined $g^\vee$ by the neural network and $g^\wedge$ using De Morgan's laws.

FC6-layer of the VGG-16 network pretrained on ImageNet [Russakovsky et al., 2015]. Consequently, the primitives are represented by 4097-dimensional vector obtained from training linear SVMs on these features. Since the bias can be implemented by adding a $+1$ fixed feature to image representation vectors, $g_\theta^\wedge(\cdot)$ is a MLP network that have (2×4097) inputs and two fully connected layers with outputs of size (1.5×4097) and (4097), respectively. We use the LeakyReLU non-linearity, with slope set to 0.1, in between the layers and linear activation on the outputs. Figure 5.3 shows our neural network architecture in details.

During training, we approximate the objective Equation 5.3 by batches of 32 expressions, 5 positive and 5 negative images for each expression sampled uniformly. We first train our neural algebra of classifiers module alone during 50 epochs, then we finetune the features jointly during 10 epochs more. Since the primitives are represented by linear SVM classifiers, we decide to use the hinge loss,

$$\Delta\left(s_{ij}, y_{ij}\right) = \max(1 - y_{ij} s_{ij})$$

where s_{ij} is the score assigned to the image x_i by the classifier predicted for the expression e_j. In addition, we use the standard ℓ_2 regularization in the network weights as our regularization function $R(\cdot)$.

5.3 Experiments

We now evaluate the performance of our method and compare against several baselines. We first describe the experimental setup, datasets, metrics, and baselines used in our experiments. Then, we analyze how effectively our model can compose classifiers for simple and arbitrary compositions of concepts in addition to presenting a qualitative evaluation of our method.

5.3.1 Experimental Setup

We are interested in the task of predicting whether a given image contains the complex concept described by a boolean expression of primitives which may not have any training data. Towards this end, we first define two disjoint sets of boolean expressions of primitives named "training expressions" and "test expressions" and three disjoint sets of images named "training images", "validation images" and "test images". Second, we learn the primitive representation, train our model and baselines using training images and training expressions. Then, we evaluate the performance of our method and baselines classifying images on the validation set according to training expressions, named "known expressions performance", and classifying images on the test set according to test expressions, named "unknown expressions performance". The former suggests how well a model learns to compose classifiers and the latter how well a model generalizes for expressions not seen in training.

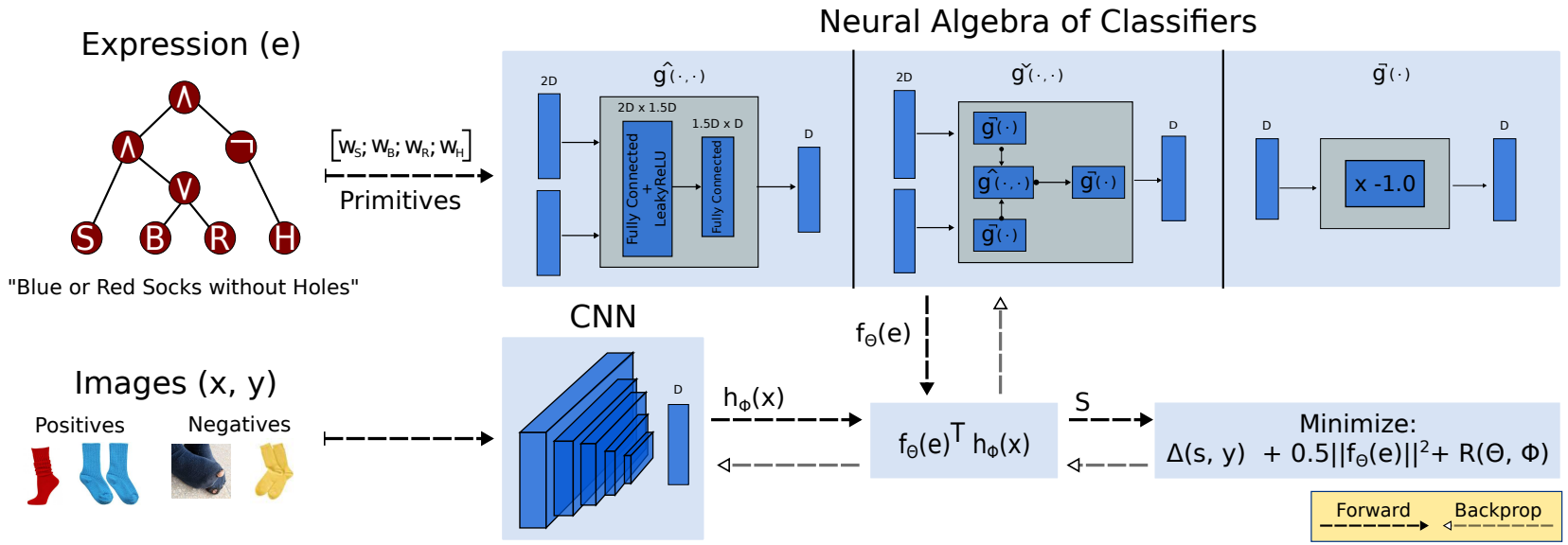


Figure 5.3: Neural algebra of classifiers: Our method composes classifiers for complex visual concepts expressed as boolean expressions of primitive concepts. We first represent every primitive by a classifier and every image by a feature vector. Then, we use a subset of training expressions to learn a set of composition functions which generalizes to concepts represented by unseen expressions or even unseen primitives. In order to make such learning problem easier, we explore geometry and boolean algebra fundamentals such as hyperplanes and De Morgan's laws.

Datasets. We use the CUB-200 Birds (CUB-200) [Wah et al., 2011] and Animal With Attributes 2 (AwA2) [Xian et al., 2017] datasets in our experiments. Since none of these datasets were designed for our purpose, we split these datasets in order to perform controlled experiments. First, we compute all possible binary conjunctive and disjunctive expressions of primitives and filter out the ones that do not have reasonable amount of positive and negative images. Then, we randomly split the images between train, validation, and test images making sure that every expression and primitive have reasonable amounts of positive and negative samples in each image split. As a result, we create approximately $3k$ training expressions and $1k$ test expressions using 250 primitives for the CUB-200 dataset, while we create approximately $1.5k$ training and 600 test expressions using 80 primitives for the AwA2 dataset. In order to make easier to reproduce our results, the experiment code and these data splits are available in the first author's homepage.

Metrics. A boolean expression of primitives defines a binary classification problem where images are classified as relevant or irrelevant for the visual concept described. Therefore, we use well-known evaluation metrics of image retrieval and binary classification. More specifically, we use the mean average precision (MAP), area under the ROC curve (AUC) and equal error rate (EER). We compute these metrics globally in order to take the data imbalance in account since some expressions are naturally rarer than others.

Baselines. We compare our method to several baselines in order to evaluate empirically how well we can compose classifiers for complex concepts:

- Chance: This is an empirical lower bound for the problem and consists of assigning random scores for image and expression pairs.
- Supervised: This is an empirical upper bound for the problem and consists of training SVMs for every training expression. Thus, it is a fully supervised approach which can not be extended for unknown expressions. Therefore, we just report its performance for known expressions.
- Independent Classifiers: This baseline assumes that visual concepts are independent events and uses basic probability rules to estimate the probability of a complex concept being depicted in an image. They are defined according to the following rules,

$$\begin{aligned} p(v_1 \ \wedge \ v_2) &= p(v_1)p(v_2) \\ p(v_1 \ \vee \ v_2) &= p(v_1) + p(v_2) - p(v_1)p(v_2) \\ p(\neg \, v) &= 1 - p(v) \end{aligned} \tag{5.5}$$

  where $p(v)$ is the probability of a given image has the primitive v estimated by the classifier w_v. Note that in order to estimate these probabilities we calibrate the learned SVMs using a small held-out subset of the training images ($\approx 10\%$) and Platt's calibration method [Platt et al., 1999].

Table 5.1: Evaluating known/unknown disjunctive and conjunctive expressions on the CUB-200 Birds dataset.

	Disjunctive Expressions						Conjunctive Expressions					
	Known Exp.			Unknown Exp.			Known Exp.			Unknown Exp.		
Metrics	MAP	AUC	EER	MAP	AUC	EER	MAP	AUC	EER	MAP	AUC	EER
Chance	39.70	50.00	50.0	40.60	50.00	50.0	4.55	50.0	50.0	4.59	50.0	50.0
Supervised	65.25	74.76	31.58	-	-	-	22.87	78.02	29.69	-	-	-
Independent	58.73	68.39	36.76	60.66	69.28	36.10	17.23	77.22	29.94	19.16	78.00	29.28
Neural Alg. Classifiers	**70.10**	**77.36**	**29.44**	**71.18**	**77.76**	**29.04**	**23.09**	**81.54**	**26.36**	**23.87**	**81.98**	**25.85**

5.3.2 Simple Binary Expressions

In this experiment, we focus on evaluating how well our model can learn to compose classifiers for simple binary conjunctive and disjunctive expressions. We follow the procedure explained in Section 5.3.1 and evaluate our model and baselines on both cases separately. We do not report the result with simple negative expressions since it is a trivial mapping in classifier space as explained in Section 5.2.

We present the results for our methods and baselines on the CUB-200 and AwA2 datasets in Table 5.1 and Table 5.2 respectively. As expected, the supervised method presents good performance on both types of expressions but it is limited to expressions known at training phase. Thus, it can not be used in large scale recognition problems where the number of complex concepts that can be composed is very large.

On the other hand, the independent approach seems to be a strong baseline. It produces slightly worse results than the supervised approach for known expression, mainly on conjunctive expressions, while can classify images according to unknown expressions. However, we note that such a performance is due to the high accuracy of the primitive classifiers, it can reach the AUC of approximately 85% for the CUB-200 and 95% for the AwA2 when classifying validation and test images according to primitive concepts. Then, its performance should decrease drastically in more challenging datasets where the primitive classifiers are often less accurate.

However, our method shows significant superior performance on every setting on both datasets. For instance, the proposed method reaches improvements around 10% for disjunctive expressions and 5% for conjunctive expressions in the CUB-200 dataset. In fact, it is able to surpass the supervised methods on known expression since it allows to learn specific features for complex compositions in addition to reason about correlations between primitives. It is also important to mention that our hypothesis of implementing the disjunctive composition function as the combination of the negation and conjunction according to the De Morgan's laws is verified, since we reach similar performance, when we train a specific MLP network for disjunctive expressions.

Despite the differences highlighted in Section 5.1, we acknowledge the similarity between the transformation function proposed by Misra et al. [2017] and our AND composition function. More specifically, we both learn an MLP, but we use different network architectures and optimize different objectives. Then, we evaluate their model in our simple binary conjunctive expression experiment, the only one that

Table 5.2: Evaluating known/unknown disjunctive and conjunctive expressions on the AwA2 dataset.

	Disjunctive Expressions						Conjunctive Expressions					
	Known Exp.			Unknown Exp.			Known Exp.			Unknown Exp.		
Metrics	MAP	AUC	EER	MAP	AUC	EER	MAP	AUC	EER	MAP	AUC	EER
Chance	53.19	50.0	50.0	53.04	50.0	50.0	18.77	50.0	50.0	21.17	50.0	50.0
Supervised	97.47	97.20	8.13	-	-	-	94.90	98.53	6.00	-	-	-
Independent	97.28	97.12	8.70	97.86	97.58	6.77	93.95	98.13	6.80	93.90	97.87	7.36
Neural Alg. Classifiers	**98.84**	**98.67**	**5.84**	**99.05**	**98.91**	**5.24**	**95.95**	**98.79**	**5.29**	**96.50**	**98.81**	**5.34**

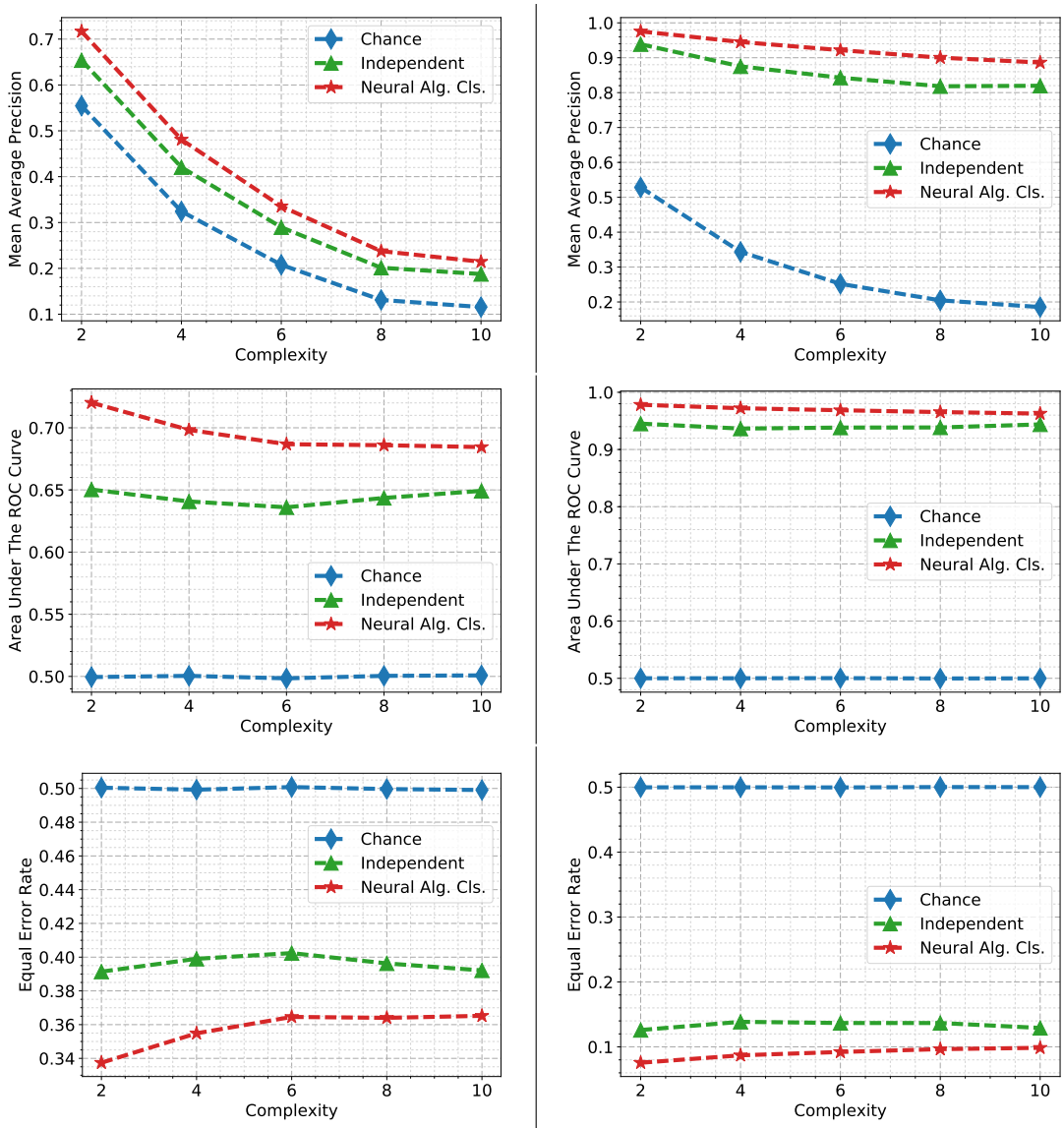
their model is able to handle. Despite their model having approximately 2.7x more learnable parameters, it performs slightly worse than our AND composition (at least 1% in most of the metrics used) which demonstrates the efficiency of our architecture and loss function.

5.3.3 Complex Expressions

From previous experiments, we can conclude that our model is able to learn composition rules for simple binary expressions. However, we still need to show that these models are suitable for arbitrary expressions. According to boolean algebra, every boolean expression can be written in generic forms such as Normal Disjunctive From (NDF) and Normal Conjunctive form (NCF). The former consists of an OR of ANDs, e.g., $(p_1 \wedge q_1) \vee (p_2 \wedge q_2) \vee \ldots \vee (p_c \wedge q_c)$, and the latter consists of an AND of ORs, e.g., $(p_1 \vee q_1) \wedge (p_2 \vee q_2) \wedge \ldots \wedge (p_c \vee q_c)$ where p and q are visual primitives which may appear negated and c is the number of simple terms in those expressions. From the visual recognition perspective, c can be seen as an indicator of the complexity of an expression since long expressions usually defines more specific visual concepts than short expressions. For instance, $(Blue \vee Red) \wedge Socks \wedge (\neg Holes)$ is a more specific visual concept than any of its subexpressions such as $((Blue \vee Red) \wedge Socks)$ and $(Socks \wedge (\neg Holes))$.

Since it is straightforward to convert any expression for both normal forms [Monk and Bonnet, 1989], we decide to examine the performance of our method and baselines on complex expressions in the normal conjunctive form. Towards this end, we randomly generate 1k test CNF expressions of complexity 2, 4, 6, 8, 10 from simple unknown disjunctive expressions. In order to avoid normalization issues when combining linear classifiers produced by our method and the primitives classifiers, we finetune our method using training images and CNF expressions of complexity 4 formed from known simple disjunctive expressions. Then, we use our method and baselines to classify test images according to the sampled CNF expressions of different complexities. Again, the finetune and test expression sets are disjoint as well as the training and test image sets. We also do not evaluate the supervised baseline because we do not have training images for the test expressions.

In Figure 5.4, we plot baselines and our method performance in terms of mean average precision, area under the ROC curve and equal error rate on CNF expressions of different complexities composed by unknown simple binary expressions.



(a) CUB-200 (b) AWA2

Figure 5.4: Performance of the proposed method and baselines on classifying images according to unknown expressions of different complexity described in conjunctive normal form (CNF). The first column presents the results for CUB-200 dataset, while the second column presents the results for AwA2 dataset. In the first row the performance in measured in terms of mean average precision (higher is better), while the second row reports the area under the ROC curve (higher is better), and the third row reports the equal error rate (lower is better).

As expected, the performance of all evaluated methods decrease as we increase the complexity of the test expressions. This is more noticeable in our method which stabilizes for complexity greater or equal to 6. However, we consistently outperform the baselines on classifying images according to expressions of different complexities in both datasets.

5.3.4 Qualitative Evaluation

We now evaluate the proposed method qualitatively by visualizing the classification results of some interesting expressions. More specifically, we classify the test images by scoring them according to manually picked unknown expressions and thresholding using the equal error rate threshold. In Figure 5.5, we show some randomly selected true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) for every selected expression.

Looking back to our motivational example and analyzing the ground truth of CUB-200 dataset, we can state that albatrosses and gulls are birds with hooked beak (HB), black eyes (BE), solid wings pattern (WPS) which do not have black upper tail (UTB) or gray wings (WG). We examine such a statement in the first row of Figure 5.5 by analyzing the classification results produced by our method for the respective boolean expression of these primitives. We note that most of the positive predictions are from different species of albatrosses and gulls. Furthermore, long wings (LW) is a good visual feature to discriminate albatrosses from gulls. Then, we add such a term in the boolean expression and note the predominance of gulls in the predicted positive examples in the second row of Figure 5.5. This example shows qualitatively that our approach is able to group and discriminate objects according to different visual features.

In addition, we can also use our method to find specific combinations of visual features. For instance, consider the following visual features: blue breast (BB), red breast (RB), yellow breast (YB), blue crown (BC), red crown (RC) and yellow crown (YC). In the third row of Figure 5.5, we are looking for birds that have the breast and crown of the same color which could be blue, red or yellow. While in the fourth row of Figure 5.5, we aim for a more specific combinations of these visual primitives like birds that have different breast and crown color. We can note that the predicted positives are predominately unicolor in the former expression, while they are more colorful in the latter one. Furthermore, the false positives usually present part of the desired composition of visual primitives which is perhaps a consequence of the compositional principle.

From the perspective of boolean algebra, two equivalent expressions must have the same truth table. Translating to our context, we can say that two equivalent composition of primitives should have similar classification results. In order to demonstrate such a property, we express the set of big (B) and fast (F) animals that are not hunter (H) in two different ways using De Morgan's Laws: (B AND F) AND (NOT H) and (NOT (S OR SL)) AND (NOT H) where small (S) and slow (SL) are the opposite concepts of fast and big respectively. As we can see in the last two rows of

Figure 5.5, the positive and negative predictions have basically instances from the same classes such as gorillas, deers, horses and dolphins for the positives while elephants, tigers and lions for the negatives. Therefore, our proposed method spans an algebra of visual primitives where complex visual concepts can be described by different compositions.

5.4 Chapter Summary

While the Chapters 3 and 4 provide an effective way to learn visual recognition systems on small human annotated datasets by leveraging the structure on the visual input and output data, the current chapter provides an approach to scale-up these recognition systems to an immeasurable number of visual concepts. Towards this end, we leverage the structural information present on visual classifiers to tackle the problem of learning to synthesize classifiers for complex visual concepts expressed in terms of visual primitives. We formulated such a problem as an algebra of classifiers where the composition rules are learned from data and complex visual concepts are expressed by boolean expressions of primitives. Through a variety of experiments, we show that our framework can synthesize accurate classifiers for known expressions, and generalize to arbitrary unknown expressions. It consistently outperforms the baselines across different metrics and datasets. Besides, we demonstrate qualitatively different queries that can be answered by our model.

Going forward, one compelling direction of investigation is to extend our model for weighted compositions of primitives where we would be able to assign the importance of visual primitives in the composed visual concept. Such a framework would benefit learning-to-rank problems such as image ranking and recommender systems. As an example of application, search queries for online shopping could be described as weighted compositions of visual attributes and the permutation learning framework described in Chapter 3 could be adapted to rank products according tho these compositions. Another important direction is to perform detection and segmentation according to compositions of visual primitives.

Figure 5.5: Randomly selected true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) test images classified according to manually selected unknown expressions of the following visual primitives: hooked beak (HB), black eyes (BE), solid wings pattern (WPS), long wings (LW), blue breast (BB), red breast (RB), yellow breast (YB), blue crown (BC), red crown (RC), yellow crown (YC), big (B), fast (F), hunter (H), small (S) and slow (SL).

Chapter 6

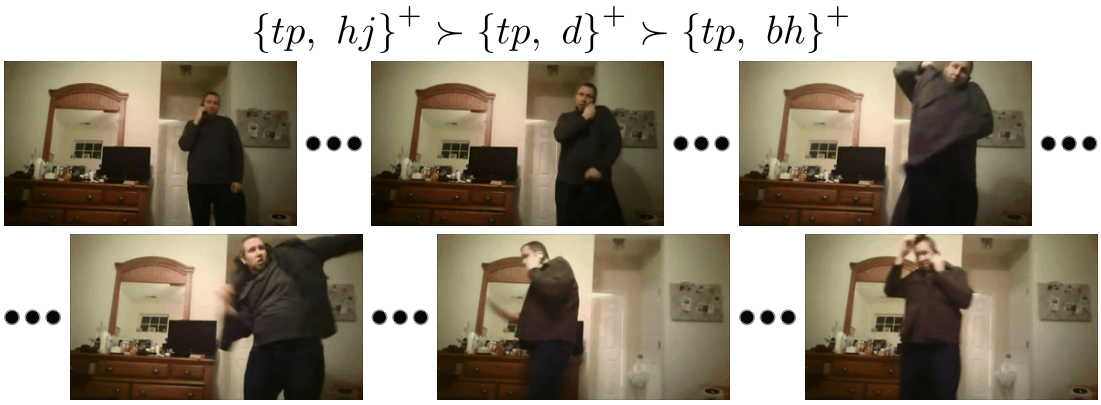
Activity Recognition as Inferring Action Patterns

"Information is the resolution of uncertainty."

Claude Shannon, 1948

In the previous chapters, we have explored structural information and visual priors existent in the input, output and model spaces in order to produce better image rankers, unsupervised image representations and zero-shot image classifiers. While these approaches allow current deep learning models to work in difficult scenarios where annotated data is scarce or the target application has an uncountable number of visual concepts, they still require learning new models which is cumbersome. The current chapter instead proposes to extend existing models for more rich and difficult tasks without requiring new machine learning models or additional data annotation. More specifically, we present a method to recognize compositional activities expressed in terms of regular expressions of simpler actions in videos. To this end, we derive a probabilistic inference framework which provides robust predictions of complex activities in videos with little additional computational effort over standard action classifiers. The proposed approach allows us to unambiguously distinguish between fine-grained actions, retrieve very specific activity instances, and recognize complex composites of actions that may not have a single training sample.

Let us start this chapter by discussing how action recognition has been studied by the computer vision community lately [Kang and Wildes, 2016; Herath et al., 2017]. This problem refers to the act of classifying or localizing an action of interest in videos. In this context, actions can range from being simple and atomic like "running", passing through complex activities with a lot of variability like "cooking a meal", to group activities such as a coordinated movement in team sports. However, the current state-of-the-art models for action recognition tackle these problems in a very limited way where huge volumes of videos are annotated with a limited number of action labels in order to train machine learning models that aim to recognize the annotated actions in new video instances. This paradigm presents notorious limitations since recognizing new action categories requires annotating hundreds of videos and retraining computationally expensive machine learning models.



Natural language queries:
(a) "Someone is talking on the phone, dressing a jacket and brushing hair.";
(b) "Someone is talking on the phone and holding a jacket, then he dresses it and brushes his hair.";
(c) "Someone is talking on the phone while dressing a jacket and brushing hair.";

Figure 6.1: A complex activity can be described by natural language queries, which are often incomplete and have vague and/or ambiguous temporal relations between the constituent actions. For instance, option (a) does not mention all the actions involved, and it is not clear from options (b) and (c) whether the actions happen simultaneously or sequentially. In contrast, a regular expression of primitive actions can precisely describe the activity of interest. For instance, given the primitive actions "talking on the phone" (tp), "holding a jacket" (hj), "dressing" (d), and "brushing hair" (bh), the regular expression $\{tp,\ hj\}^+ \succ \{tp,\ d\}^+ \succ \{tp,\ bh\}^+$ precisely describes the activity depicted in the frames, where the sets of primitive actions, the regular language operator 'concatenation' ($\succ$) and the operator 'one-or-more repetition' ($+$) define concurrent, sequential and recurrent actions, respectively.

Recent methods try to circumvent these limitations by leveraging textual data, allowing zero-shot action classification [Jain et al., 2015; Mettes and Snoek, 2017], action localization [Gao et al., 2017; Hendricks et al., 2017; Liu et al., 2018], and actor and action segmentation [Gavrilyuk et al., 2018] from natural language sentences and word vector representations. However, natural language descriptions are inherently ambiguous and not suitable to describe the activity of interest precisely. As an example, all natural language descriptions listed for the video shown in Figure 6.1 are true, but none of them describe the sequence of events precisely. Description (a) is incomplete since it does not mention all the events that occurs, and it is not clear from descriptions (b) and (c) whether the actions happen simultaneously or sequentially, e.g., , it is not clear whether the man is talking on the phone at the same time he is holding the jacket.

In this chapter, we instead build on the insight that complex activities are fundamentally compositional action patterns and develop a probabilistic inference framework to *unambiguously* describe and efficiently recognize compositional activities in videos. Towards this end, we first propose to describe complex activities as regular expressions of simple primitive actions using *regular language operators*. Then, we de-

velop a probabilistic model that can recognize these regular expressions in videos. Returning to the example in Figure 6.1, given the primitive actions "talking on the phone" (tp), "holding a jacket" (hj), "dressing" (d), and "brushing hair" (bh), the regular expression $\{tp,\ hj\}^+ \succ \{tp,\ d\}^+ \succ \{tp,\ bh\}^+$ precisely describes the activity depicted in the video, where the sets of primitive actions $(\{\cdot\})$, the regular language operators 'concatenation' $(\succ)$ and 'one-or-more repetition' $(+)$ define concurrent, sequential, and recurrent actions respectively.

In summary, the proposed approach provides a framework to precisely describe complex activities and recognize instances of them in videos. For example, our approach can recognize the activity of "making a caesar salad" by exploiting action classifiers for "boiling eggs", "chopping leaves", and "preparing dressing". Moreover, such a framework is able to predict complex unseen activities. For example, it is possible to identify an event of "Olympic goal" in a soccer game from primitive actions such as "corner kick", "ball traveling", and "goal", even if such a complex activity does not exist in the training data. Our framework can also express alternative ways an activity can be performed and form groups of activities by using the alternation operator $(|)$. Therefore, our framework scales up action recognition systems for complex and dynamic scenarios.

However, the development of such a probabilistic framework is a challenging task. Naively, we can think of recognizing a compositional activity whenever a set of primitive action classifiers activate simultaneously, regardless of the temporal relations between primitives. However, this naive approach cannot express the sequential or alternative order of primitive actions. In contrast, the proposed approach provides a formal language that allows us to express these temporal relations between primitives in order to recognize complex activities, specific instances, and groups of activities without additional annotations.

The current chapter contributions are threefold. First, we propose rich compositional activity recognition as the task of recognizing complex activities described by patterns of primitive actions in videos. We formulate a framework for this task that resembles a regular expression engine in which we can perform inference for any activity that can be described by a regular expression of primitive actions. Second, we derive deterministic and probabilistic models for solving the inference problem based on uncertain classifiers. Third, we present an extensive evaluation of the proposed models under different scenarios simulated by a synthetic dataset, in addition to applications in trimmed and untrimmed video composite action classification using challenge datasets such as MultiTHUMOS [Yeung et al., 2017] and Charades [Sigurdsson et al., 2016].

6.1 Scaling-Up Action Recognition Models

State-of-the-art action recognition methods aim to recognize actions from a predefined fixed vocabulary of actions [Bilen et al., 2017; Carreira and Zisserman, 2017; Donahue et al., 2015; Wang and Cherian, 2018] and ignore the recognition of long

tailed distribution of complex activities. In this section, we review methods that attempt to overcome this limitation through zero-shot, compositional, and natural language-based learning approaches.

Zero-shot learning consists of recognizing unseen visual concepts by exploring some external source of information [Lampert et al., 2014]. In the context of zero-shot action recognition, different external sources of information have been explored such as action–attribute relationship [Liu et al., 2011], object annotations [Jain et al., 2015], word embeddings learned on a large corpus [Xu et al., 2017c], and textual descriptions from web data [Habibian et al., 2017]. These models, however, still nurture the interpretation of action recognition as the assignment of simple action labels. In contrast, we propose a compositional view of action recognition where complex actions are inferred from simple primitive actions.

Recently, we have seen the success of vision-language models in related problems in the image domain [Hu et al., 2016b,a; Li et al., 2017]. Inspired by the success of these approaches, Gao et al. [2017] and Hendricks et al. [2017] strive to localize activities by natural language queries using cross-model alignment frameworks, Gavrilyuk et al. [2018] propose an encoder-decoder neural network architecture to perform action and actor segmentation from natural language sentences, and Liu et al. [2018] propose a modular network for the task of video retrieval using natural language queries. We argue, however, that natural language sentences may lead to ambiguous descriptions of complex activities as shown in Figure 6.1, which makes it difficult to find correct matches between videos and queries. In order to solve this problem, we provide a regular language to unambiguously describe and efficiently infer compositional activities in videos.

Serving as inspirations for our approach, the innovative works of İkizler and Forsyth [2008] and Vo and Bobick [2014] recognize human-centered activities using compositions of primitive actions. In addition to mainly focusing on human-centered activity recognition problems, these works differ from ours in the expressiveness of the language used to specify the activity query. The former uses *strings* of primitive actions, while the latter use a *simplified context-free grammar* whose production rules are AND-rules or OR-rules without recursion. These approaches can only express sequential or alternative primitive actions of fixed length. In contrast, we propose a more expressive and complete language for querying complex activities. More specifically, we propose *regular expressions on subsets of primitive actions* that can express sequential, concurrent, alternative, and recursive actions. Furthermore, our approach focuses on zero-shot recognition of complex activities, unlike these approaches which require training data for the queried activities.

It is also important to distinguish our approach from ones that perform structured prediction of sequences of primitive actions from a fixed vocabulary, leveraging training data of human annotated action composites. For instance, Richard and Gall [2016] use a language model while Piergiovanni and Ryoo [2018] use temporal filters to learn temporal and contextual correlations between primitive actions in order to better infer sequences of these primitive actions in videos. Note that these models address the problem of recognizing primitive actions from a fixed vocabulary

and they cannot automatically infer whether an unseen action composite occurs in a given video, unlike our proposed model.

6.2 Inferring Action Patterns in Videos

In this section, we start by formalizing the problem of recognizing complex activities described by regular expressions of primitive actions. Then we derive our approach starting from a deterministic model and evolving to a probabilistic framework where the uncertainty of the predictions are taken into account.

6.2.1 Action Patterns Formulation

Our problem consists of recognizing activities expressed as regular expressions of subsets of primitive actions. We denote these expressions as action patterns. By way of example, let us assume we have a set of known actions, called primitives, like "driving" (d), "getting in the car" (gc), "talking on the cellphone" (tc), "talking to someone" (ts). In addition, consider three basic composition rules inspired by the standard regular expressions operators: concatenation ($\succ$) which defines sequences of patterns, alternation ($|$) which builds a union of patterns, and Kleene star ($\star$) which allow us to express recurrent patterns. Other useful operators can be defined in terms of these ones, e.g., one-or-more repetition ($+$) is defined as $x^+ \triangleq x \succ x^\star$. The problem then becomes how to recognize whether a video depicts a complex activity described by recursive compositions of subsets of primitive actions and these operators. For instance, can we find on YouTube "someone driving and talking on the phone or to someone, repeatedly, just after getting in the car", which can be described without ambiguity as "$a_{gc} \succ (\{a_d,\ a_{tc}\} \mid \{a_d,\ a_{ts}\})^\star$".

Formally, let us define a set of *primitive actions* $\mathcal{A} = \{a_i\}_{i=1}^{M}$. We can express a complex activity by forming *action patterns*, an arbitrary regular expression r combining subsets of primitives $w \in \mathcal{P}(\mathcal{A})$, where $\mathcal{P}(\mathcal{A})$ is the power-set of $\mathcal{A}$, with the aforementioned *composition rules* $\mathcal{O} = \{\succ, |, \star\}$. Note that this formulation expresses concurrent actions as subsets of primitive actions. Consequently, background actions and non-action video segments are represented by the null primitive $\emptyset \in \mathcal{P}(\mathcal{A})$. Our goal then is to model a function $f_r : \mathcal{V} \to [0,1]$ that assigns high values to a video $v \in \mathcal{V}$, where $\mathcal{V}$ is the set of all videos, if it depicts the action pattern described by the regular expression r and low values otherwise.

It is also important to mention that we assume the set of primitive actions $\mathcal{A}$ used to specify the action patterns are defined a priori. This set is a design decision highly dependent of the target application. For instance, in a soccer match broadcasting application, primitive actions like shooting, header, and tackling are very important to specify meaningful events. On the other hand, in a surveillance application, primitive actions like 'breaking in', running, and hiding are key to recognize complex misbehavior. In addition, one should also consider technical factors and trade-offs like data availability, discriminativeness, and classifier accuracy of the selected action primitives. Therefore, optimally defining this set of primitive actions is a complex

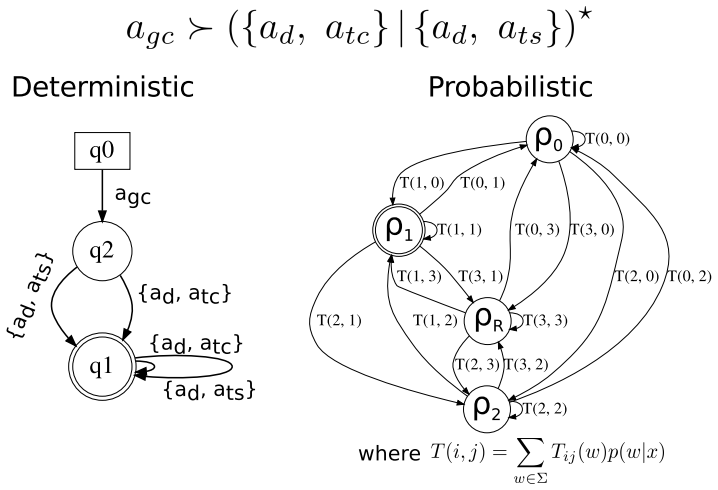


Figure 6.2: Deterministic and probabilistic inference models for the compositional action "driving and talking on the phone or to someone, repeatedly, just after getting in the car" described by the expression "$a_{gc} \succ (\{a_d,\ a_{tc}\} \mid \{a_d,\ a_{ts}\})^\star$". The deterministic model is a DFA and the probabilistic model is a PA, compiled for the given regular expression.

problem by itself which is beyond the scope of this chapter, but it is a very compelling direction for future work.

6.2.2 Deterministic Inference Model

Regular expressions have been studied for many years in the field of theoretical computer science [Lawson, 2003] and formal language theory [Mitkov, 2003]. In the natural language processing context, a regular expression is used to concisely specify a pattern of characters for matching and searching in large texts [Sedgewick and Wayne, 2011]. Inspired by these ideas, we first propose a deterministic model based on Deterministic Finite Automaton (DFA) [McCulloch and Pitts, 1943; Rabin and Scott, 1959] to the problem of recognizing activities described by regular expressions of action primitives.

Let us start by defining a DFA M_r for a regular expression r as a 5-tuple $\big(\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{F}\big)$, consisting of a finite set of states Q, a finite set of input symbols called the alphabet Σ, a transition function $\delta : \mathcal{Q} \times \Sigma \to \mathcal{Q}$, an initial state $q_0 \in \mathcal{Q}$ and a set of accept states $\mathcal{F} \subseteq \mathcal{Q}$. In our problem, the alphabet Σ is the power-set of action primitives $\mathcal{P}(\mathcal{A})$ and any subset $w \in \mathcal{P}(\mathcal{A})$ can define a transition in δ. Note that all these structures are constructed from a given regular expression r and can be efficiently obtained and optimized with traditional algorithms such as non-deterministic finite automaton (NFA) construction [Ilie and Yu, 2002], the NFA to DFA subset construction algorithm [Rabin and Scott, 1959], and Hopcroft's DFA minimization algorithm [Hopcroft, 1971]. Figure 6.2 shows an example of an action pattern and its compiled DFA.

Additionally, let us denote the probability of a primitive action $a \in \mathcal{A}$ happening in frame x as $p(a|x)$ which can be obtained from the output of a probabilistic classi-

fier. Then, we say that the deterministic model accepts an input video $v = \langle x_1, \ldots, x_n \rangle$ if and only if there exists a sequence of states $\langle q_0, \ldots, q_n \rangle$ in $\mathcal{Q}$ such that (i) it starts in the initial state q_0, (ii) subsequent states q_i is defined as $q_{i+1} = \delta\left(q_i, w\left(x_{i+1}\right)\right)$ for $i = 0, \ldots, n-1$, and (iii) it finishes in a final state $q_n \in \mathcal{F}$. Here, the symbol $w(x) = \{a \in \mathcal{A} \mid p(a|x) \geq \tau\}$ for the frame x is obtained by thresholding the primitive actions predictions $p(a|x)$ with the model's hyper-parameter τ which should be set by cross-validation.

This procedure defines a binary function that assigns a value of one to videos that reach the final state of the compiled DFA M_r and zero otherwise. This is a very strict classification rule since a positive match using non-perfect classifiers is very improbable. In order to relax such a classification rule, we propose implementing the score function

$$f_r(v) = \frac{dist(q_0, \hat{q})}{dist(q_0, \hat{q}) + \min_{q_f \in \mathcal{F}} dist\left(\hat{q}, q_f\right)} \tag{6.1}$$

where $\hat{q}$ is the state in which the compiled DFA M_r halted when simulating the sequence of frames defined by the video v, and the function $dist(q_x, q_y)$ computes the minimum number of transitions to be taken to reach the state q_y from state q_x. That is, for a given regular expression, the deterministic model scores a video according to the fraction of transitions taken before halting in the shortest path to a final state in the compiled DFA.

In summary, the proposed deterministic model implements the function f_r by computing Equation 6.1 after simulating the DFA M_r compiled for the regular expression r on the sequence of symbols generated by thresholding the action primitive classifiers $p(a|x)$ on every frame x of the input video v. Note that this model has considerable limitations since it requires the primitive classifiers to produce correct predictions for all primitives in every frame of the input video.

6.2.3 Probabilistic Inference Model

Probabilistic models are generally preferable for pattern recognition problems because they are able to break the problem down into two separate stages: the inference stage where we estimate posterior probabilities, and the subsequent decision stage where we use these posterior probabilities to make optimal decisions which are often influenced by the application requirements. In order to develop a proper probabilistic model for our problem, we propose to use Probabilistic Automatons (PA) [Rabin, 1963] instead of a DFAs as the backbone of our framework.

Mathematically, let us define a probabilistic automaton U_r for a regular expression r as a 5-tuple $(\mathcal{Q}, \Sigma, T, \boldsymbol{\rho}, \mathcal{F})$ where $\mathcal{Q}$, Σ, and $\mathcal{F}$ are defined as before, while $T(\cdot)$ is a function from the alphabet Σ into the states' transition distributions and $\boldsymbol{\rho}$ is the initial distribution over states. More specifically, $\boldsymbol{\rho} \in \mathbb{R}^{|\mathcal{Q}|}$ is a stochastic vector and $\boldsymbol{\rho}_i$ is the probability that the automaton starts at state q_i. Likewise, $T(w) \in \mathbb{R}^{|\mathcal{Q}| \times |\mathcal{Q}|}$ is a row stochastic matrix associated with the symbol w and the entry $T_{i,j}(w)$ is the probability that the automaton transit from the state q_i to the state q_j after reading

the symbol $w \in \Sigma$.

Note that all these structures can be estimated from the transition function δ, initial state q_0, and final states $\mathcal{F}$ of the compiled DFA M_r for the same regular expression r as follows,

$$\begin{aligned} T_{i,j}(w) &= \frac{[\![\delta(i,w) = j]\!] + \alpha}{\sum_{k \in \mathcal{Q}} [\![\delta(i,w) = k]\!] + \alpha|\mathcal{Q}|}\,, \\ \boldsymbol{\rho}_i &= \frac{[\![q_0 = i]\!] + \alpha}{\sum_{k \in \mathcal{Q}} [\![q_0 = k]\!] + \alpha|\mathcal{Q}|}\,, \end{aligned} \tag{6.2}$$

where the indicator function $[\![c]\!]$ evaluates to one when the condition c is true and zero otherwise. The smoothing factor α is model hyper-parameter that regularizes our model by providing non-zero probability for every distribution in our model. In addition, unlike a DFA which fails to match when there is no transition for a given state and input symbol, the PA needs to explicitly model the reject state by adding transitions to it whenever an unexpected symbol appears with high-probability in a given state. Figure 6.2 shows an example of a regular expression and its PA.

However, PAs do not model uncertainty in the input sequence which is a requirement of our problem, since we do not know what actions are depicted in a frame during inference. Therefore, we propose to extend the PA framework by introducing a distribution over the alphabet Σ given a video frame. In order to make use of off-the-shelf action classifiers like modern deep leaning models, we assume independence between the action primitives and estimate the probability of a symbol given a frame $p(w|x)$ as

$$p(w|x) = \left(\prod_{a \in \mathcal{A}} p(a|x)^{[\![a \in w]\!]} \left(1 - p(a|x)\right)^{(1 - [\![a \in w]\!])} \right)^{\gamma}, \tag{6.3}$$

where $p(a|x)$ is the prediction of a primitive action classifier as before and γ is a hyper-parameter that compensates for violations to the independence assumption. After such a correction, we need to re-normalize the $p(w|x)$ probabilities in order to form a distribution.

Finally, we can compute the normalized matching probability as the probability of reaching a final state after seeing an input video $v = \langle x_1, \ldots x_n \rangle$ as

$$P_{U_r}(v) = \left(\boldsymbol{\rho}^{\intercal} \prod_{i=1}^{|v|} \sum_{w \in \Sigma} T(w) p(w \mid x_i) \right)^{\frac{1}{|v|}} \boldsymbol{f}\,, \tag{6.4}$$

where $\boldsymbol{f}$ is an indicator vector such that $\boldsymbol{f}_i = 1$ if and only if $q_i \in \mathcal{F}$ and 0 otherwise. The normalization by $|v|$ calibrates the probabilities to allow comparisons between videos of different length. It is also important to note that naively computing such a probability is problematic since it requires marginalization over every symbol in our

large alphabet Σ, which is the power-set of action primitives $\mathcal{P}(\mathcal{A})$. For instance, a modestly sized set of 100 primitive actions would generate an alphabet of 2^{100} symbols. In order to circumvent such a limitation, we factorize the marginalization over the alphabet in Equation 6.4 as

$$\sum_{w \in \Sigma} T(w) p(w \mid x_i) = \sum_{w \in \Sigma'} T(w) p(w \mid x_i) + \hat{T} \left(1 - \sum_{w \in \Sigma'} p(w \mid x_i) \right), \tag{6.5}$$

where we first define a typically small subset of symbols $\Sigma' \subseteq \Sigma$ composed of symbols that have at least one transition in the compiled DFA M_r and make use of the fact that the other symbols will have exactly the same transition distribution matrix $\hat{T}$ and the sum of their probability is equal to $(1 - \sum_{w \in \Sigma'} p(w \mid x_i))$. Therefore, the matching probability can be efficiently computed without enumerating all symbols in the alphabet.

In summary, our goal is to compute the match probability between a input video $v = \langle x_1, \dots x_n \rangle$ and action pattern r, where the video is defined by a sequence of frames x and the action pattern by the regular expression operators $\mathcal{O}$ and the set of action primitives $\mathcal{A}$. We also assume the existence of probabilistic classifiers for these primitive actions $\{p(a|x) | a \in \mathcal{A}\}$, e.g., a neural network trained to classify these primitive actions in a video frame. In order to meet our goal, we first compile the action pattern r to a DFA as described in Section 6.2.2 and transform the resulting DFA to a PA using Equation 6.2. Such a step produces the 5-tuple $U_r = (\mathcal{Q}, \Sigma, T, \boldsymbol{\rho}, \mathcal{F})$ defining the PA U_r for action pattern r. Then, we estimate the symbol distribution $p(w|x)$ for every frame x of the input video v using the primitive action classifiers $p(a|x)$ according to Equation 6.3. Finally, we compute the matching probability $P_{U_r}(v)$ between the action pattern r and the video v by applying Equation 6.4 factorized as Equation 6.5 which are defined in terms of the elements in the 5-tuple U_r and the just computed symbol distribution $p(w|x)$. Intuitively, this formulation assigns higher probabilities to videos that exhibit the primitive actions according to the given action pattern and measured by the primitive action classifiers.

6.3 Experiments

We now evaluate the proposed inference models for rich compositional activity recognition. We first perform a detailed analysis of the proposed approaches on controlled experiments using synthetic data. Then, we test the utility of our methods on challenging action recognition tasks using well-known datasets.

6.3.1 Analysis with Synthetic Data

It is unrealistic to collect video data for the immense number of possible regular expressions that our models may encounter. As such, we resort to the use of synthetically generated data inspired by the well known Moving MNIST dataset [Srivastava

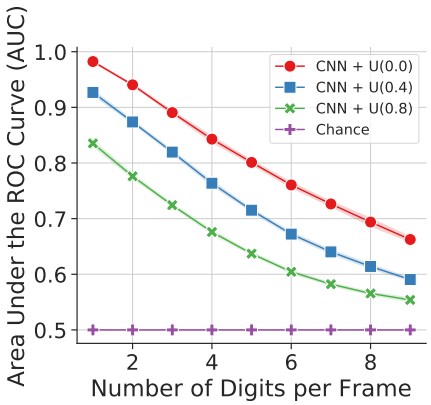


Figure 6.3: The performance of the primitive classifiers on the test set with a different number of digits per frame and under different noise levels. $U(x)$ denotes uniform additive noise between $[-x, x]$ and the classifiers' predictions are re-normalized using a softmax function.

et al., 2015a]. More specifically, we develop a parametrized data generation procedure to produce moving MNIST videos depicting different patterns of appearing MNIST digits. Such a procedure can generate videos that match regular expressions of the form

$$w_1^+ \succ \cdots \succ \left(\left(w_s^{1^+} \succ \cdots \succ w_n^{1^+} \right) \Big| \cdots \Big| \left(w_s^{d^+} \succ \cdots \succ w_n^{d^+} \right) \right), \tag{6.6}$$

where the symbols $w \in \mathcal{P}(\mathcal{A})$ are subsets of the primitives $\mathcal{A}$ which are the ten digit classes. The data generation procedure has the following parameters: the number of primitives that simultaneously appear in a frame $|w|$, the total number of different sequential symbols n, the number of alternative sequences of symbols d, the start position s of each alternative sequence in the pattern, and the total number of generated frames. Since complex patterns can match different sequences of symbols due to the the alternation operator $(|)$, we perform random walks from the start state until reaching a final state in the compiled DFA in order to generate video samples for a given regular expression.

Figure 6.4 shows in detail different types of expressions and the resulting video clips generated by this data generation procedure. We start with a simple example in the first row, where we have just one moving digit—starting as a six and transitioning to a seven as the video progresses. In the action recognition context, this example is analogous to two sequential actions such as "running followed by jumping". In the second row, we show a more complicated expression which has three digits per frame $(|w| = 3)$. This expression simulates concurrent primitives actions, i.e., actions that happen simultaneously for a period of time. For instance, "talking on the phone while holding a jacket". Likewise, in the third row, we show an even more complex pattern where we increase the number sequential symbols $(n = 4)$. Now, we have four sets of three concurrent actions. The fourth and fifth rows in Figure 6.4, show

two ways that the 'alternation' $(|)$ operator can be used to produce different types of regular expressions. We form a union of two different patterns in the fourth row $(d = 2, s = 0)$, while we show a pattern with two alternative endings $(d = 2, s = 2)$ in the fifth row. In the action recognition context, the former is analogous to groups of activities, while the latter can describe alternative ways that an activity can be performed. Note also that we use different digit images when generating the videos, we can generate an arbitrary number of frames for each expression, and we can also generate videos that does not match with any given expression. The current section presents a quantitative evaluation of the proposed inference models on these increasingly complex regular expressions.

Using the synthetically generated data, we first train the primitive classifiers on frames depicting a different number of digits obtained from the MNIST training split. The primitive classifiers consist of a shallow CNN trained to minimize the binary cross entropy loss for all digits in a vast number of frames. In order to evaluate the robustness of the proposed models, we also generate worse versions of these classifiers by adding noise to their predictions. Figure 6.3 shows the performance of the learned primitive classifiers on different levels of noise and different numbers of digits per frame. Note that more digits per frame implies more occlusion between digits since the frame size is kept constant, which also decreases the classifier's performance.

Finally, using this synthetic data and the trained primitive classifiers, we test our models for the inference of different regular expressions by setting all the data generation parameters to default values with the exception of the one being evaluated. We use the following default values — $|w| = 3$ digits per frame, $n = 3$ different sequential symbols, $d = 2$ alternative sequences starting from $s = 2$, depicted on video clips of 32 frames. In Figure 6.5, we plot standard classification/retrieval metrics, e.g., Area Under the ROC Curve (AUC) and Mean Average Precision (MAP), against different data generation parameters. More specifically, at each configuration, using the MNIST test split, we generate 100 expressions with 20 positive samples, totaling about 2000 video clips. In order to robustly report our results, we repeat the experiment ten times reporting the mean and standard deviation of the evaluation metrics. We also cross-validate the model hyper-parameters, τ for the deterministic model and α and γ for the probabilistic model, in a validation set formed by expressions of similar type as the ones to be tested, but with video clips generated from a held-out set of digit images extracted from the training split of the MNIST dataset.

As can be seen, the probabilistic model performs consistently better than the deterministic model in all experiments, providing precise and robust predictions. In most of the experiments, the probabilistic model presents performance about 40% better than its deterministic counterpart on both metrics. Furthermore, the probabilistic model is more robust to high levels of noise in the primitive classifiers' predictions. While the deterministic model works as poorly as random guessing with high noise levels, e.g., $U(0.8)$, the probabilistic model still produces good results.

In addition, the probabilistic model works consistently across different kinds of regular expressions. Its performance is almost invariant to most of the regular ex-

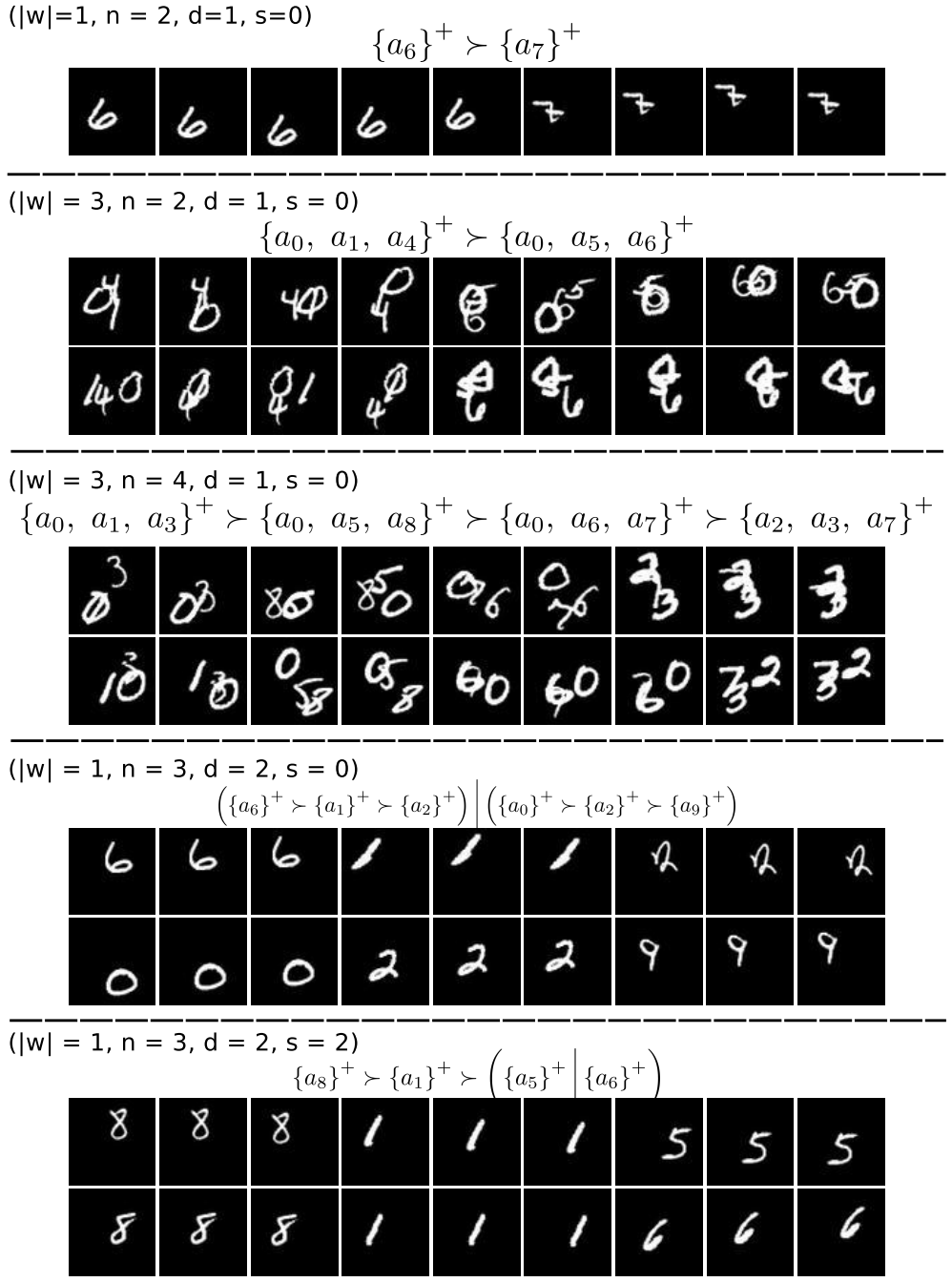


Figure 6.4: Regular expressions and corresponding positive video clips synthetically generated using the Moving MNIST dataset [Srivastava et al., 2015a]. The expressions are parametrized according to Equation 6.6 and the parameters: the number of primitives that simultaneously appear in a frame $|w|$, the total number of different sequential symbols n, the number of alternative sequences of symbols d, and the start position s of each alternative sequence in the pattern.

pressions parameters evaluated, except the number of digits per frame $|w|$ for which some performance degradation is observed. Such a degradation correlates with the decrease in performance presented by the primitive classifiers as the number of digits per frames is increased (see Figure 6.3). The probabilistic model, however, is able to mitigate such a degradation. For example, comparing the performance at two and five digits per frame, we observe that a drop of about 16% in AUC on the primitive classifiers performance causes a reduction smaller than 6% in AUC on the probabilistic model performance.

6.3.2 Evaluation on Action Recognition Datasets

We now focus on evaluating the utility of our model for action recognition problems. We first describe the experimental setup, metrics and datasets used in our experiments. Then we analyze how effectively our model can recognize activities described by regular expressions in trimmed and untrimmed videos.

Experimental Setup. In order to evaluate the proposed inference models in the action recognition context, we collect datasets of regular expressions and video clips by mining the ground-truth annotation of multilabel action recognition datasets such as Charades [Sigurdsson et al., 2016] and MultiTHUMOS [Yeung et al., 2017]. More specifically, we search for regular expressions of the type defined in Equation 6.6 where the symbols w are subsets of the primitive actions annotated in the datasets. For instance, Charades has 157 actions, while MultiTHUMOS has 65 actions. Given the regular expressions parameters, we first form instances of regular expressions using the primitive actions present in the datasets, keeping the ones that have at least one positive video clip. Then, using these instances of regular expressions, we search for all positive video clips in the dataset in order to form a new dataset of regular expressions and video clips which will be used in our experiments.

Aiming at fair evaluation of the inference models proposed, we train the primitive classifiers to independently recognize the primitive actions on the training split of the selected datasets. We use the I3D model [Carreira and Zisserman, 2017], finetuned on the Charades and MultiTHUMOS datasets, as our primitive action classifiers. In this work, we only use the I3D-RGB stream, but optical flow and other information can be easily added since our formulation depends only on the final predictions of the primitive classifiers. Using the frame-level evaluation protocol (i.e., Charades localization setting), this model reaches 16.12% and 24.93% in MAP on classifying frames into primitive actions on the test split of Charades and MultiTHUMOS datasets respectively. Once these classifiers are learned, we use them in the proposed models to infer compositional activities mined from the action recognition datasets. We, first, cross-validate the hyper-parameters of the proposed inference models using expressions and video clips mined from the training split, and then we evaluate the models in a different set of expressions mined from the test split of the action recognition datasets. It is important to emphasize that the expressions mined for testing are *completely* different from the ones used for cross-validation. Therefore, the pro-

posed models have not seen any test frame or the same action pattern before, which provides an unbiased evaluation protocol. In order to provide robust estimators of performance, in the experiments of the current section, we repeat the data collection of 50 regular expressions and the test procedure steps ten times, reporting the mean and standard deviation of the evaluation metrics AUC and MAP. Note that these metrics are computed over the recognition of the *whole complex activity* as a singleton label. They are *not* computed per primitive.

Comparison To Standard Action Recognition. Traditional action classification aims to recognize a single action in a video, making no distinction if the action is performed alone or in conjunction with other actions. Abusing the proposed regular expression notation, for now consider the symbols w in Equation 6.6 as the collection of all subsets of the primitive actions that contains the actions in w. For instance, only here the symbol $\{a_2, a_3\}$ represents the set of symbols $\Big\{\{a_2, a_3\}, \{a_2, a_3, a_4\}, \ldots, \{a_2, a_3, a_4, \ldots, a_{|\mathcal{A}|}\}\Big\}$. Then, we can say that the traditional action classification problem is the simplest instance of our formulation where the input regular expressions are of the type $\{a\}^+$, meaning one or more frames depicting the action a alone or in conjunction with other actions. Therefore, starting from this simplified setup, we analyze how our models behave as we increase the difficult of the problem by dealing with more complex regular expressions. More specifically, we start from this simplest form, where all the regular expression parameters are set to one, and evolve to more complex expressions by varying some of the parameters separately. Figure 6.6 presents the results on the MultiTHUMOS and Charades datasets where we vary the number of concurrent (columns 1 and 4), sequential (columns 2 and 5), and alternated actions (columns 3 and 6) by varying the number of primitives per symbol $|w|$, the number of sequential symbols n, and the number of alternative sequences d in the mined regular expression and video clip data, respectively.

Note that there is a significant difference in performance when compared to the results in Section 6.3.1. Such a difference is due to the quality of the primitive classifiers available for a challenging problem like action classification. For instance, the digits classifiers for the MNIST dataset are at least three times more accurate than the primitive action classifiers for Charades or MultiTHUMOS. However, different from the deterministic model, the probabilistic model is able to extend the primitive action classifiers, the I3D model, for complex expressions without degenerating the performance significantly. For instance, considering all setups, the probabilistic model presents a reduction in performance of at most 15% in both datasets and metrics used. It is a very useful result which means that the proposed probabilistic inference procedure can scale up the developments in traditional action classification to compositional activity recognition without significant additional effort.

Figure 6.5: Plots of the performance, in terms of AUC and MAP, of the proposed methods on the generated synthetic dataset using primitive classifiers with different levels of noise as shown in Figure 6.3. The generated data consists of video clips depicting regular expressions parametrized according to Equation 6.6. We evaluate the proposed approaches according to the following data parameters: the number of digits that simultaneously appear in a frame $|w|$, the total number of different sequential symbols n, the variance in number of frames in the videos, the number of alternative sequences of symbols d, and the start position s of each alternative sequence in the pattern respectively.

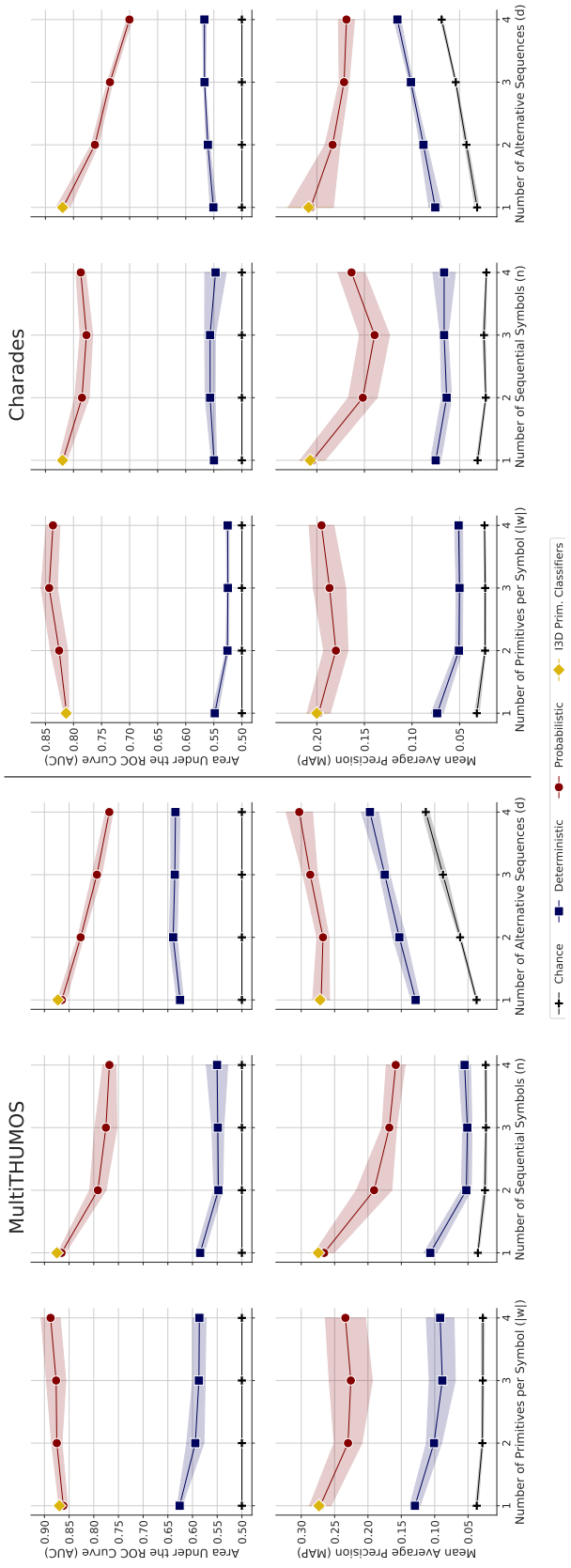


Figure 6.6: Comparison with standard action classification. Plots of the performance, in terms of AUC and MAP, of the proposed methods using the I3D model [Carreira and Zisserman, 2017] as the primitive action classifier. We evaluate the models on collections of regular expressions of different complexity mined from the test videos of MultiTHUMOS and Charades datasets. These regular expressions follows the format defined in Equation 6.6 where all the variables are set to 1 with the exception of the one being evaluated. For instance, for the plot with variable number of sequential symbols (n) the expressions are of the type $(w_1^+), \ldots, (w_1^+ \succ \cdots \succ w_4^+)$. Differently from the other experiments, the symbols here denote any subset that contains the primitives.

Table 6.1: Results for activity classification in trimmed videos on MultiTHUMOS and Charades datasets.

	MultiTHUMOS		Charades	
Method	AUC	MAP	AUC	MAP
Chance	50.00 ($\pm$0.0)	2.00 ($\pm$0.00)	50.00 ($\pm$0.0)	2.00 ($\pm$0.00)
Deterministic	52.46 ($\pm$0.77)	3.66 ($\pm$0.48)	51.85 ($\pm$0.83)	4.40 ($\pm$1.15)
Probabilistic	73.84 ($\pm$2.63)	13.76 ($\pm$1.93)	74.73 ($\pm$2.35)	15.19 ($\pm$1.09)

Trimmed Compositional Activity Classification. In this experiment, we evaluate the ability of the proposed algorithms to recognize very specific activities in trimmed video clips which depict only the entire activities from the beginning to the end. Different from the previous experiment, but like the other experiments, the input regular expressions are formed by symbols that are only subsets of primitives. For instance, the symbol $\{a_2, a_3\}$ means that the primitive actions $a_2, a_3 \in \mathcal{A}$ happen exclusively in a frame. In addition, we mined test sets for regular expressions with different combinations of parameters ranging jointly from 1 to 6. Table 6.1 presents the results.

We would like to emphasize the difficulty of the problem where the chance performance is only about 2% MAP in both datasets. The deterministic model works only slightly better than chance, which is also a consequence of the imperfect quality of the primitive classifiers due to the difficult of action recognition as discussed before. On the other hand, the probabilistic model provides gains above 20% in AUC and 10% in MAP compared to the deterministic approach in both datasets. This shows the capability of the probabilistic formulation to surpass the primitive classifiers' imprecision even when the activity of interest is very specific, producing a very complex regular expression.

Untrimmed Compositional Activity Classification. In this task, we evaluate the capability of the proposed models for recognizing specific activities in untrimmed videos which may depict the entire activity of interest at any part of the video. Here, videos can contain more than one activity, and typically large time periods are not related to any activity of interest. In this context, we modify the mined regular expressions to allow matches starting at any position in the input video. It is easily accomplished by doing the following transformation: $re \rightarrow .^\star re.^\star$ where $(.)$ is the "wildcard" in standard regular expression engines and in our formulation consists in every subset of primitive actions. In addition, we do not trim the video clips, instead, we compute matches between the mined regular expressions and the whole video aiming to find at least an occurrence of the pattern in the entire video. We present the results on Table 6.2 where we compute matches between regular expressions and the videos that have at least one positive video clip for the set of mined regular expressions.

In the same fashion as the previous experiments, the probabilistic model performs significantly better than the deterministic model. More specifically, the perfor-

Table 6.2: Results for activity classification in untrimmed videos on MultiTHUMOS and Charades datasets.

	MultiTHUMOS		Charades	
Method	AUC	MAP	AUC	MAP
Chance	50.00(±0.0)	4.21(±0.20)	50.00(±0.0)	2.58(±0.01)
Deterministic	65.69(±1.34)	12.59(±1.32)	55.76(±1.21)	6.77(±1.20)
Probabilistic	75.96(±1.49)	26.03(±1.45)	75.43(±1.35)	17.90(±1.25)

mance of the probabilistic model is at least 10% better than the deterministic model in this experiment on both metrics and datasets. Therefore, the proposed probabilistic model is able to analyze entire videos and generate their global classification as accurately as it does with trimmed video clips.

6.3.3 Qualitative Evaluation

While Section 6.3.2 presents a quantitative evaluative of the proposed inference models, the current section presents a qualitative evaluation by visualizing the inference results of some interesting regular expressions. More specifically, we show examples of regular expressions and video clips that match with high probability by our proposed probabilistic model. Figures 6.7 and 6.8 show examples from the MultiTHUMOS [Yeung et al., 2017] dataset, while Figures 6.9 and 6.10 show examples from the Charades [Sigurdsson et al., 2016] dataset. We also discuss the characteristics of the expressions used for evaluation and failure cases of our model which are highlighted by a red frame in these figures.

As explained in Section 6.3.2, the expressions used in the action recognition experiments are mined with the intent of evaluating our model's prediction accuracy on compositional activities with concurrent, sequential and alternative primitive actions. The mined expressions follow the format defined by Equation 6.6, where concurrent, sequential and alternative actions are determined by the number of primitives per symbol $|w|$, number of sequential symbols n and number of alternative sequences d parameters, respectively. These expressions can be as simple as the expression in the first row in Figure 6.7 where we have ($|w| = 1$, $\mathbf{n} = \mathbf{3}$, $d = 1$), or as complex as the expression presented in the first row in Figure 6.10 where we have two alternative sub-expressions that has two and three sequential patterns of three or four concurrent actions. We evaluate the proposed inference models quantitatively on these mined regular expressions of different complexities in Section 6.3.2.

Figure 6.7: Examples of regular expressions and video clips mined from the MultiTHUMOS [Yeung et al., 2017] dataset and matched by the proposed probabilistic model. The primitive actions used to form these expressions are running (Run), jumping (Jump), falling (Fall), body rolling (Body-Roll), body bending (Body Bend), basketball dribbling (BaDr), and basketball dunking (BaDu). The video clips with red border are false positives. Best seen in color and zoomed in.

Figure 6.8: Examples of regular expressions and video clips mined from the MultiTHUMOS [Yeung et al., 2017] dataset and matched by the proposed probabilistic model. The primitive actions used to form these expressions are standing (Stand), throwing (Throw), golf swinging (Golf-Swing), clapping hands (Clap-Hands), body contraction (Body-Contract), squatting (Squat), sitting (Sit) , clean and jerk (Clean-And-Jerk), picking up (Pick-Up), and standing up (Stand-Up). The video clips with red border are false positives. Best seen in color and zoomed in.

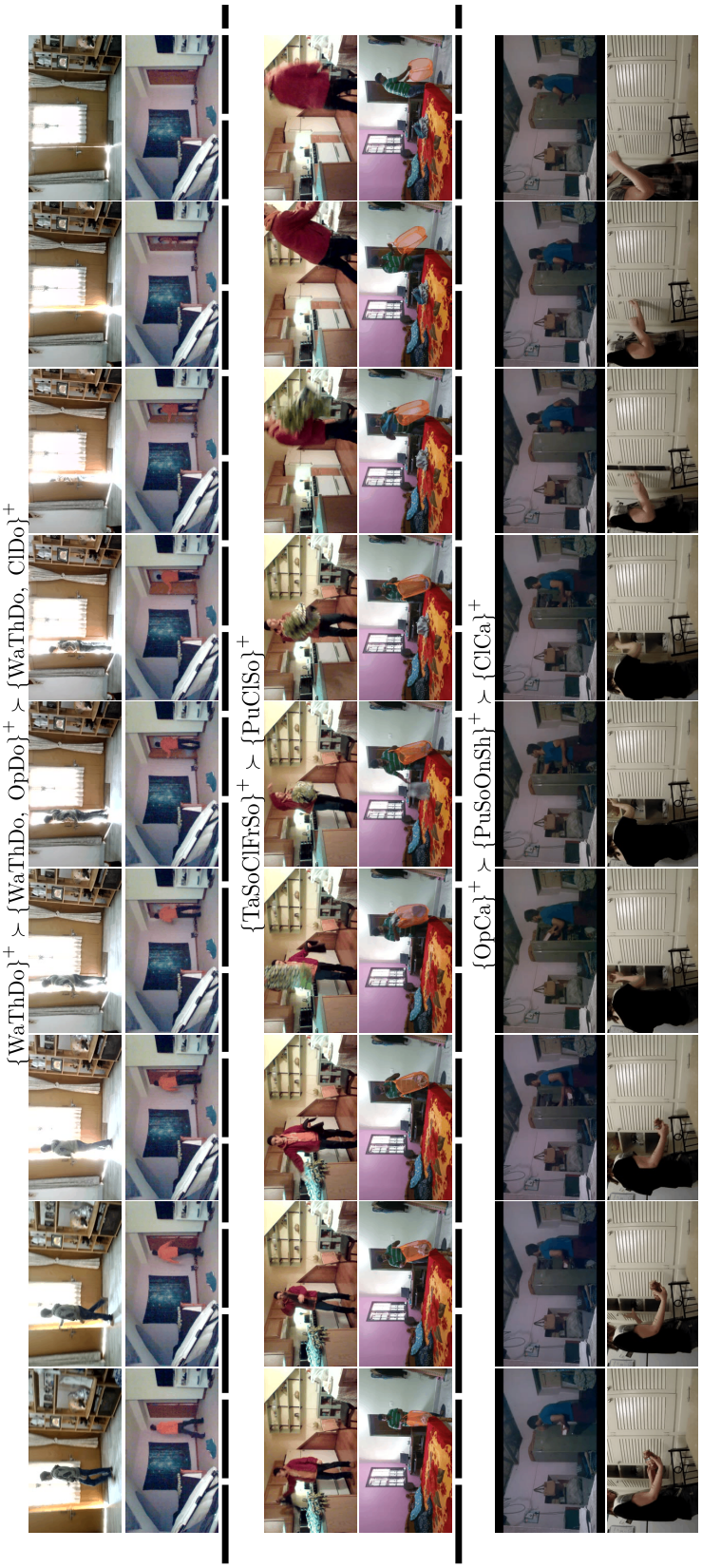


Figure 6.9: Examples of regular expressions and video clips mined from the Charades [Sigurdsson et al., 2016] dataset and matched by the proposed probabilistic model. The primitive actions used to form these regular expressions are walking through a doorway (WaThDo), opening a door (OpDo), closing a door (ClDo), taking some clothes from somewhere (TaSoClFrSo), putting clothes in somewhere (PuClSo), opening a cabinet (OpCa), putting something on a shelf (PuSoOnSh), and closing a cabinet (ClCa). The video clips with red border are false positives. Best seen in color and zoomed in.

Figure 6.10: Examples of regular expressions and video clips mined from the Charades [Sigurdsson et al., 2016] dataset and matched by the proposed probabilistic model. The primitive actions used to form these regular expressions are walking through a doorway (WaThDo), opening a door (OpDo), closing a door (ClDo), drinking from a glass (DrFrGl), sitting in a chair (SiInCh), someone going from standing to sitting (SoGoFrStToSi), holding a glass (HoGl), holding a broom (HoBr), tidying up with a broom (TyUpWiBr), tidying something on the floor (TySoOnFl), someone is sneezing (SoSn), grasping on the doorknob (GrOnDoKn), and holding a bag (HoBa). The video clips with red border are false positives. Best seen in color and zoomed in.

There are many challenges in recognizing compositional activities described by these regular expressions in videos. Some expressions can be very complex and small differences in the videos can produce different results. See the first row in Figure 6.8, the difference between the positive video and the negative video is only a few frames where one man throws a ball to another man who is about to perform a "Golf Swing". See also the third row in Figure 6.8, the given expression requires matching the action "Sit" which is performed by a person that is in the background of the scene. In the same fashion, videos can depict complex scenes where multiple activities are performed independently making the problem harder. For instance, in the basketball game depicted in the third row of Figure 6.7, some activities different to the one of interest are happening, like "Basketball Guard". There are also edited videos with multiple takes of different parts of the scene as in the second video in the second row in Figure 6.8, where two athletes are preparing to perform a "javelin throw" and supporters are "clapping hands".

Moreover, it is well known that action recognition datasets contain inconsistencies in their ground-truth annotations, which increases the difficulty of evaluating our proposed rich compositional activity recognition task. For instance, frames are not consistently annotated throughout the datasets, since there is no consensus about when a certain action starts or ends between the annotators. In order to illustrate this fact, consider the second row in Figure 6.9, both video clips are correct for the given expression according to the ground-truth, but they differ substantially since the actor collects the clothes and then puts them somewhere else in the first video, while the actor performs both actions simultaneously in the second video. Likewise, the primitive action "walking through a doorway" for some videos starts before the actor reaches the door (like in the first row in Figure 6.9), while for other videos it starts as soon as he opens the door. Similar observation can be done for other primitive actions.

Despite the aforementioned challenges, the proposed probabilistic model is able to accurately infer compositional activities in videos. We can see examples of expressions with sequential actions, concurrent actions, and mixing these two types of temporal relation between primitive actions. As an example of sequential actions, we point to the second row in Figure 6.7 where an athlete is performing a jump followed by body roll and body bend in a diving competition. Another example is the third row in Figure 6.9 where someone is opening a cabinet, putting something on a shelf and closing the cabinet. As an example of concurrent actions, we refer to the third row in Figure 6.10 where someone is walking through a doorway while holding a glass. For expressions with both concurrent and sequential actions, we point to either the first row in Figure 6.8 where someone performs a golf-swing just after receiving a ball thrown by someone else or the first row in Figure 6.9 where someone opens and closes a door to perform a "walking trough a doorway" action. We also show an example of groups of activities in the first row of Figure 6.10 where someone is either sitting in a chair while holding a glass and drinking from a glass or someone that is tidying up with a broom and suddenly sneezes.

In addition to the challenges already discussed, the inaccuracy of the primitive

classifiers is one of the main causes of errors produced by our model. We observed that the primitive action classifiers rely greatly on the context of the scene such as the background and objects present. As such, they tend to produce wrong predictions when different primitive actions are performed in the same context. For instance consider the false positive example shown in the third row in Figure 6.8, the second video has nothing related to the corresponding expression but the context depicted, e.g., gym room with a lot weights and boxes, is similar to ground-truth video clips like the first video. Similar observations can be done about the second and third rows in Figure 6.10. Such an issue is a compelling direction for future works.

6.4 Chapter Summary

While Chapter 3 proposes a learning framework to learn image rankers leveraging the structure in the visual output space, Chapter 4 extends such a framework providing a self-supervised approach to learn transferable features for object recognition tasks such as object classification, detection and segmentation. These chapters together describe an effective way to pretrain deep learning models in order to mitigate the need for large scale human annotated datasets in the target applications. Following the same goal of reducing the human supervision in visual recognition systems, Chapter 5 presents an approach to scale-up recognition systems beyond the number of annotated visual concepts providing a recognition systems able to recognize visual concepts without a single training samples by leveraging the compositionality of visual primitives. Similarly, the current chapter provides an approach to scale-up action recognition systems by leveraging existing action classifiers allowing to precisely represent and recognize complex activities in videos.

More specifically, this chapter addresses the problem of recognizing complex compositional activities in videos. Towards this end, we proposed to describe activities unambiguously as regular expressions of simple primitive actions and developed deterministic and probabilistic frameworks to recognize instances of these regular expressions in videos. Through a variety of controlled experiments using synthetic data, we showed that our probabilistic framework excels in this task even when using noisy primitive classifies. In the action recognition context, the proposed model was able to extend state-of-the-art action classifiers to vastly more complex activities without additional data annotation effort or large performance degradation.

Going forward, one compelling direction of investigation is to incorporate correlations, co-occurrences, and contextual information between primitive actions into the proposed inference framework. The main idea is to learn these factors from data aiming to eliminate semantically inconsistent predictions of the primitive classifiers, e.g., it is not possible having someone running and sleeping at the same time.

Chapter 7

Conclusion and Future Directions

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

Alan Turing, 1950

This thesis focuses on reducing the exhaustive and expensive human supervision required by the current state-of-the-art models for visual recognition. We also aim to tackle visual recognition in a more realistic scenario where the visual concepts are not defined a priori and we can not annotate large volumes of data for them. We accomplish these goals by exploring the structure, priors and regularities existent in the visual world. This final chapter summarizes the contributions of this thesis and discuss some open problems and challenges for future research.

7.1 Summary

In this dissertation, we acknowledge the advances in visual recognition accomplished by current deep learning models. However, we argue that the dependence of these fully supervised approaches on large scale human annotated datasets is the main obstacle to the goal of developing visual recognition systems as capable as the human visual system. As explained in Chapter 1, it is unrealistic to produce a dataset at the scale and richness of the visual world and even if it was possible, such an approach would be problematic due to artificial bias, inconsistencies and ambiguities in the curated data and learning process itself. Therefore, we propose methods that reduce the need for extensive human supervision by leveraging the structure in the visual world. We call this approach visual recognition from structured supervision and explore the inherent structure existent in the outputs, inputs, and models for visual recognition.

In order to contextualize our research with the scientific literature and provide the background necessary to understand this dissertation, we provide a background chapter in Chapter 2 before discuss the technical chapters (Chapters 3 – 6). More specifically, such a chapter defines a visual recognition problem, explains the current data-driven approach, and present state-of-the-art models for the applications

relevant for this thesis. In addition, it provides a concise literature review on existing works that also attempt to reduce the exhaustive human supervision in visual recognition models. In order to facilitate the presentation of our contributions, the technical chapters also contain sections with related work and background information that are only relevant to the chapter in discussion.

Once our research is contrasted against existing works in the literature review and background information presented, we start our technical development in Chapter 3. In this chapter, we propose the *visual permutation learning* framework as a generic formulation to learn structural concepts in ordered image sequences. Towards this end, we first formulate such a problem as the prediction of the permutation matrix that recovers the structure of the data from shuffled samples of it. Then, we leverage the geometry of permutation matrices and its continuous surrogates to prune unfeasible solutions for our learning and inference algorithms in order to accurately and efficiently solve such a problem. The proposed model can be efficiently learned via backpropagation and stochastic gradient descent in an end-to-end manner. In our experiments, we evaluate our proposed framework on different image ranking application (e.g., relative attributes and supervised learning-to-rank) outperforming existing works by a considerable margin using the same amount of annotated data.

In the same way the outputs of visual recognition systems are structured, the visual inputs like images and videos depict visual priors and regularities that are useful to solve computer vision tasks. In Chapter 4, we propose to use the spatial structure intrinsically existent in unlabelled images to learn image representations without human supervision. More specifically, we first define an auxiliary task resembling image jigsaw puzzles. Then, motivated by the effectiveness of the proposed visual permutation learning framework on image ranking applications, we hypothesize that such a model trained to solve this task also leans object features transferable for object recognition tasks. We validate this hypothesis on transfer learning experiments where we outperform baselines and contemporary self-supervised image representation learning algorithms in object recognition tasks such as object classification, detection and segmentation. It is important to highlight that this approach allows us to train large deep learning models on small datasets by performing a simple pretraining on a unlabelled image collection which are very easy to obtain.

While we present an effective way to alleviate the need for human supervision when training large scale machine learning models for visual recognition in Chapters 3 and 4, we introduce an strategy to scale-up recognition systems beyond a fixed and considerably small number visual concepts in Chapter 5. Towards this end, we leverage the structure in the model space to develop neural network modules that can synthesize classifiers for complex visual concepts described by boolean expressions of visual primitives even if we do not have a single training sample for such complex concepts. We name this framework as *neural algebra of classifiers* and simplifies its components using the well-known De Morgan's laws. In our experiments, the neural algebra of classifiers framework is more accurate than the baselines when composing classifiers for expressions of animals attributes on two well-known datasets and different classification metrics. It is important to note that this chapter presents an ef-

fective way to scale-up recognition systems to complex and dynamic scenarios where the concepts to be recognized can not be defined a priori or enumerated.

Another important contribution of this dissertation is how to use existing models to perform more expressive tasks without requiring additional human annotated data. In this context, we propose to recognize complex activities in videos from the prediction of simple action classifiers in Chapter 6. We first describe complex activities as regular expressions of simple primitive actions named *action patterns*, then we derive a probabilistic framework to efficiently recognize these action patterns in videos. The proposed approach allows us to unambiguously distinguish between fine-grained actions, retrieve very specific activity instances and recognize complex composite of actions that may not have a single training sample. Our experiments show that the proposed model is able to extend state-of-the-art action classifiers to vastly more complex activities without additional data annotation effort or large performance degradation.

In summary, the methods proposed in this thesis provide more accurate, extensible, and interpretable vision models using much less human supervision than black-box fully supervised deep learning approaches. We also tackle visual recognition in a more realistic scenario where the visual concepts are not defined a priori and we can not annotate large volumes of data for them. Therefore, this thesis presents a more feasible direction towards the development of visual recognition algorithms with the capabilities of the human visual system.

7.2 Open Problems and Future Directions

Our work has made progress towards the long-term goal of visual recognition by proposing methods to reduce the need for extensive human supervision and tackling more expressive and complex recognition problems. However, as suggested by the quote at the beginning of this chapter, there are still many questions left unanswered and plenty more work to be done. This section discusses some of the limitations of the proposed methods and suggests a number of possible directions for building on our work.

7.2.1 Visual Permutation Learning Beyond Static Images

In Chapter 3, we describe the visual permutation learning as a generic formulation to learn structural concepts in ordered image sequences and validate the utility of such a framework on image ranking applications. In Chapter 4, we extend this pool of applications by learning image representation without human supervision for object recognition tasks following a self-supervised approach. Note that these applications basically explore structural information within images, neglecting the richness of structural information and visual priors existent in video data which can also been seen as time ordered sequences of images. Therefore, one compelling direction is to evaluate the proposed visual permutation learning framework on tasks using video data.

Along these lines, our ideas described in Chapter 4 can be extended to action representation learning. Specifically, we propose to learn representations for actions from unlabeled collections of videos by exploiting the temporal coherence naturally present on video frames. In video domain, we can describe an action as a collection of salient movements coherently distributed in time and space. Then, we can create a pretext task where the objective is to recover an action sequence given its temporal artificially shuffled version. As before, we can train the proposed visual permutation learning framework on this task hypothesizing it should learn what are actions, what are their sub-actions and how those sub-actions happen through time in order to solve such a task. Finally, this knowledge can also be transferred to larger and more complex deep learning based models for action recognition tasks like activity classification and action detection. Note that this application is more appealing than the unsupervised learning of image representation described in Chapter 4, since labelling video data is much more difficult and time consuming than curating labelled image datasets.

Furthermore, the visual permutation learning framework seems intuitively a very effective way to capture and exploit the temporal coherence depicted in video frames. Such a structural information is very valuable in different applications and the proposed framework can be used with other models in subsequent research to tackle these problems. For instance, video summarization consists of distilling a raw video into a compact form without losing its main semantic information. Most of the techniques developed for this task focus on retaining the most important parts of the video neglecting the transitions between these parts. The proposed visual permutation learning framework can be extended to smooth these transitions providing video summaries more visually pleasant. In the same fashion, the problem of synthesizing new video frames in an existing video, either in-between existing frames (interpolation) or subsequent to them (extrapolation), can benefit from the temporal coherence leaned by the visual permutation learning framework. Specifically, in order to obtain realistic views, we need to be aware of the temporal coherence which can be captured by the visual permutation learning framework when permuting shuffled sequences of video frames. Therefore, we expect that by building on our visual permutation learning framework, many different applications using video data can have better solutions.

7.2.2 Compositional Models Beyond Classification

In the context of visual recognition, we can think of classification as a recognition task where we are only interested in identifying semantic concepts that are depicted in a given visual data. For instance, object classification consists of identifying which objects appears in an image, while action classification aims to identify which actions are happening in a video clip. According to these definitions, Chapter 5 proposes a compositional model for synthesize visual concept classifiers and evaluates this model on image classification tasks, whereas Chapter 6 proposes a compositional inference procedure and evaluates this procedure on activity classification tasks using

trimmed and untrimmed video clips. We argue that these proposed compositional models can be extended to even more challenging recognition tasks like detection and segmentation which the objective is to predict labels for image regions and pixels, respectively.

In the context of object recognition, there are recognition models in the literature that can be extended to become compositional models inheriting all advantages over traditional models explained in Chapter 5. More specifically, the Fast R-CNN and the Mask R-CNN models proposed respectively by Girshick [2015] and He et al. [2017] can be made compositional by replacing the layers responsible for the final predictions with the neural algebra of classifiers schema. This approach would allow us to detect and segment complex objects based on boolean expression of simple primitives like visual attributes.

Likewise, we can propose compositional models for activity detection by combining the inference procedure proposed in Chapter 6 and existing models for action proposals like [Yu and Yuan, 2015] and Escorcia et al. [2016]. However, the proposed inference procedure allows a more elegant solution to the problem of temporally localizing activity instances in videos. The accepting probability described in Equation 6.4 has a very particular behaviour of peaking when the target activity has happened in a video. This can be used as termination signal and the beginning of the activity can be computed by finding a peak when matching the video and the regular expression in the opposite direction (i.e., from the end to the beginning). Therefore, activities instances can be localized in time by analysing such a probability measure using adaptive thresholds.

7.2.3 Modelling Action Correlation, Cooccurrence, and Contextuality

In Chapter 6, we propose a probabilistic inference framework to recognize complex activities described by regular expressions of primitive actions in videos. During the derivation of the proposed inference procedure, we assume independence between primitive actions when defining the distribution over subsets of actions that may happen in a given frame as described in Equation 6.3. While such an assumption simplifies our model and allows us to leverage off-the-shelf accurate action classifiers like modern deep learning models (e.g., C3D [Tran et al., 2015] and I3D [Carreira and Zisserman, 2017]), it does not hold in the real-world since actions present correlations according to their meaning, context and similarity. For instance, "driving" and "walking" are antagonistic, "watching tv" and "eating a snack" often happens jointly, and "brushing teeth" is more likely to happen after someone "wake up" than "going to sleep". Therefore, we intend to explore these regularities in order to provide a more accurate model for activity recognition.

A very promising direction towards this goal is to use a Determinantal Point Process (DPP) [Macchi, 1975; Kulesza et al., 2012] to define a distribution over subsets of actions that may happen in a given frame. This probabilistic model is mathematically elegant, computationally efficient, and can capture high-order dependencies between primitive actions. Furthermore, DPPs are closed under conditioning [Kulesza

et al., 2012] allowing us to condition the probability of a subset of actions occurring in the current frame on the subset of actions occurred in the previous frame. Another promising direction is to extend the neural algebra of classifiers formulation described in Chapter 5 to the problem of activity recognition. More specifically, we can represent the primitive actions in a vector space, define neural network modules for every regular expression operator, and parse the input regular expression to a sequence of computations of these modules. Despite to be a more complex model, this formulation allows us to go beyond the Markov model provided by the DPP formulation. Therefore, as a future work, we propose to investigate these approaches aiming to provide a more accurate model for activity recognition.

7.3 Conclusion

This thesis has explored the challenge of visual recognition using minimal human supervision. Towards this end, we have leverage the structure existent in visual inputs, outputs, and models to propose methods to reduce the exhaustive human supervision required by the state-of-the-art models for visual recognition. We also have proposed novel solutions to learn unsupervised image representations for object recognition, to synthesize classifiers for visual concepts without annotated data, and to extend action classifiers for activity recognition. However, as discussed in the previous section, there is still much work to be done on improving these methods and exciting directions in which they can be extended. We hope that our work can provide the direction on which further research can stand and free researchers to explore problems for which massive labeled datasets do not exist.

Bibliography

Abdulnabi, A. H.; Wang, G.; Lu, J.; and Jia, K., 2015. Multi-task cnn model for attribute prediction. *IEEE Transactions on Multimedia*, 17, 11 (2015), 1949–1959. (cited on page 55)

Acuna, D.; Ling, H.; Kar, A.; and Fidler, S., 2018. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 33)

Adams, R. P. and Zemel, R. S., 2011. Ranking via Sinkhorn propagation. *arXiv preprint arXiv:1106.1925*, (2011). (cited on page 49)

Agrawal, P.; Carreira, J.; and Malik, J., 2015. Learning to see by moving. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 68)

Agrawal, P.; Nair, A. V.; Abbeel, P.; Malik, J.; and Levine, S., 2016. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 31)

Alcorn, M. A.; Li, Q.; Gong, Z.; Wang, C.; Mai, L.; Ku, W.-S.; and Nguyen, A., 2018. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. *arXiv preprint arXiv:1811.11553*, (2018). (cited on page 4)

Aljundi, R.; Chakravarty, P.; and Tuytelaars, T., 2017. Expert gate: Lifelong learning with a network of experts. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 28)

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D., 2016. Neural module networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 73)

Baldi, P. and Hornik, K., 1989. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2, 1 (1989), 53–58. (cited on page 28)

Ballan, L., 2018. Visual recognition in the deep learning era. URL: http://ssie.dei.unipd.it/past-editions/technical-program-2018/. (cited on page 14)

Baranes, A. and Oudeyer, P.-Y., 2013. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61, 1 (Jan 2013), 49–73. (cited on page 35)

Barbosa, I. B.; Cristani, M.; Caputo, B.; Rognhaugen, A.; and Theoharis, T., 2018. Looking beyond appearances: Synthetic training data for deep cnns in re-identification. *Journal of Computer Vision and Image Understanding (CVIU)*, 167 (2018), 50–62. (cited on page 32)

Bearman, A.; Russakovsky, O.; Ferrari, V.; and Fei-Fei, L., 2016a. What's the point: Semantic segmentation with point supervision. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 30)

Bearman, A. L.; Russakovsky, O.; Ferrari, V.; and Li, F., 2016b. What's the point: Semantic segmentation with point supervision. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 33)

Bilen, H.; Fernando, B.; Gavves, E.; and Vedaldi, A., 2017. Action recognition with dynamic image networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, (2017). (cited on page 89)

Birkhoff, G., 1946. Three observations on linear algebra. *Univ. Nac. Tucumán. Revista A*, 5 (1946), 147–151. (cited on page 42)

Bishop, C. M., 2006. *Pattern Recognition and Machine Learning*. Springer. (cited on pages 13 and 28)

Boiman, O. and Irani, M., 2007. Similarity by composition. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 74)

Bojanowski, P. and Joulin, A., 2017. Unsupervised learning by predicting noise. In *Proc. of the International Conference on Machine Learning (ICML)*. (cited on page 68)

Boole, G., 1854. *An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities*. Dover Publications. (cited on page 73)

Bousmalis, K.; Silberman, N.; Dohan, D.; Erhan, D.; and Krishnan, D., 2017. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 32)

Boyd, S. and Vandenberghe, L., 2004. *Convex Optimization*. Cambridge University Press. (cited on pages 15 and 43)

Branson, S.; Beijbom, O.; and Belongie, S., 2013. Efficient large-scale structured learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 40)

Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; and Shah, R., 1994. Signature verification using a" siamese" time delay neural network. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 20)

Brown, B. J.; Toler-Franklin, C.; Nehab, D.; Burns, M.; Dobkin, D.; Vlachopoulos, A.; Doumas, C.; Rusinkiewicz, S.; and Weyrich, T., 2008. A system for high-volume acquisition and matching of fresco fragments: Reassembling theran wall paintings. In *ACM Transactions on Graphics (TOG)*. (cited on pages 38 and 39)

Brualdi, R. A., 1988. Some applications of doubly stochastic matrices. *Linear Algebra and its Applications*, 107 (1988), 77 – 100. (cited on page 42)

Bucher, M.; Herbin, S.; and Jurie, F., 2016. Improving semantic embedding consistency by metric learning for zero-shot classiffication. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 34)

Burges, C.; Shaked, T.; Renshaw, E.; Lazier, A.; Deeds, M.; Hamilton, N.; and Hullender, G. N., 2005. Learning to rank using gradient descent. In *Proc. of the International Conference on Machine Learning (ICML)*. (cited on page 27)

Burnyeat, M. et al., 1990. *The Theaetetus of Plato*. Hackett Publishing. (cited on page 73)

Caba, F.; Escorcia, V.; Ghanem, B.; and Niebles, J., 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 16 and 33)

Caba Heilbron, F.; Carlos Niebles, J.; and Ghanem, B., 2016. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 26)

Cao, Z.; Qin, T.; Liu, T.-Y.; Tsai, M.-F.; and Li, H., 2007. Learning to rank: from pairwise approach to listwise approach. In *Proc. of the International Conference on Machine Learning (ICML)*. (cited on pages 37 and 58)

Carreira, J. and Zisserman, A., 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 24, 30, 35, 89, 99, 102, and 115)

Castrejón, L.; Kundu, K.; Urtasun, R.; and Fidler, S., 2017. Annotating object instances with a polygon-rnn. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 33)

Chao, Y.-W.; Vijayanarasimhan, S.; Seybold, B.; Ross, D. A.; Deng, J.; and Sukthankar, R., 2018. Rethinking the faster r-cnn architecture for temporal action localization. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 26)

Chapelle, O.; Scholkopf, B.; and Zien, A., 2006. *Semi-Supervised Learning*. MIT Press. (cited on page 29)

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L., 2018. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 40, 4 (2018), 834–848. (cited on page 23)

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, (2017). (cited on page 23)

Chen, X. and Gupta, A., 2015. Webly supervised learning of convolutional networks. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 31)

Chen, X.; Shrivastava, A.; and Gupta, A., 2013. Neil: Extracting visual knowledge from web data. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 31)

Cheng, J.-Z.; Ni, D.; Chou, Y.-H.; Qin, J.; Tiu, C.-M.; Chang, Y.-C.; Huang, C.-S.; Shen, D.; and Chen, C.-M., 2016. Computer-aided diagnosis with deep learning architecture: applications to breast lesions in us images and pulmonary nodules in ct scans. *Scientific reports*, 6 (2016), 24454. (cited on page 1)

Cherian, A.; Fernando, B.; Harandi, M.; and Gould, S., 2017. Generalized rank pooling for activity recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 25)

Cho, T. S.; Avidan, S.; and Freeman, W. T., 2010. The patch transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 32, 8 (2010). (cited on page 39)

Chopra, S.; Hadsell, R.; and LeCun, Y., 2005. Learning a similarity metric discriminatively, with application to face verification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 20)

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS Deep Learning Workshop*, (2014). (cited on page 73)

Ciresan, D.; Giusti, A.; Gambardella, L. M.; and Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 22)

Collins, B.; Deng, J.; Li, K.; and Fei-Fei, L., 2008. Towards scalable dataset construction: An active learning approach. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 35)

Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Machine learning*, 20, 3 (1995), 273–297. (cited on page 15)

Cossock, D. and Zhang, T., 2006. Subset ranking using regression. In *International Conference on Computational Learning Theory*. (cited on page 27)

Crammer, K. and Singer, Y., 2002. Pranking with ranking. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 26)

Croitoru, I.; Bogolin, S.-V.; and Leordeanu, M., 2019. Unsupervised learning of foreground object segmentation. *International Journal of Computer Vision (IJCV)*, (2019), 1–24. (cited on page 33)

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2, 4 (1989), 303–314. (cited on page 16)

Dai, X.; Singh, B.; Zhang, G.; Davis, L. S.; and Qiu Chen, Y., 2017. Temporal context network for activity localization in videos. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 26)

Dalal, N. and Triggs, B., 2005. Histograms of oriented gradients for human detection. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 1 and 15)

de Sa, V. R. and Ballard, D. H., 1993. Self-teaching through correlated input. In *Computation and neural systems*, 437–441. Springer. (cited on page 31)

Dean, J., 2017. Trends and developments in deep learning research. url: https://www.youtube.com/watch?v=jCB_z7SDo1c. (cited on page 2)

Deng, J.; Dong, W.; Socher, R.; Li, L. J.; Li, K.; and Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 16)

Diamond, S. and Boyd, S., 2016. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research (JMLR)*, 17, 83 (2016), 1–5. (cited on page 50)

DiCarlo, J. J.; Zoccolan, D.; and Rust, N. C., 2012. How does the brain solve visual object recognition? *Neuron*, 73, 3 (2012), 415–434. (cited on page 1)

Divvala, S. K.; Farhadi, A.; and Guestrin, C., 2014. Learning everything about anything: Webly-supervised visual concept learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3270–3277. (cited on page 31)

Doersch, C., 2016. *Supervision Beyond Manual Annotations for Learning Visual Representations*. Ph.D. thesis, Machine Learning Department School of Computer Science at Carnegie Mellon University. (cited on pages 4 and 66)

Doersch, C.; Gupta, A.; and Efros, A. A., 2015. Unsupervised visual representation learning by context prediction. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages 31, 62, 63, 64, 67, 68, and 73)

Dollár, P.; Appel, R.; Belongie, S.; and Perona, P., 2014. Fast feature pyramids for object detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 36, 8 (2014), 1532–1545. (cited on pages 1, 15, and 20)

Domke, J., 2012. Generic methods for optimization-based modeling. In *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. (cited on page 48)

Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T., 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 25 and 89)

Donahue, J.; Krähenbühl, P.; and Darrell, T., 2017. Adversarial feature learning. In *Proc. of the International Conference on Learning Representations (ICLR)*. (cited on pages 63, 67, and 68)

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; and Brox, T., 2015. Flownet: Learning optical flow with convolutional networks. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 32)

Dwibedi, D.; Misra, I.; and Hebert, M., 2017. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 33)

Engilberge, M.; Chevallier, L.; Perez, P.; and Cord, M., 2019. Sodeep: a sorting deep net to learn ranking loss surrogates. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 27)

Escorcia, V.; Heilbron, F. C.; Niebles, J. C.; and Ghanem, B., 2016. Daps: Deep action proposals for action understanding. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on pages 26 and 115)

Everingham, M.; Eslami, S. M.; Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A., 2015. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 111, 1 (2015), 98–136. (cited on page 2)

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A., 2007. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. (cited on pages 39, 53, 54, 63, and 68)

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A., 2012. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. (cited on pages 39, 53, 54, 63, and 68)

Facebook Inc., 2013. A focus on efficiency: A whitepaper from facebook, ericsson and qualcomm. Technical report, Internet org. (cited on page 3)

Faktor, A. and Irani, M., 2012. "clustering by composition"–unsupervised discovery of image categories. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 74)

Faktor, A. and Irani, M., 2013. Co-segmentation by composition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 74)

Fanello, S. R.; Keskin, C.; Izadi, S.; Kohli, P.; Kim, D.; Sweeney, D.; Criminisi, A.; Shotton, J.; Kang, S. B.; and Paek, T., 2014. Learning to be a depth camera for close-range human capture and interaction. In *Proc. of the Intl. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*. (cited on page 32)

Farhadi, A.; Hejrati, M.; Sadeghi, M. A.; Young, P.; Rashtchian, C.; Hockenmaier, J.; and Forsyth, D., 2010. Every picture tells a story: Generating sentences from images. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 40)

Faugeras, O., 1993. *Three-dimensional Computer Vision: A Geometric Viewpoint*. MIT Press. (cited on pages 45 and 48)

Feichtenhofer, C.; Pinz, A.; and Zisserman, A., 2016. Convolutional two-stream network fusion for video action recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 25)

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; and Ramanan, D., 2010. Object detection with discriminatively trained part-based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 32, 9 (2010), 1627–1645. (cited on pages 1, 20, and 73)

Fergus, R.; Fei-Fei, L.; Perona, P.; and Zisserman, A., 2010. Learning object categories from internet image searches. *Proceedings of the IEEE*, 98, 8 (2010), 1453–1466. (cited on page 31)

Fernando, B.; Anderson, P.; Hutter, M.; and Gould, S., 2016. Discriminative hierarchical rank pooling for activity recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 25)

Fernando, B.; Bilen, H.; Gavves, E.; and Gould, S., 2017. Self-supervised video representation learning with odd-one-out networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 62, 63, and 73)

Fernando, B.; Gavves, E.; Muselet, D.; and Tuytelaars, T., 2015a. Learning-to-rank based on subsequences. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages 28 and 58)

Fernando, B.; Gavves, E.; Oramas, J. M.; Ghodrati, A.; and Tuytelaars, T., 2015b. Modeling video evolution for action recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 25)

Fernando, B. and Gould, S., 2016. Learning end-to-end video classification with rank-pooling. In *Proc. of the International Conference on Machine Learning (ICML)*. (cited on pages 25, 44, and 48)

Frank, M.; Leitner, J.; Stollenga, M. F.; Förster, A.; and Schmidhuber, J., 2013. Curiosity driven reinforcement learning for motion planning on humanoids. In *Frontiers in neurorobotics*. (cited on page 35)

Frege, G., 1948. Sense and reference. *The Philosophical Review*, 57, 3 (1948), 209–230. (cited on page 73)

Freund, Y.; Schapire, R.; and Abe, N., 1999. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14, 771-780 (1999), 1612. (cited on page 15)

Friedman, J.; Hastie, T.; and Tibshirani, R., 2001. *The elements of statistical learning*, vol. 1. Springer series in statistics New York. (cited on page 13)

Frome, A.; Corrado, G. S.; Shlens, J.; Bengio, S.; Dean, J.; Mikolov, T.; et al., 2013. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 74)

Gaidon, A.; Wang, Q.; Cabon, Y.; and Vig, E., 2016. Virtual worlds as proxy for multi-object tracking analysis. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 32)

Gan, C.; Lin, M.; Yang, Y.; de Melo, G.; and Hauptmann, A. G., 2016. Concepts not alone: Exploring pairwise relationships for zero-shot video activity recognition. In *Proc. of the AAAI Conference on Artificial Intelligence (AAAI)*. (cited on page 34)

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R., 2017. Tall: Temporal activity localization via language query. *Proc. of the International Conference on Computer Vision (ICCV)*, (2017). (cited on pages 88 and 90)

Gavrilyuk, K.; Ghodrati, A.; Li, Z.; and Snoek, C. G., 2018. Actor and action video segmentation from a sentence. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 88 and 90)

Ghosh, R., 2018. Deep learning for videos: A 2018 guide to action recognition. url: http://blog.qure.ai/notes/deep-learning-for-videos-action-recognition-review. (cited on page 24)

Gidaris, S.; Singh, P.; and Komodakis, N., 2018. Unsupervised representation learning by predicting image rotations. In *Proc. of the International Conference on Learning Representations (ICLR)*. (cited on pages 62, 63, and 68)

Girdhar, R.; Ramanan, D.; Gupta, A.; Sivic, J.; and Russell, B., 2017. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 25)

Girshick, R., 2015. Fast R-CNN. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages xiii, 20, 21, 69, and 115)

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 20 and 63)

Girshick, R. B.; Felzenszwalb, P. F.; and Mcallester, D. A., 2011. Object detection with grammar models. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 73)

Gkioxari, G. and Malik, J., 2015. Finding action tubes. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 26)

Glorot, X. and Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. (cited on page 50)

Golge, E. and Duygulu, P., 2014. Conceptmap: Mining noisy web data for concept learning. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 31)

Goodale, M. A. and Milner, A. D., 1992. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15, 1 (1992), 20–25. (cited on page 24)

Gould, S.; Fernando, B.; Cherian, A.; Anderson, P.; Santa Cruz, R.; and Guo, E., 2016. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, (2016). (cited on pages 9, 45, and 48)

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D., 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 2)

Grabar, H., 2018. We now know why the self-driving uber that killed a pedestrian didn't brake. *Future Tense*, (2018). https://slate.com/technology/2018/05/uber-car-in-fatal-arizona-crash-perceived-pedestrian-1-3-seconds-before-impact.html. Access date: 11-12-2018. (cited on page 4)

Gupta, A.; Vedaldi, A.; and Zisserman, A., 2016. Synthetic data for text localisation in natural images. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 33)

Gurobi Optimization, I., 2016. Gurobi optimizer reference manual. http://www.gurobi.com. (cited on page 44)

Gygli, M.; Grabner, H.; Riemenschneider, H.; Nater, F.; and Van Gool, L., 2013. The interestingness of images. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages 39, 53, and 58)

Habibian, A.; Mensink, T.; and Snoek, C. G., 2017. Video2vec embeddings recognize events when examples are scarce. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 39, 10 (2017), 2089–2103. (cited on pages 34 and 90)

Haeusser, P.; Mordvintsev, A.; and Cremers, D., 2017. Learning by association-a versatile semi-supervised training method for neural networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 29)

Haykin, S. S.; Haykin, S. S.; Haykin, S. S.; and Haykin, S. S., 2009. *Neural networks and learning machines*, vol. 3. Pearson Upper Saddle River, NJ, USA:. (cited on page 77)

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R., 2017. Mask r-cnn. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages 23 and 115)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 37, 9 (2015), 1904–1916. (cited on page 20)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2016. Deep residual learning for image recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. (cited on pages 2, 18, and 19)

Hendricks, L. A.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B., 2017. Localizing moments in video with natural language. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages 34, 88, and 90)

Herath, S.; Harandi, M.; and Porikli, F., 2017. Going deeper into action recognition: A survey. *Image and vision computing*, 60 (2017), 4–21. (cited on pages 23 and 87)

Herbrich, R.; Graepel, T.; and Obermayer, K., 2000. Large margin rank boundaries for ordinal regression. *MIT Press, Cambridge, MA*, (2000). (cited on page 27)

Hinton, G. E. and Zemel, R. S., 1994. Autoencoders, minimum description length and helmholtz free energy. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 28)

Hochreiter, S.; Bengio, Y.; Frasconi, P.; Schmidhuber, J.; et al., 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. (cited on page 18)

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural Computation*, 9, 8 (1997), 1735–1780. (cited on page 73)

Hoffman, J.; Gupta, S.; and Darrell, T., 2016. Learning with side information through modality hallucination. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 34)

Hopcroft, J., 1971. An n log n algorithm for minimizing states in a finite automaton. *Theory of machines and computations*, (1971), 189–196. (cited on page 92)

Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4, 2 (1991), 251–257. (cited on page 16)

Houlsby, N.; Huszar, F.; Ghahramani, Z.; and Lengyel, M., 2011. Bayesian active learning for classification and preference learning. *arXiv preprint arxiv:1112.5745*, (2011). (cited on page 35)

Hu, R.; Andreas, J.; Rohrbach, M.; Darrell, T.; and Saenko, K., 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 73)

Hu, R.; Dollár, P.; He, K.; Darrell, T.; and Girshick, R., 2018. Learning to segment every thing. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 30)

Hu, R.; Rohrbach, M.; and Darrell, T., 2016a. Segmentation from natural language expressions. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on pages 34 and 90)

Hu, R.; Xu, H.; Rohrbach, M.; Feng, J.; Saenko, K.; and Darrell, T., 2016b. Natural language object retrieval. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 90)

Huang, C.; Change Loy, C.; and Tang, X., 2016. Unsupervised learning of discriminative attributes and visual representations. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 40)

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q., 2017. Densely connected convolutional networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 18)

İkizler, N. and Forsyth, D. A., 2008. Searching for complex human activities with no visual examples. *International Journal of Computer Vision (IJCV)*, 80, 3 (2008), 337–357. (cited on page 90)

Ilie, L. and Yu, S., 2002. Constructing nfas by optimal use of positions in regular expressions. In *Annual Symposium on Combinatorial Pattern Matching*. (cited on page 92)

Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of the International Conference on Machine Learning (ICML)*. (cited on page 20)

Isola, P.; Zoran, D.; Krishnan, D.; and Adelson, E. H., 2016. Learning visual groups from co-occurrences in space and time. *ICLR Workshop*, (2016). (cited on page 63)

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al., 2015. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 41)

Jain, M.; van Gemert, J. C.; Mensink, T.; and Snoek, C. G., 2015. Objects2action: Classifying and localizing actions without any video example. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages 88 and 90)

Jayaraman, D. and Grauman, K., 2015. Learning image representations tied to ego-motion. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 31)

Jenni, S. and Favaro, P., 2018. Self-supervised feature learning by learning to spot artifacts. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 68)

Ji, S.; Xu, W.; Yang, M.; and Yu, K., 2010. 3d convolutional neural networks for human action recognition. In *Proc. of the International Conference on Machine Learning (ICML)*. (cited on page 24)

Joachims, T., 2006. Training linear svms in linear time. In *Proc. of the ACM Conf. on Knowledge Discovery and Data Mining (SIGKDD)*. (cited on pages 40 and 58)

Joshi, A. J.; Porikli, F.; and Papanikolopoulos, N., 2009. Multi-class active learning for image classification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 35)

Kalogeiton, V.; Weinzaepfel, P.; Ferrari, V.; and Schmid, C., 2017. Action tubelet detector for spatio-temporal action localization. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 26)

Kang, S. M. and Wildes, R. P., 2016. Review of action recognition and detection methods. *arXiv preprint arXiv:1610.06906*, (2016). (cited on pages 23 and 87)

Kim, D.; Cho, D.; Yoo, D.; and Kweon, I. S., 2018. Learning image representations by completing damaged jigsaw puzzles. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*. (cited on page 68)

Knight, P. A., 2008. The sinkhorn-knopp algorithm: convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30, 1 (2008), 261–275. (cited on page 42)

Kovashka, A.; Parikh, D.; and Grauman, K., 2012. WhittleSearch: Image Search with Relative Attribute Feedback. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 40)

Krähenbühl, P. and Koltun, V., 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 23)

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 1097–1105. (cited on pages 2, 17, 19, 47, 53, 54, 56, 68, and 69)

Kulesza, A.; Taskar, B.; et al., 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5, 2–3 (2012), 123–286. (cited on page 115)

Kulkarni, T. D.; Narasimhan, K.; Saeedi, A.; and Tenenbaum, J., 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 35)

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the International Conference on Machine Learning (ICML)*. (cited on page 37)

Lambert, F., 2016. Understanding the fatal tesla accident on autopilot and the nhtsa probe. *Electrek, July*, (2016). (cited on page 4)

Lampert, C. H.; Nickisch, H.; and Harmeling, S., 2009. Learning to detect unseen object classes by between-class attribute transfer. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 34 and 74)

Lampert, C. H.; Nickisch, H.; and Harmeling, S., 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 36, 3 (2014), 453–465. (cited on pages 40 and 90)

Larsson, G.; Maire, M.; and Shakhnarovich, G., 2016. Learning representations for automatic colorization. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 31)

Larsson, G.; Maire, M.; and Shakhnarovich, G., 2017. Colorization as a proxy task for visual understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 67 and 68)

Lawson, M. V., 2003. *Finite automata*. Chapman and Hall/CRC. (cited on page 92)

LeCun, Y.; Bengio, Y.; and Hinton, G., 2015. Deep learning. *Nature*, 521, 7553 (2015), 436. (cited on page 2)

Lee, D.-H., 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*. (cited on page 29)

Lee, H.-Y.; Huang, J.-B.; Singh, M.; and Yang, M.-H., 2017. Unsupervised representation learning by sorting sequences. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages 51, 62, 64, 66, and 68)

Lee, Y. J.; Efros, A. A.; and Hebert, M., 2013. Style-aware mid-level representation for discovering visual connections in space and time. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages 39, 53, 54, and 58)

Lei Ba, J.; Swersky, K.; Fidler, S.; et al., 2015. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 34 and 74)

Li, F.-F.; Johnson, J.; and Yeung, S., 2019. Cs231n: Convolutional neural networks for visual recognition. url: http://cs231n.stanford.edu/. (cited on page 15)

Li, H.; Wu, H.; Li, D.; Lin, S.; Su, Z.; and Luo, X., 2018. Psi: A probabilistic semantic interpretable framework for fine-grained image ranking. *Journal of the Association for Information Science and Technology*, 69, 12 (2018), 1488–1501. (cited on page 27)

Li, L.-J. and Fei-Fei, L., 2010. Optimol: automatic online picture collection via incremental model learning. *International Journal of Computer Vision (IJCV)*, 88, 2 (2010), 147–168. (cited on page 31)

Li, P.; Wu, Q.; and Burges, C. J., 2008. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 27)

Li, S.; Shan, S.; and Chen, X., 2012. Relative forest for attribute prediction. In *Proc. of the Asian Conference on Computer Vision (ACCV)*. (cited on page 56)

Li, Z.; Tao, R.; Gavves, E.; Snoek, C. G.; Smeulders, A. W.; et al., 2017. Tracking by natural language specification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 34 and 90)

Liang, L. and Grauman, K., 2014. Beyond comparing image pairs: Setwise active learning for relative attributes. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 26)

Lin, M., Tsung-Yiand Maire; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L., 2014. Microsoft COCO: Common objects in context. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on pages 16, 33, and 65)

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S., 2017. Feature pyramid networks for object detection. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 23)

Liu, B.; Yeung, S.; Chou, E.; Huang, D.-A.; Fei-Fei, L.; and Niebles, J. C., 2018. Temporal modular networks for retrieving complex compositional activities in videos. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on pages 88 and 90)

Liu, J.; Kuipers, B.; and Savarese, S., 2011. Recognizing human actions by attributes. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 90)

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C., 2016. SSD: Single shot multibox detector. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 21)

Long, J.; Shelhamer, E.; and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 22, 63, and 69)

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60, 2 (Nov 2004), 91–110. (cited on pages 1 and 15)

Macchi, O., 1975. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7, 1 (1975), 83–122. (cited on page 115)

Mahajan, D.; Girshick, R.; Ramanathan, V.; He, K.; Paluri, M.; Li, Y.; Bharambe, A.; and van der Maaten, L., 2018. Exploring the limits of weakly supervised pretraining. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 32)

Marande, W. and Burger, G., 2007. Mitochondrial dna as a genomic jigsaw puzzle. *Science*, 318, 5849 (2007). (cited on page 39)

Marr, D. and Nishihara, H. K., 1978. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 200, 1140 (1978), 269–294. (cited on page 1)

Marszalek, M.; Laptev, I.; and Schmid, C., 2009. Actions in context. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 61)

McCulloch, W. S. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 4 (1943), 115–133. (cited on page 92)

Mettes, P. and Snoek, C. G., 2017. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 88)

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J., 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, (2013). (cited on page 31)

Miller, G. A., 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38, 11 (1995), 39–41. (cited on page 3)

Minsky, M., 1988. *Society of mind*. Simon and Schuster. (cited on page 1)

Misra, I., 2018. *Visual Learning with Minimal Human Supervision*. Ph.D. thesis, The Robotics Institute at Carnegie Mellon University. (cited on page 4)

Misra, I.; Girshick, R.; Fergus, R.; Hebert, M.; Gupta, A.; and van der Maaten, L., 2018. Learning by asking questions. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 35)

Misra, I.; Gupta, A.; and Hebert, M., 2017. From red wine to red tomato: Composition with context. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 6, 72, 74, and 81)

Misra, I.; Zitnick, C. L.; and Hebert, M., 2016. Shuffle and learn: Unsupervised learning using temporal order verification. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on pages 62, 64, and 66)

Mitkov, R., 2003. *The Oxford Handbook of Computational Linguistics (Oxford Handbooks in Linguistics S.)*. Oxford University Press, Inc., New York, NY, USA. ISBN 0198238827. (cited on page 92)

Mohapatra, P.; Rolínek, M.; Jawahar, C.; Kolmogorov, V.; and Pawan Kumar, M., 2018. Efficient optimization for rank-based loss functions. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 27)

Monk, J. and Bonnet, R., 1989. *Handbook of Boolean algebras*. No. v. 2 in Handbook of Boolean Algebras. North-Holland. (cited on page 82)

Mottaghi, R.; Chen, X.; Liu, X.; Cho, N. G.; Lee, S. W.; Fidler, S.; Urtasun, R.; and Yuille, A., 2014. The role of context for object detection and semantic segmentation in the wild. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 61)

Movshovitz-Attias, Y.; Kanade, T.; and Sheikh, Y., 2016. How useful is photo-realistic rendering for visual learning? In *ECCV 2016 Workshops*. (cited on page 32)

Murphy, K. P., 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press. (cited on page 13)

Nathan Mundhenk, T.; Ho, D.; and Chen, B. Y., 2018. Improvements to context based self-supervised learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 68)

Neelakantan, A.; Le, Q. V.; and Sutskever, I., 2016. Neural programmer: Inducing latent programs with gradient descent. In *Proc. of the International Conference on Learning Representations (ICLR)*. (cited on page 74)

Ni, B.; Yang, X.; and Gao, S., 2016. Progressively parsing interactional objects for fine grained action detection. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 26)

Nickolls, J.; Buck, I.; Garland, M.; and Skadron, K., 2008. Scalable parallel programming with cuda. In *ACM SIGGRAPH 2008 classes*. (cited on page 16)

Nie, D.; Zhang, H.; Adeli, E.; Liu, L.; and Shen, D., 2016. 3d deep learning for multi-modal imaging-guided survival time prediction of brain tumor patients. In *Proc. of the Intl. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 212–220. Springer. (cited on page 1)

Niu, L.; Veeraraghavan, A.; and Sabharwal, A., 2018. Webly supervised learning meets zero-shot learning: A hybrid approach for fine-grained classification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 34)

Noroozi, M. and Favaro, P., 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on pages 31, 51, 62, 63, 67, 68, and 69)

Noroozi, M.; Pirsiavash, H.; and Favaro, P., 2017. Representation learning by learning to count. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages 62, 63, and 68)

Ochs, P.; Ranftl, R.; Brox, T.; and Pock, T., 2015. Bilevel optimization with nonsmooth lower level problems. In *International Conference on Scale Space and Variational Methods in Computer Vision*. (cited on pages 44 and 48)

Oliver, A.; Odena, A.; Raffel, C.; Cubuk, E. D.; and Goodfellow, I. J., 2018. Realistic evaluation of deep semi-supervised learning algorithms. *arXiv preprint arXiv:1804.09170*, (2018). (cited on page 29)

Owens, A.; Wu, J.; McDermott, J. H.; Freeman, W. T.; and Torralba, A., 2016. Ambient sound provides supervision for visual learning. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on pages 34 and 68)

Palatucci, M.; Pomerleau, D.; Hinton, G. E.; and Mitchell, T. M., 2009. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 74)

Papadopoulos, D. P.; Clarke, A. D. F.; Keller, F.; and Ferrari, V., 2014. Training object class detectors from eye tracking data. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 33)

Papadopoulos, D. P.; Uijlings, J. R. R.; Keller, F.; and Ferrari, V., 2016. We don't need no bounding-boxes: Training object class detectors using only human verification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 33)

Papadopoulos, D. P.; Uijlings, J. R. R.; Keller, F.; and Ferrari, V., 2017. Training object class detectors with click supervision. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 33)

Parikh, D. and Grauman, K., 2011. Relative attributes. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 26, 39, 40, 53, 54, 55, and 56)

Pathak, D.; Girshick, R.; Dollár, P.; Darrell, T.; and Hariharan, B., 2017. Learning features by watching objects move. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 31, 63, and 68)

Pathak, D.; Krähenbühl, P.; Donahue, J.; Darrell, T.; and Efros, A., 2016. Context encoders: Feature learning by inpainting. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 63, 67, and 68)

Pathak, D.; Shelhamer, E.; Long, J.; and Darrell, T., 2014. Fully convolutional multi-class multiple instance learning. *Proc. of the International Conference on Learning Representations (ICLR)*, (2014). (cited on pages 29 and 30)

Peng, X.; Sun, B.; Ali, K.; and Saenko, K., 2015. Learning deep object detectors from 3d models. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 32)

Piergiovanni, A. and Ryoo, M. S., 2018. Learning latent super-events to detect multiple activities in videos. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 90)

Pinggera, P.; Ramos, S.; Gehrig, S.; Franke, U.; Rother, C.; and Mester, R., 2016. Lost and found: detecting small road hazards for self-driving vehicles. In *Proc. of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 1099–1106. IEEE. (cited on page 1)

Platt, J. et al., 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10, 3 (1999), 61–74. (cited on page 80)

Prince, S. J., 2012. *Computer vision: models, learning, and inference*. Cambridge University Press. (cited on page 13)

Qiu, Z.; Yao, T.; and Mei, T., 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 24)

Rabin, M. O., 1963. Probabilistic automata. *Information and control*, 6, 3 (1963), 230–245. (cited on page 93)

Rabin, M. O. and Scott, D., 1959. Finite automata and their decision problems. *IBM journal of research and development*, 3, 2 (1959), 114–125. (cited on page 92)

Rahmani, H. and Mian, A., 2016. 3d action recognition from novel viewpoints. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 32)

Rasmus, A.; Berglund, M.; Honkala, M.; Valpola, H.; and Raiko, T., 2015. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 29)

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages xiii and 21)

Redmon, J. and Farhadi, A., 2017. Yolo9000: better, faster, stronger. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 21)

Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, (2018). (cited on page 21)

Ren, S.; He, K.; Girshick, R.; and Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on pages 20 and 35)

Ren, Z. and Jae Lee, Y., 2018. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 67 and 68)

Richard, A. and Gall, J., 2016. Temporal action detection using a statistical language model. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 90)

Richard, A.; Kuehne, H.; and Gall, J., 2018a. Action sets: Weakly supervised action segmentation without ordering constraints. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 29 and 30)

Richard, A.; Kuehne, H.; Iqbal, A.; and Gall, J., 2018b. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 29 and 30)

Roberto de Souza, C.; Gaidon, A.; Cabon, Y.; and Manuel Lopez, A., 2017. Procedural generation of videos to train deep action recognition networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 33)

Ronneberger, O.; Fischer, P.; and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In *Proc. of the Intl. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. (cited on page 22)

Rosenblatt, F., 1961. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, CORNELL AERONAUTICAL LAB INC BUFFALO NY. (cited on page 16)

Rumelhart, D. E.; Hinton, G. E.; Williams, R. J.; et al., 1988. Learning representations by back-propagating errors. *Cognitive modeling*, 5, 3 (1988), 1. (cited on page 15)

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al., 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115, 3 (2015), 211–252. (cited on pages 2, 65, and 78)

Russell, S. J. and Norvig, P., 2016. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,. (cited on page 1)

Sadeghi, F. and Levine, S., 2016. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, (2016). (cited on page 33)

Sajjadi, M.; Javanmardi, M.; and Tasdizen, T., 2016. Mutual exclusivity loss for semi-supervised deep learning. In *Proc. of the International Conference on Image Processing (ICIP)*. (cited on page 29)

Santa Cruz, R.; Campbell, D.; Fernando, B.; Cherian, A.; and Gould, S., 2019. Inferring rich compositional activities in videos. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 10)

Santa Cruz, R.; Fernando, B.; Cherian, A.; and Gould, S., 2017. Deeppermnet: Visual permutation learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 9 and 73)

Santa Cruz, R.; Fernando, B.; Cherian, A.; and Gould, S., 2018a. Neural algebra of classifiers. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*. (cited on page 10)

Santa Cruz, R.; Fernando, B.; Cherian, A.; and Gould, S., 2018b. Visual permutation learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, (2018). (cited on page 9)

Saxena, A.; Sun, M.; and Ng, A. Y., 2009. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 31, 5 (2009), 824–840. (cited on page 61)

Schroff, F.; Criminisi, A.; and Zisserman, A., 2011. Harvesting image databases from the web. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 33, 4 (2011), 754–766. (cited on page 31)

Sedgewick, R. and Wayne, K., 2011. *Algorithms*. Addison-Wesley Professional. (cited on page 92)

Seo, J.; Han, S.; Lee, S.; and Kim, H., 2015. Computer vision techniques for construction safety and health monitoring. *Advanced Engineering Informatics*, 29, 2 (2015), 239–251. (cited on page 1)

Settles, B., 2010. Active learning literature survey. Technical report, Wisconsin, Madison. (cited on page 35)

Shankar, S.; Garg, V. K.; and Cipolla, R., 2015. Deep-carving: Discovering visual attributes by carving deep neural nets. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 40)

Shao, D.; Xiong, Y.; Zhao, Y.; Huang, Q.; Qiao, Y.; and Lin, D., 2018. Find and focus: Retrieve and localize video events with natural language queries. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 34)

Shashua, A. and Levin, A., 2003. Ranking with large margin principle: Two approaches. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 26)

Shi, M.; Caesar, H.; and Ferrari, V., 2017. Weakly supervised object localization using things and stuff transfer. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages 29 and 30)

Sholomon, D.; David, O.; and Netanyahu, N. S., 2013. A genetic algorithm-based solver for very large jigsaw puzzles. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 39)

Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; and Webb, R., 2017. Learning from simulated and unsupervised images through adversarial training. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 32)

Si, Z. and Zhu, S.-C., 2013. Learning and-or templates for object recognition and detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 35, 9 (2013), 2189–2205. (cited on page 73)

Sigurdsson, G. A.; Varol, G.; Wang, X.; Farhadi, A.; Laptev, I.; and Gupta, A., 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on pages 89, 99, 104, 107, and 108)

Simonyan, K.; Vedaldi, A.; and Zisserman, A., 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*. (cited on page 56)

Simonyan, K. and Zisserman, A., 2014a. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on pages 24 and 25)

Simonyan, K. and Zisserman, A., 2014b. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, abs/1409.1556 (2014). (cited on pages 18, 19, 73, and 77)

Singh, K. K. and Lee, Y. J., 2016. End-to-end localization and ranking for relative attributes. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on pages 27, 41, 55, and 56)

Sinkhorn, R. and Knopp, P., 1967. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21, 2 (1967). (cited on page 42)

Socher, R.; Ganjoo, M.; Manning, C. D.; and Ng, A., 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 34)

Socher, R.; Lin, C. C.; Manning, C.; and Ng, A. Y., 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proc. of the International Conference on Machine Learning (ICML)*. (cited on page 73)

Sorokin, A. and Forsyth, D. A., 2008. Utility data annotation with amazon mechanical turk. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, (2008), 1–8. (cited on page 3)

Souri, Y.; Noury, E.; and Adeli-Mosabbeb, E., 2016. Deep relative attributes. In *Proc. of the Asian Conference on Computer Vision (ACCV)*. (cited on pages 27, 40, 55, and 56)

Srivastava, N.; Mansimov, E.; and Salakhudinov, R., 2015a. Unsupervised learning of video representations using lstms. In *Proc. of the International Conference on Machine Learning (ICML)*. (cited on pages xiv, 95, and 98)

Srivastava, R. K.; Greff, K.; and Schmidhuber, J., 2015b. Highway networks. *arXiv preprint arXiv:1505.00387*, (2015). (cited on page 18)

Su, H.; Qi, C. R.; Li, Y.; and Guibas, L. J., 2015. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 33)

Sun, C.; Shrivastava, A.; Singh, S.; and Gupta, A., 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 3)

Sundermeyer, M.; Marton, Z.-C.; Durner, M.; Brucker, M.; and Triebel, R., 2018. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 28)

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A., 2015. Going deeper with convolutions. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 18 and 19)

Szeliski, R., 2010. *Computer vision: algorithms and applications*. Springer Science & Business Media. (cited on pages 1 and 13)

Tang, W.; Yu, P.; Zhou, J.; and Wu, Y., 2017. Towards a unified compositional model for visual pattern modeling. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 73)

Tarvainen, A. and Valpola, H., 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 29)

Taylor, M.; Guiver, J.; Robertson, S.; and Minka, T., 2008. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. (cited on page 27)

Tesla Motors, Reuters, 2016. Tesla car on autopilot crashes, killing driver. *The Straits Times*, (2016). https://www.straitstimes.com/world/united-states/tesla-car-on-autopilot-crashes-killing-driver. Access date: 11-12-2018. (cited on page 4)

Tian, Y.; Pei, K.; Jana, S.; and Ray, B., 2018. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering*. (cited on page 4)

Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; and Abbeel, P., 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *Proc. of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*. (cited on page 32)

Tokmakov, P.; Schmid, C.; and Alahari, K., 2019. Learning to segment moving objects. *International Journal of Computer Vision (IJCV)*, 127, 3 (2019), 282–301. (cited on page 32)

Torralba, A. and Efros, A. A., 2011. Unbiased look at dataset bias. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 4 and 62)

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M., 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages 24, 25, and 115)

Tsochantaridis, I.; Hofmann, T.; Joachims, T.; and Altun, Y., 2004. Support vector machine learning for interdependent and structured output spaces. In *Proc. of the International Conference on Machine Learning (ICML)*. (cited on page 37)

Tu, Z.; Chen, X.; Yuille, A. L.; and Zhu, S.-C., 2005. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision (IJCV)*, 63, 2 (2005), 113–140. (cited on page 73)

Uijlings, J. R.; Van De Sande, K. E.; Gevers, T.; and Smeulders, A. W., 2013. Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, 104, 2 (2013), 154–171. (cited on page 20)

Varol, G.; Laptev, I.; and Schmid, C., 2018. Long-term temporal convolutions for action recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 40, 6 (2018), 1510–1517. (cited on page 24)

Varol, G.; Romero, J.; Martin, X.; Mahmood, N.; Black, M. J.; Laptev, I.; and Schmid, C., 2017. Learning from synthetic humans. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 33)

Vijayanarasimhan, S. and Grauman, K., 2011. Large-scale live active learning: Training object detectors with crawled data and crowds. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 35)

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A., 2008. Extracting and composing robust features with denoising autoencoders. In *Proc. of the International Conference on Machine Learning (ICML)*. (cited on page 28)

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P.-A., 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research (JMLR)*, 11, Dec (2010), 3371–3408. (cited on page 28)

Viola, P. and Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, I–I. IEEE. (cited on pages 1 and 16)

Vo, N. N. and Bobick, A. F., 2014. From stochastic grammar to bayes network: Probabilistic parsing of complex activity. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2641–2648. (cited on page 90)

Von Neumann, J., 1953. A certain zero-sum two-person game equivalent to the optimal assignment problem. *Contributions to the Theory of Games*, 2 (1953), 5–12. (cited on page 42)

Vondrick, C., 2017. *Predictive Vision*. Ph.D. thesis, Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology. (cited on page 4)

Vondrick, C.; Shrivastava, A.; Fathi, A.; Guadarrama, S.; and Murphy, K., 2018. Tracking emerges by colorizing videos. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on pages 31 and 64)

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S., 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology. (cited on page 80)

Wang, H. and Schmid, C., 2013. Action recognition with improved trajectories. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 25)

Wang, J. and Cherian, A., 2018. Learning discriminative video representations using adversarial perturbations. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on pages 25 and 89)

Wang, J.; Cherian, A.; Porikli, F.; and Gould, S., 2018. Video representation learning using discriminative pooling. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 25)

Wang, L.; Qiao, Y.; and Tang, X., 2015. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 25)

Wang, T.; Gong, S.; Zhu, X.; and Wang, S., 2016. Person re-identification by discriminative selection in video ranking. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 38, 12 (2016), 2501–2514. (cited on page 26)

Wang, X. and Gupta, A., 2015. Unsupervised learning of visual representations using videos. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on pages 31, 63, and 68)

Wang, X.; Wu, J.; Zhang, D.; Su, Y.; and Wang, W. Y., 2019. Learning to compose topic-aware mixture of experts for zero-shot video captioning. In *Proc. of the AAAI Conference on Artificial Intelligence (AAAI)*. (cited on page 34)

Wohlhart, P.; Lepetit, V.; Klatzer, T.; and Pock, T., 2015. Continuous hyper-parameter learning for support vector machines. In *Computer Vision Winter Workshop*. (cited on page 44)

Wu, Q.; Burges, C. J.; Svore, K. M.; and Gao, J., 2010. Adapting boosting for information retrieval measures. *Information Retrieval*, 13, 3 (2010), 254–270. (cited on page 58)

Wu, T. and Zhu, S.-C., 2011. A numerical study of the bottom-up and top-down inference processes in and-or graphs. *International Journal of Computer Vision (IJCV)*, 93, 2 (2011), 226–252. (cited on page 73)

Wu, Z.; Jiang, Y.-G.; Wang, X.; Ye, H.; Xue, X.; and Wang, J., 2015a. Fusing multi-stream deep networks for video classification. *arXiv preprint arXiv:1509.06086*, (2015). (cited on page 25)

Wu, Z.; Wang, X.; Jiang, Y.-G.; Ye, H.; and Xue, X., 2015b. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *In the Proceedings of the 23rd ACM international conference on Multimedia*. (cited on page 25)

Xia, F.; Liu, T.-Y.; Wang, J.; Zhang, W.; and Li, H., 2008. Listwise approach to learning to rank: theory and algorithm. In *Proc. of the International Conference on Machine Learning (ICML)*. (cited on pages 38 and 58)

Xian, Y.; Lampert, C. H.; Schiele, B.; and Akata, Z., 2017. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 80)

Xiong, B.; Jain, S. D.; and Grauman, K., 2019. Pixel objectness: Learning to segment generic objects automatically in images and videos. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 41, 11 (Nov 2019), 2677–2692. (cited on page 33)

Xu, D.; Ouyang, W.; Ricci, E.; Wang, X.; and Sebe, N., 2017a. Learning cross-modal deep representations for robust pedestrian detection. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 34)

Xu, H.; Das, A.; and Saenko, K., 2017b. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 26)

Xu, J. and Li, H., 2007. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. (cited on pages 27 and 58)

Xu, X.; Hospedales, T.; and Gong, S., 2017c. Transductive zero-shot action recognition by word-vector embedding. *International Journal of Computer Vision (IJCV)*, 123, 3 (2017), 309–333. (cited on pages 34 and 90)

Yang, L. and Hanjalic, A., 2010. Supervised reranking for web image search. In *Proceedings of the 18th ACM international conference on Multimedia*. (cited on page 26)

Yeung, S.; Russakovsky, O.; Jin, N.; Andriluka, M.; Mori, G.; and Fei-Fei, L., 2017. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision (IJCV)*, (2017). (cited on pages 89, 99, 104, 105, and 106)

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H., 2014. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NIPS)*. (cited on pages 35 and 65)

Yu, A. and Grauman, K., 2014. Fine-grained visual comparisons with local learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 40 and 56)

Yu, F. and Koltun, V., 2015. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, (2015). (cited on page 23)

Yu, G. and Yuan, J., 2015. Fast action proposals for human action detection and search. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 115)

Yu, X. and Porikli, F., 2017. Hallucinating very low-resolution unaligned and noisy face images by transformative discriminative autoencoders. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 28)

Yuan, J.; Ni, B.; Yang, X.; and Kassim, A. A., 2016. Temporal action localization with pyramid of score distribution features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 26)

Yue-Hei Ng, J.; Hausknecht, M.; Vijayanarasimhan, S.; Vinyals, O.; Monga, R.; and Toderici, G., 2015. Beyond short snippets: Deep networks for video classification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 25)

Zeiler, M. D. and Fergus, R., 2014. Visualizing and understanding convolutional networks. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 73)

Zellers, R. and Choi, Y., 2017. Zero-shot activity recognition with verb attribute induction. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*. (cited on page 34)

Zhang, L.; Xiang, T.; and Gong, S., 2017a. Learning a deep embedding model for zero-shot learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 34)

Zhang, R.; Isola, P.; and Efros, A. A., 2016. Colorful image colorization. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on pages 5, 31, 63, and 68)

Zhang, R.; Isola, P.; and Efros, A. A., 2017b. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 68)

Zhao, H.; Qi, X.; Shen, X.; Shi, J.; and Jia, J., 2018. Icnet for real-time semantic segmentation on high-resolution images. In *Proc. of the European Conference on Computer Vision (ECCV)*. (cited on page 35)

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J., 2017. Pyramid scene parsing network. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 23)

Zhou, Z.-H., 2017. A brief introduction to weakly supervised learning. *National Science Review*, 5, 1 (2017), 44–53. (cited on page 30)

Zhu, L.; Chen, Y.; Lu, Y.; Lin, C.; and Yuille, A., 2008. Max margin and/or graph learning for parsing the human body. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 73)

Zhu, L.; Chen, Y.; Yuille, A.; and Freeman, W., 2010. Latent hierarchical structural learning for object detection. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 73)

Zhu, S.-C.; Mumford, D.; et al., 2007. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2, 4 (2007), 259–362. (cited on page 73)

Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; and Hu, S., 2016. Traffic-sign detection and classification in the wild. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2110–2118. (cited on page 1)

Zhuang, B.; Liu, L.; Li, Y.; Shen, C.; and Reid, I., 2017. Attend in groups: a weakly-supervised deep learning framework for learning from web data. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 31)

Zolfaghari, M.; Oliveira, G. L.; Sedaghat, N.; and Brox, T., 2017. Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In *Proc. of the International Conference on Computer Vision (ICCV)*. (cited on page 25)