UNIVERSITY OF CALIFORNIA

Los Angeles

Sampling Algorithms to Handle Nuisances in Large-Scale Recognition

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

Nikolaos Karianakis

2017

ABSTRACT OF THE DISSERTATION

Sampling Algorithms to Handle Nuisances in Large-Scale Recognition

by

Nikolaos Karianakis
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2017
Professor Stefano Soatto, Chair

Convolutional neural networks (CNNs) have risen to be the *de-facto* paragon for detecting the presence of objects in a scene, as portrayed by an image. CNNs are described as being "approximately invariant" to nuisance transformations such as planar translation, both by virtue of their convolutional architecture and by virtue of their approximation properties that, given sufficient parameters and training data, could in principle yield discriminants that are insensitive to nuisance transformations of the data. The fact that contemporary deep convolutional architectures appear very effective in classifying images as containing a given object regardless of its position, scale, and aspect ratio in large-scale benchmarks suggests that the network can effectively manage such nuisance variability. We conduct an empirical study and show that, contrary to popular belief, at the current level of complexity of convolutional architectures and scale of the data sets used to train them, CNNs are not very effective at marginalizing nuisance variability.

This discovery leaves researchers the choice of investing more effort in the design of models that are less sensitive to nuisances or designing better region proposal algorithms in an effort to predict where the objects of interest lie and center the model around these regions. In this thesis steps towards both directions are made. First, we introduce DSP-CNN, which deploys domain-size pooling in order to transform the neural networks to be scale invariant in the convolutional operator level. Second, motivated by our empirical analysis, we propose novel sampling and pruning techniques for region proposal schemes that improve the end-to-end performance in large-scale classification, detection and wide-baseline correspondence to state-of-the-art levels. Additionally,

since a proposal algorithm involves the design of a classifier, whose results are to be fed to another classifier (a Category CNN), it seems natural to leverage on the latter to design the former. Thus, we introduce a method that leverages on filters learned in the lower layers of CNNs to design a binary boosting classifier for generating class-agnostic proposals. Finally, we extend sampling over time by designing a temporal, hard-attention layer which is trained with reinforcement learning, with application in video sequences for person re-identification.

The dissertation of Nikolaos Karianakis is approved.

Ameet Talwalkar

Ying Nian Wu

Alan Yuille

Stefano Soatto, Committee Chair

University of California, Los Angeles

2017

To Joanna and Elias, my parents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am grateful to my Ph.D. advisor Prof. Stefano Soatto for his brilliant guidance all these years. His sharp feedback was critical for me to accelerate my progress toward our common goal. His deep knowledge of science and his rich life experience had a strong influence on me. He believed in my efforts and provided generous resources to make possible what seemed impossible in the beginning. His open mind allowed me to be creative and shape my own research agenda.

Many thanks go to the members of my committee, Profs. Alan Yuille, Ying Nian Wu and Ameet Talwalkar for their support and suggestions.

I am very fortunate to have worked alongside many exceptional individuals all these years both in UCLA Vision Lab and the places where I interned. I would like to especially stress my collaboration with Jingming Dong. His help and support during all Ph.D. years was invaluable and critical toward the final success. Luckily enough, our forthcoming jobs are in the same city, which will allow us to continue our close friendship. I would like to have a mention to Jason Yosinski, my office mate during our summer internship in JPL. My interaction with him was very influential on my research route. His passion for science and technology is contagious. I would also like to wholeheartedly thank my mentors in my internships for their excellent guidance: Zicheng Liu (Microsoft Research), Yusuke Watanabe, Akira Nakamura and Kenta Kawamoto (Sony Corporation), Thomas Fuchs (JPL) and Yizhou Wang (Peking University).

I met many talented people in UCLA, who made my journey special: Among others, former members Jonathan Balzer, Alper Ayvaci, Michalis Raptis, Josh Hernandez, Konstantine Tsotsos, George Georgiadis, Ioannis Pefkianakis, Masaki Nakada, Amogh Param, Kostas Sideris, Brian Taylor, Vasiliy Karasev, Joachim Valente, Virginia Estellers, Simon Korman, Damek Davis, Siyang Tang, Luca Valente, Byung-Woo Hong, Ren Zhou and Chaohui Wang. I would like to also thank all the current members: Pratik Chaudhari, Xiaohan Fei, Yanchao Yang, Xinzhu Bei, Alessandro Achille, Tong He, Kareem Ahmed, Alhussein Fawzi, Shay Deutsch, Safa Cicek, Alexandre Tiard, Isaac Deutsch, Weize Liu and Peng Zhao. It was a pleasure to work with you.

Finally, I would like to acknowledge the unconditional support of my parents, Joanna and Elias, my sister Erica and my wife Aleka, to whom the thesis is dedicated.

VITA

2011	Diploma in Electrical and Computer Engineering, National Technical University of Athens, Greece.
2011-2014	M.S. in Computer Science, UCLA.
2013	Research Intern, Peking University, Beijing.
2014	Research Intern, NASA's Jet Propulsion Laboratory, Pasadena.
2015	R & D Engineering Intern, Sony, Tokyo.
2012–2015	Teaching Assistant/Associate, Computer Science, UCLA.
2016	Research intern, Microsoft Research, Redmond.
2011–2017	Research Assistant, Computer Science, UCLA Vision Lab.

PUBLICATIONS

N. Karianakis, J. Dong, and S. Soatto. "An Empirical Evaluation of Current Convolutional Architectures' Ability to Manage Nuisance Location and Scale Variability". In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

J. Dong, N. Karianakis, D. Davis, J. Hernandez, J. Balzer, and S. Soatto. "Multiview Feature Engineering and Learning". In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

S. Soatto, J. Dong, and N. Karianakis. “Visual scene representations: Contrast, scaling and occlusion”. In *International Conference on Learning Representations - Workshop Session*, 2015.

S. Soatto, J. Dong, and N. Karianakis. “Visual scene representations: Contrast, scaling and occlusion”. *Technical report, UCLA CSD:140024 - extended version*, 2014.

N. Karianakis and P. Maragos. “An integrated System for Digital Restoration of Prehistoric Theran Wall Paintings”. In *IEEE International Conference on Digital Signal Processing*, 2013.

N. Karianakis, Z. Liu, Y. Chen and S. Soatto. “Person Depth ReID: Robust Person Re-identification with Commodity Depth Sensors”. In *arXiv:1705.09882*, 2017.

N. Karianakis, T. Fuchs, and S. Soatto. “Boosting Convolutional Features for Robust Object Proposals”. In *arXiv:1503.06350*, 2015.

N. Karianakis, Y. Wang and S. Soatto. “Learning to discriminate in the wild: Representation-learning network for nuisance-invariant image comparison”. *Technical Report, UCLA Computer Science Department*, 2013.

CHAPTER 1

Introduction

Over the last decade, Convolutional neural networks (CNNs) [97] have risen to be the *de-facto* paragon for detecting the presence of objects in a scene, as portrayed by an image. CNNs are described as being "approximately invariant" to nuisance transformations such as planar translation, both by virtue of their architecture (the same operation is repeated at every image location akin to a "sliding window") and by virtue of their approximation properties that, given sufficient parameters and training data, could in principle yield discriminants that are insensitive to nuisance transformations of the data. In addition to planar translation, an object detector must manage variability due to scaling (possibly anisotropic along the coordinate axes, yielding different aspect ratios) and partial occlusion. Some nuisances are elements of a transformation group, *e.g.,* the (anisotropic) location-scale group for the case of position, scale and aspect ratio of the object's support (*i.e.,* the region of the image the objects projects onto, often approximated by a bounding box). The fact that contemporary convolutional architectures [91, 146, 158, 68] appear very effective in classifying images as containing a given object regardless of its position, scale, and aspect ratio in large-scale benchmarks (*e.g.,* Imagenet [136]) suggests that the network can effectively manage such nuisance variability. We conduct an empirical study and show in Chapter 2 that, *contrary to popular belief, at the current level of complexity of convolutional architectures and scale of the data sets used to train them, CNNs are not very effective at marginalizing nuisance variability*.

Our empirical analysis reveals a *tradeoff* between *conditioning* the CNN on the "true" location, scale and aspect ratio of the object of interest, which should *improve* performance by suppressing nuisance variability, and *context subtraction* which should *reduce* performance. The fact that padding the ground-truth object bounding box in Imagenet Classification with a 25% rim *improves* performance by a wide margin (*e.g.,* 24% for AlexNet [91] and 37% for VGG-16 [146]), compared

to using the whole image, is non-obvious and indicative of the inability of the CNN to successfully capture context beyond a few pixels. Someone may expect that a CNN, in principle, has the ability to capture co-occurrence statistics on the entire image domain, since the "receptive field" (regions of the image plane) subtended by filters at higher layers encompass a large area of the image. However, the experiments conducted indicate that the CNN is not effectively leveraging such context. This is shown in three steps: First, the baseline performance is comparable to restricting the image to a bounding box containing the object of interest. Second, the baseline performance increases if the image is restricted to the bounding box plus a small rim around it, suggesting that the network indeed can leverage some context. Third, continuing to increase the rim size only hurts the classification accuracy (see Fig. 2.1).

This discovery leaves researchers the choice of investing more effort in the design of models and learning algorithms that are more robust with the nuisances or invent region proposal algorithms with higher recall and accuracy [73] in an effort to predict where the objects of interest lie and center the model around these regions. The latter approach has emerged in split pipelines [60, 159] whereby the image is first pre-processed to yield *proposals*, which are subsets of the image domain (bounding boxes) to be tested for the presence of an object class by a "Category CNN". This seems to be counterproductive, as that way we discard the image outside the proposal window, thus possibly forgoing side information or "context". However, the recent success in [136] has led many researchers away from letting the CNN manage all nuisance variability and instead towards creating better object proposal schemes and subtracting duties from the Category CNN downstream. In this thesis steps toward both directions are made.

A brief justification of the concept of region proposals follows and we state where different chapters of this manuscript fit in that picture. Next, a technique that targets at managing nuisances in the model design is introduced. Afterwards, we present a model that learns away nuisance variability and is applied on one task with no intrinsic variability (occlusion detection) and one task with intra-class variability (segmentation). Finally, a theses outline is provided.

1.1 Conditioning on estimated nuisance transformations

One can think of the conditional distribution of a class c given an image x, $p(c|x)$, as defined by a CNN, as the class posterior $\int_G p(c|x,g)dP(g|x)$ marginalized with respect to the nuisance group G. If the nuisances are known, one can use the class-conditionals $p(c|x,g_r)$ at each nuisance $g_r \in G$ in order to approximate $p(c|x)$ with a weighted average of conditionals, *i.e.,* $p(c|x) \simeq \sum_r p(c|x,g_r)p(g_r|x)$.

When a CNN is tested on a proposal $r \subseteq x$ determined by a reference frame x_r, it computes $p(c|x_{|_r})$ (x restricted to r), which is an approximation of $p(c|x,g_r)$. Then, explicit marginalization (assuming uniform weights) computes $\frac{1}{|r|}\sum_r p(c|x_{|_r})$ which is different from $\frac{1}{|r|}\sum_r p(c|x,g_r)$ which in turn is different from $\sum_r p(c|x,g_r)p(g_r|x)$. This approach is therefore, on average, a lower bound on proper marginalization, and the fact that it would outperform the direct computation of $p(c|x)$ is worth investigating empirically and our primary motivation for our work in Section 2.

Put differently, rather than computing the *posterior* distribution with nuisance transformations automatically marginalized, the CNN is used to compute the *conditional* distribution of classes given the data *and* a sample element that approximates the nuisance transformation, represented by a bounding box. If the goal is the nuisance itself (object support, as in *detection* [136]) it can be found via maximum-likelihood (*max-out*) by selecting the bounding box that yields the highest probability of any class [60]. If the goal is the class regardless of the transformation (as in *categorization* [136]), the nuisance can be approximately *marginalized out* by averaging the conditional distributions with respect to an estimation of the nuisance transformations.

Our empirical analysis in Chapter 2 motivates the widespread use of proposals and extends them in an adaptive fashion. Improved sampling and pruning techniques for heuristic proposal schemes are proposed that improve the end-to-end performance in large-scale classification, detection and wide-baseline correspondence to state-of-the-art levels.

In principle, the best proposal algorithm is the one that densely samples the group. However, even for small-dimensional transformations such as the anisotropic translation-scale group, a single image could yield billions of proposals. As in any sampling procedure, the goal is to trade off

performance with complexity. To this end, *adaptive sampling* schemes can be employed to select proposals based on the data. Since a proposal algorithm involves the design of a classifier, whose results are to be fed to another classifier (a Category CNN), it seems natural to want to leverage on the latter to design the former. For instance, if the Category CNN computes a multi-class posterior with pose, scale and aspect-ratio "marginalized", we could recycle its powerful components to produce a binary distribution (object vs. not) for a *given* pose, scale and aspect ratio, by marginalizing the classes. Therefore, in Chapter 3 we introduce an object proposal method that leverages on filters learned in the lower layers of CNNs to design a binary boosting classifier and a linear regressor to discard as many windows as possible that are unlikely to contain objects of interest.

Sampling proposals can be perceived as an *attention mechanism* as well. Attention models are prevalent in both the computer vision and natural language processing communities and serve as a powerful way to shrink the space of nuisance transformations in tractable levels. Attention is important in both spatial and temporal domain. In Chapter 4 we introduce a novel hard-attention mechanism in video sequences for person re-identification. Our approach is based on the REINFORCE rule and an agent in the form of a recurrent deep neural network. Unlike Chapter 2 where Rényi entropy and max-out are deployed for pruning samples, here a dedicated layer is designed to evaluate the importance of current sample (frame) based on the generated reward, as the latter one is quantified by the expected classification accuracy.

1.2 Handling nuisances in the model architecture

Our former work is based on the premise that a CNN is not as effective in dealing with simple group transformations, which is derived by our analysis in Chapter 2 and the empirical success of Regions-with-CNN approaches [60, 135] in the current benchmarks in use in the community. Of course, empirical tests involve a large number of parameters and design choices that confound the comparison, so it is possible that improvements in the design of CNNs, for instance by allowing them to manage convolutions with respect to larger groups of transformations [59, 32], would render the use of proposals moot. On the other hand, it is possible that the training cost of marginalizing known classes of transformations such as location, scale, aspect ratio, in terms of

size of the data set, may be too high for current architectures, even for convolutional networks that are carefully designed to manage such variability.

A more desirable course of academic action than empirical evaluation, with the ensuing escalating size of the datasets and number of parameters, would be to analyze the representational properties of convolutional architectures to determine the extent in which they can effectively marginalize nuisance variability *by design*, without the need to learn away nuisance variability that is known to exist and well understood. In Appendix A.3 we study the structure of representations, defined as approximations of minimal sufficient statistics that are maximal invariants to nuisance factors, for visual data subject to scaling and occlusion of line-of-sight. We derive analytical expressions for such representations and show that, under certain restrictive assumptions, they are related to features commonly in use in the computer vision community. This link highlights the conditions tacitly assumed by these descriptors, and also suggests ways to improve and generalize them. One proposed technique is domain-size pooling which transforms the convolutional neural networks to be scale invariant in the convolutional operator level. Our model is termed DSP-CNN and we present implementation and experimental results in Section A.3.4. We have also extended DSP-CNN to be deployable for a correspondence task, which is presented in Section 2.2.2.

1.3 Learning away nuisance variability

Our investigation of graphical models starts with Gated Restricted Boltzmann machine [157], which is trained to *learn away* the nuisance variability present in in images, owing to noise and changes of viewpoint and illumination. First, we establish a binary classification task with no intrinsic variability, which amounts to the determination of co-visibility from different images of the same underlying scene. Later, we test our hypothesis in Image Segmentation from a single frame, where the intrinsic variability of the scene objects adds up to the challenge.

1.4 Thesis outline

• Chapter 2: We conduct an empirical study to test the ability of convolutional neural networks (CNNs) to reduce the effects of nuisance transformations of the input data, such as location, scale and aspect ratio. We isolate factors by adopting a common convolutional architecture either deployed globally on the image to compute class posterior distributions, or restricted locally to compute class conditional distributions given location, scale and aspect ratios of bounding boxes determined by proposal heuristics. In theory, averaging the latter should yield inferior performance compared to proper marginalization. Yet empirical evidence suggests the converse, leading us to conclude that – at the current level of complexity of convolutional architectures and scale of the data sets used to train them – CNNs are not very effective at marginalizing nuisance variability. We also quantify the effects of context on the overall classification task and its impact on the performance of CNNs, and propose improved sampling techniques for heuristic proposal schemes that improve end-to-end performance to state-of-the-art levels. We test our hypothesis on a classification task using the ImageNet Challenge benchmark [136] and on a wide-baseline matching task using the Oxford [119] and Fischer [54] datasets.

• Chapter 3: We present a method to generate object proposals, in the form of bounding boxes in a test image, to be fed to a classifier such as a convolutional neural network (CNN), in order to reduce test time complexity of object detection and classification. We leverage on filters learned in the lower layers of CNNs to design a binary boosting classifier and a linear regressor to discard as many windows as possible that are unlikely to contain objects of interest. We test our method against competing proposal schemes, and end-to-end on the Imagenet detection challenge. We show state-of-the-art performance when at least 1000 proposals per frame are used, at a manageable computational complexity compared to alternate schemes that make heavier use of low-level image processing.

• Chapter 4: This chapter targets person re-identification (ReID) from depth sensors such as Kinect. Since depth is invariant to illumination and less sensitive than color to day-by-day appearance changes, a natural question is whether depth is an effective modality for Person ReID, especially in scenarios where individuals wear different colored clothes or over a period of several months.

We explore the use of recurrent Deep Neural Networks for learning high-level shape information from low-resolution depth images. In order to tackle the small sample size problem, we introduce regularization and a hard temporal attention unit. The whole model can be trained end to end with a hybrid supervised loss. We carry out a thorough experimental evaluation of the proposed method on three person re-identification datasets, which include side views, views from the top and sequences with varying degree of partial occlusion, pose and viewpoint variations. To that end, we introduce a new dataset with RGB-D and skeleton data. In a scenario where subjects are recorded after three months with new clothes, we demonstrate large performance gains attained using Depth ReID compared to a state-of-the-art Color ReID. Finally, we show further improvements using the temporal attention unit in multi-shot setting.

• Chapter 5: We test the hypothesis that a representation-learning architecture can train away the nuisance variability present in images, owing to noise and changes of viewpoint and illumination. First, we establish the simplest possible classification task, a binary classification with no intrinsic variability, which amounts to the determination of co-visibility from different images of the same underlying scene. This is the Occlusion Detection problem and the data are typically two sequential, but not necessarily consecutive or in order, video frames. Our network, based on a Gated Restricted Boltzmann machine, learns away the nuisance variability appearing on the background scene and the occluder, which are irrelevant with occlusions, and in turn is capable of discriminating between co-visible and occluded areas by thresholding a one-dimensional semi-metric. Our method, combined with Superpixels [122], outperforms algorithms using features specifically engineered for occlusion detection, such as optical flow, appearance, texture and boundaries. We further challenge our framework with another Computer Vision problem, Image Segmentation from a single frame. We cast it as binary classification too, but here we also have to deal with the intrinsic variability of the scene objects. We perform boundary detection according to a similarity map over all patch pairs and provide a semantic segmentation by leveraging Normalized Cuts [143].

CHAPTER 2

An Empirical Evaluation of Current Convolutional Architectures' Ability to Manage Nuisance Location and Scale Variability

2.1 Introduction

Convolutional neural networks (CNNs) are the de-facto paragon for detecting the presence of objects in a scene, as portrayed by an image. CNNs are described as being "approximately invariant" to nuisance transformations such as planar translation, both by virtue of their architecture (the same operation is repeated at every location akin to a "sliding window" and is followed by local pooling) and by virtue of their approximation properties that, given sufficient parameters and transformed training data, could in principle yield discriminants that are insensitive to nuisance transformations of the data represented in the training set. In addition to planar translation, an object detector must manage variability due to scaling (possibly anisotropic along the coordinate axes, yielding different aspect ratios) and (partial) occlusion. Some nuisances are elements of a transformation group, *e.g.,* the (anisotropic) location-scale group for the case of position, scale and aspect ratio of the object's support.[1] The fact that convolutional architectures appear effective in classifying images as containing a given object regardless of its position, scale, and aspect ratio [91, 146] suggests that the network can effectively manage such nuisance variability.

However, the quest for top performance in benchmark datasets has led researchers away from letting the CNN manage all nuisance variability. Instead, the image is first pre-processed to yield *proposals*, which are subsets of the image domain (bounding boxes) to be tested for the presence

[1]The region of the image the objects projects onto, often approximated by a bounding box.

of a given class (Regions-with-CNN [60]). Proposal mechanisms aim to remove nuisance variability due to position, scale and aspect ratio, leaving a "Category CNN" to classify the resulting bounding box as one of a number of classes it is trained with. Put differently, rather than computing the *posterior* distribution[2] with nuisance transformations automatically marginalized, the CNN is used to compute the *conditional* distribution of classes given the data *and* a sample element that approximates the nuisance transformation, represented by a bounding box. If the goal is the nuisance itself (object support, as in *detection* [136]) it can be found via maximum-likelihood (*max-out*) by selecting the bounding box that yields the highest probability of any class [60]. If the goal is the class regardless of the transformation (as in *categorization* [136]), the nuisance can be approximately *marginalized out* by averaging the conditional distributions with respect to an estimation of the nuisance transformations[2].

Now, if a CNN was an effective way of computing the marginals with respect to nuisance variability, there would be no benefit in conditioning and averaging with respect to (inferred) nuisance samples. This is a direct corollary of the Data Processing Inequality (DPI, Theorem 2.8.1 in [34]). Proposals are subsets of the whole image, so in theory less informative even after accounting for resolution/sampling artifacts (Fig. 2.1). *A fortiori*, performance should further decrease if the conditioning mechanism is not very representative of the nuisance distribution, as is the case for most proposal schemes that produce bounding boxes based on adaptively downsampling a coarse discretization of the location-scale group [73]. Class posteriors conditioned on such bounding boxes discard the image outside it, further limiting the ability of the network to leverage on side information, or "context". Should the converse be true, *i.e.,* should averaging conditional distributions restricted to proposal regions outperform a CNN operating on the entire image, that would bring into question the ability of a CNN to marginalize nuisances such as translation and scaling or else

[2]One can think of the conditional distribution of a class c given an image x, $p(c|x)$, as defined by a CNN, as the class posterior $\int_G p(c|x,g)dP(g|x)$ marginalized with respect to the nuisance group G. If the nuisances are known, one can use the class-conditionals $p(c|x,g_r)$ at each nuisance $g_r \in G$ in order to approximate $p(c|x)$ with a weighted average of conditionals, *i.e.,* $p(c|x) \simeq \sum_r p(c|x,g_r)p(g_r|x)$.

When a CNN is tested on a proposal $r \subseteq x$ determined by a reference frame x_r, it computes $p(c|x_{|_r})$ (x restricted to r), which is an approximation of $p(c|x,g_r)$. Then, explicit marginalization (assuming uniform weights) computes $\frac{1}{|r|}\sum_r p(c|x_{|_r})$ which is different from $\frac{1}{|r|}\sum_r p(c|x,g_r)$ which in turn is different from $\sum_r p(c|x,g_r)p(g_r|x)$. This approach is therefore, on average, a lower bound on proper marginalization, and the fact that it would outperform the direct computation of $p(c|x)$ is worth investigating empirically.

go against the DPI. In this paper we test this hypothesis, aiming to answer to the question: *How effective are current CNNs to reduce the effects of nuisance transformations of the input data, such as location and scaling?*

To the best of our knowledge, this has never been done in the literature, despite the keen interest in understanding the properties of CNNs [63, 160, 126, 145, 160, 179, 182] following their empirical success. We are cognizant of the dangers of drawing sure conclusions from empirical evaluations, especially when they involve a myriad of parameters and exploit training sets that can exhibit biases. To this end, in Sect. 2.2 we describe a testing protocol that uses recognized existing modules, and keep all factors constant while testing each hypothesis.

2.1.1 Contributions

We first show that a baseline (AlexNet [91]) with single-model top-5 error of 19.96% on ImageNet 2014 Classification slightly *decreases* in performance (to 20.41%) when constrained to the ground-truth bounding boxes (Table 2.1). This may seem surprising at first, as it would appear to violate Theorem 2.6.5 of [34] (on average, conditioning on the true value of the nuisance transformation must reduce uncertainty in the classifier). However, note that the restriction to bounding boxes does not just condition on the location-scale group, but also on *visibility*, as the image outside the bounding box is ignored. Thus, *the slight decrease in performance measures the loss from discarding context by ignoring the image beyond the bounding box.* When we pad the true bounding boxes with a 10-pixel rim, we show that, conditioned on such "ground-truth-with-context" indeed does decrease the error as expected, to 17.65%. In Fig. 2.1 we show the classification performance as a function of the rim size all the way to the whole image for AlexNet and VGG16 [146]. A 25% rim yields the lowest top-5 errors on the ImageNet validation set for both models. This also indicates that the context effectively leveraged by current CNN architectures is limited to a relatively small neighborhood of the object of interest.

The second contribution concerns the *proper sampling* of the nuisance group. If we interpret the CNN restricted to a bounding box as a function that maps samples of the location-scale group to class-conditional distributions, where the proposal mechanism *down-samples* the group, then

Method	AlexNet		VGG16	
Whole image	19.96		13.24	
Ground-Truth Bounding Box (GT)	20.41		12.44	
	Isotropically	Anisotropically	Isotropically	Anisotropically
GT padded with 10 px	17.66	17.65	10.91	10.30
Ave-GT, 4 domain sizes (padded with [0,30] px)	15.96	16.00	9.65	8.90
Ave-GT, 8 domain sizes (padded with [0,70] px)	14.43	14.22	8.66	7.84

Table 2.1: AlexNet's and VGG16's top-5 error on the ImageNet 2014 classification challenge when the ground-truth localization is provided, compared to applying the model on the entire image. We pad the ground truth with various rim sizes both isotropically and anisotropically. Then we show how averaging the class posteriors performs when applying the network on concentric domain sizes around the ground truth.

classical sampling theory [141] teaches that we should retain *not* the value of the function at the samples, but its *local average*, a process known as *anti-aliasing*. Also in Table 2.1, we show that simple uniform averaging of 4 and 8 samples of the isotropic *scale* group (leaving location and aspect ratio constant) reduces the error to 15.96% and 14.43% respectively. This is again unintuitive, as one expects that averaging conditional densities would produce less discriminative classifiers, but in line with recent developments concerning "domain-size pooling" [43].

To test the effect of such anti-aliasing on a CNN absent the knowledge of ground truth object location, we follow the methodology and evaluation protocol of [54] to develop a domain-size pooled CNN and test it in their benchmark task of wide-baseline correspondence of regions selected by a generic low-level detector (MSER [115]). Our third contribution is to show that this procedure improves the baseline CNN by 5–15% mean AP on standard benchmark datasets (Table 2.3 and Fig. 2.6 in Sect. 2.2.2).

Our fourth contribution goes towards answering the question set forth in the preamble: We consider two popular baselines (AlexNet and VGG16) that perform at the state-of-the-art in the ImageNet Classification challenge and introduce novel sampling and pruning methods, as well as an adaptively weighted marginalization based on the inverse Rényi entropy. Now, if *averaging* the conditional class posteriors obtained with various sampling schemes should improve overall per-

formance, that would imply that the *implicit* "marginalization" performed by the CNN is inferior to that obtained by sampling the group, and averaging the resulting class conditionals.[2] This is indeed our observation, *e.g.,* for VGG16, as we achieve an overall performance of 8.01%, compared to 13.24% when using the whole image (Table 2.2). There are, however, caveats to this answer, which we discuss in Sect. 2.10.

Our fifth contribution is to actually provide a method that performs at the state of the art in the ImageNet Classification challenge when using a single model. In Table 2.2 we provide various results and time complexity. We achieve a top-5 classification error of 15.82% and 8.01% for AlexNet and VGG16, compared to 17.55% and 8.85% error when they are tested with 150 regularly sampled crops [146], which corresponds to 9.9% and 9.4% relative error reduction, respectively. Data augmentation techniques such as scale jittering and an ensemble of several models [68, 146, 158] could be deployed along with our method.

The source code implementing our method and the scripts necessary to reproduce the evaluation are available at `http://vision.ucla.edu/~nick/proj/cnn_nuisances/`.

2.1.2 Related work

The literature on CNNs and their role in Computer Vision is rapidly evolving. Attempts to understand the inner workings of CNNs are being conducted [26, 63, 160, 99, 126, 145, 160, 179, 182], along with theoretical analysis [6, 20, 32, 150] aimed at characterizing their representational properties. Such intense interest was sparked by the surprising performance of CNNs [26, 39, 60, 67, 68, 91, 135, 138, 146, 158] in Computer Vision benchmarks [136, 48], where many couple a proposal scheme [3, 21, 29, 47, 73, 75, 90, 111, 132, 164, 192] with a CNN. As our work relates to a vast body of work, we refer the reader to references in the papers that describe the benchmarks we adopt, namely [26], [91] and [146].

Bilen et. al. [17] explored the idea of introducing proposals in classification. However, their approach leveraged on a significantly larger number of candidates and used sophisticated classifiers and post-normalization of class posteriors. Our method selects a very small subset of the most discriminative candidates among generic object proposals, while building on popular CNN models.

2.2 Experiments

2.2.1 Large-scale Image Classification

What if we trivialize location and scaling? First, we test the hypothesis that eliminating the nuisances of location and scaling by providing a bounding box for the object of interest will improve the classification accuracy. This is not a given, for restricting the network to operate on a bounding box prevents it from leveraging on context outside it. We use the AlexNet and VGG16 pre-trained models, which are provided with the MatConvNet open source library [165], and test their top-1 and top-5 classification errors on the ImageNet 2014 classification challenge [136]. The validation set consists of $50,000$ images, where at each of them one "salient" class is annotated a priori by a human. However, other ImageNet classes appear in many of the images, which can confound any classifier.

We test the classifier in various settings (Table 2.1); first, by feeding the entire image to it and letting the classifier manage the nuisances. Then we test the ground-truth annotated bounding box and concentric regions that include it. We try both isotropic and anisotropic expansion of the ground-truth region. We observe similar behavior, which is also consistent for both models.

Only for AlexNet at Table 2.1 using the object's ground-truth support performs slightly worse than using the whole image. After we pad the object region with a 10-pixel rim, the top-5 classification error decreases fast. However, there is a trade-off between context and clutter. Providing too much context has diminishing returns. In Fig. 2.1 we show how the errors vary as a function of the rim size around the object of interest. Performance starts dropping down when we add more than 25% rim size. This padding gives 15.08% and 8.37% top-5 error for AlexNet and VGG16, as opposed to 19.96% and 13.24% respectively, when classifying the whole image.

To ensure that this improvement is not due to downsampling, we repeat the experiment with fixed resolution for the whole image and every subregion. We achieve this by shrinking each region with the same downsampling factor that we apply to the whole image to pass to the CNN. Finally we rescale the downsampled region to the CNN input. These results appear with the label "same resolution" in Fig. 2.1.

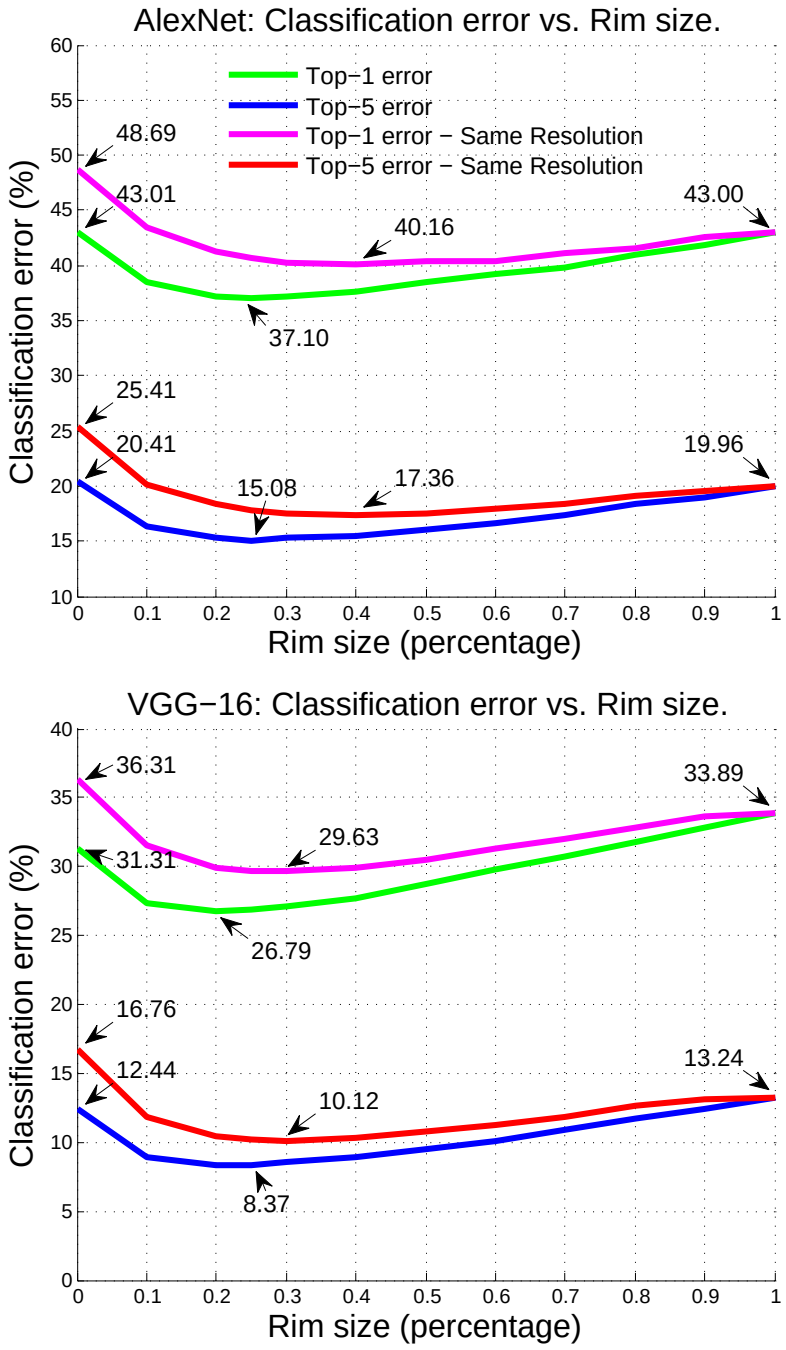


Figure 2.1: The top-1 and top-5 classification errors in ImageNet 2014 as a function of the rim size for AlexNet (above) and VGG16 (below) architecture. A 0 rim size corresponds to the ground-truth bounding box, while 1 refers to the whole image. A relatively small rim around the ground truth provides the best trade-off between informative context and clutter.

Finally, we apply domain size average pooling on the class posterior (*i.e.,* the network's softmax output layer) with 4 and 8 domain sizes that are concentric with the ground truth. The added rim has the declared size either at both dimensions (for the anisotropic case) or only along the minimum dimension (for the isotropic case), and it is uniformly sampled in the range $[0, 30]$ and $[0, 70]$, respectively. The latter one further reduces the top-5 error to 14.22% for AlexNet, which is lower than any single domain size (cf. Fig. 2.1). This suggests that explicitly marginalizing samples can be beneficial. Next we test whether the improvement stands when using object proposals.

Introducing object proposals. We deploy a proposal algorithm to generate "object" regions within the image. We use Edge Boxes [192], which provide a good trade-off between recall and speed [73].

First, we decide the number of proposals which will provide a satisfactory cover for the majority of objects present in the dataset. In a single image we search for the highest Intersection over Union (IoU) overlap between the ground-truth region and any proposed sample and in turn we evaluate the network's performance on the most overlapping sample. We repeat this process for various number of proposals N in a small subset of validation set and finally choose $N = 80$, which provides a satisfactory trade-off between classification performance and computational cost. The procedure is described in detail in Section 2.4.

Among the extracted proposals, we choose the most informative subset for our task, based on pruning criteria that we introduce below. Next we discuss what other samples we use, which are also drawn in Fig. 2.2.

Domain-size pooling and regular crops. We investigate the influence of domain-size pooling at test time both as stand-alone technique and as additional proposals for the final method which is described in Algorithm 1. We deploy domain-size aggregation of the network's class posterior over D sizes that are uniformly sampled in the range $[r, 1]$, where 1 is the normalized size of the original image. After parameter search, we choose $D = 5$ and $r = 0.6$. We use both the original and the horizontally flipped area, which gives 10 samples in total.

Finally, we use standard data augmentation techniques from the literature. As customary, the image is isotropically rescaled to a predefined size, and then a predetermined selection of crops is

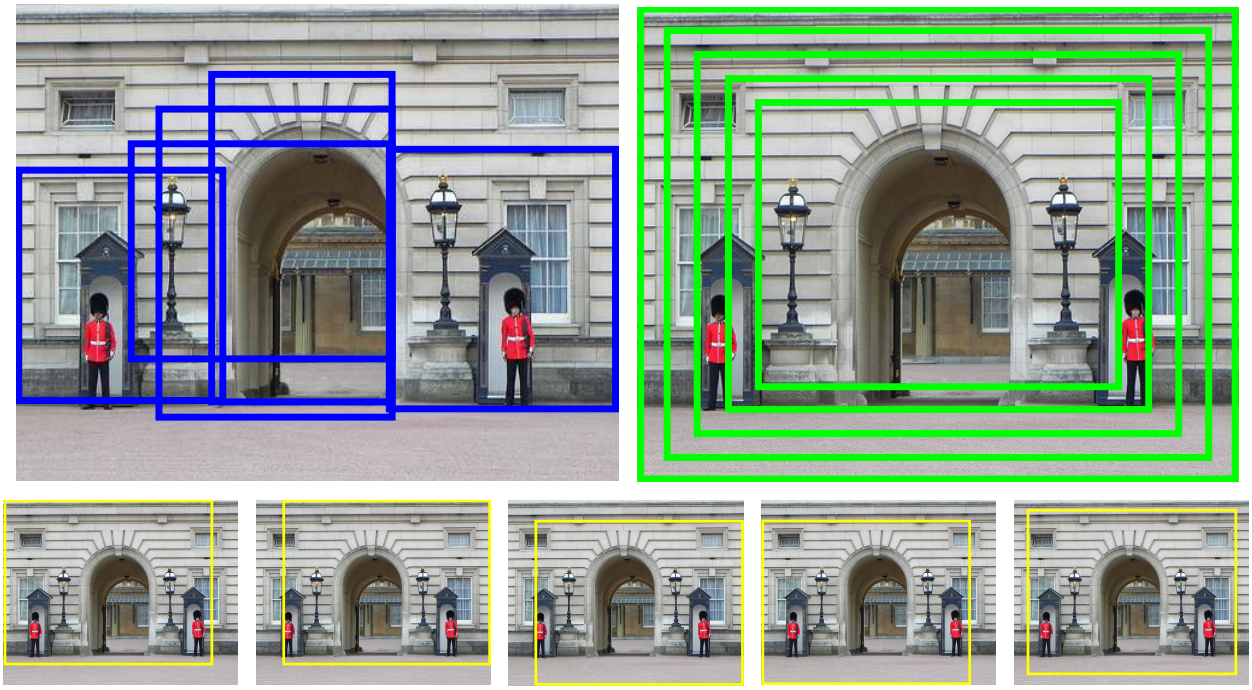


Figure 2.2: Visualizing different sampling strategies. Upper left: Object proposals. Generic proposals using Edge Boxes [192]. Upper right: Concentric domain sizes are centered at the center of the image. Below: Regular crops [91, 146, 158]. This is an ILSVRC example where the object proposals help the classifier to recognize the bearskin cap, as opposed to multi-crop augmentation.

extracted [91, 146, 158] or the network is applied densely and a class score map over the whole image is extracted [138, 146]. We compare our method with the multi-crop strategies which have been shown to perform marginally better compared to dense processing [146].

Pruning samples. Continuing to sample patches within the image has diminishing return in terms of discriminability, while including more background patches with noisy class posterior distribution. We adopt an information-theoretic criterion to filter the samples that we use for the subsequent approximate marginalization.

For each candidate proposal $n \in N$ we evaluate the network and take the normalized softmax output $v^n \in \mathbb{R}^{\mathcal{C}}$, where $v_i^n \in [0,1], i = \{1,\ldots,\mathcal{C}\}$ and $\mathcal{C} = 1,000$ on ILSVRC classification. The output is a set of non-negative numbers which sum up to 1. We can interpret the vector v^n as a probability distribution on the discrete space of classes $\{1,\ldots,\mathcal{C}\}$ and compute the Rényi entropy as $\mathbb{H}_\alpha(v^n) = \frac{1}{1-\alpha} log(\sum_{i=1}^{\mathcal{C}} (v_i^n)^\alpha)$.

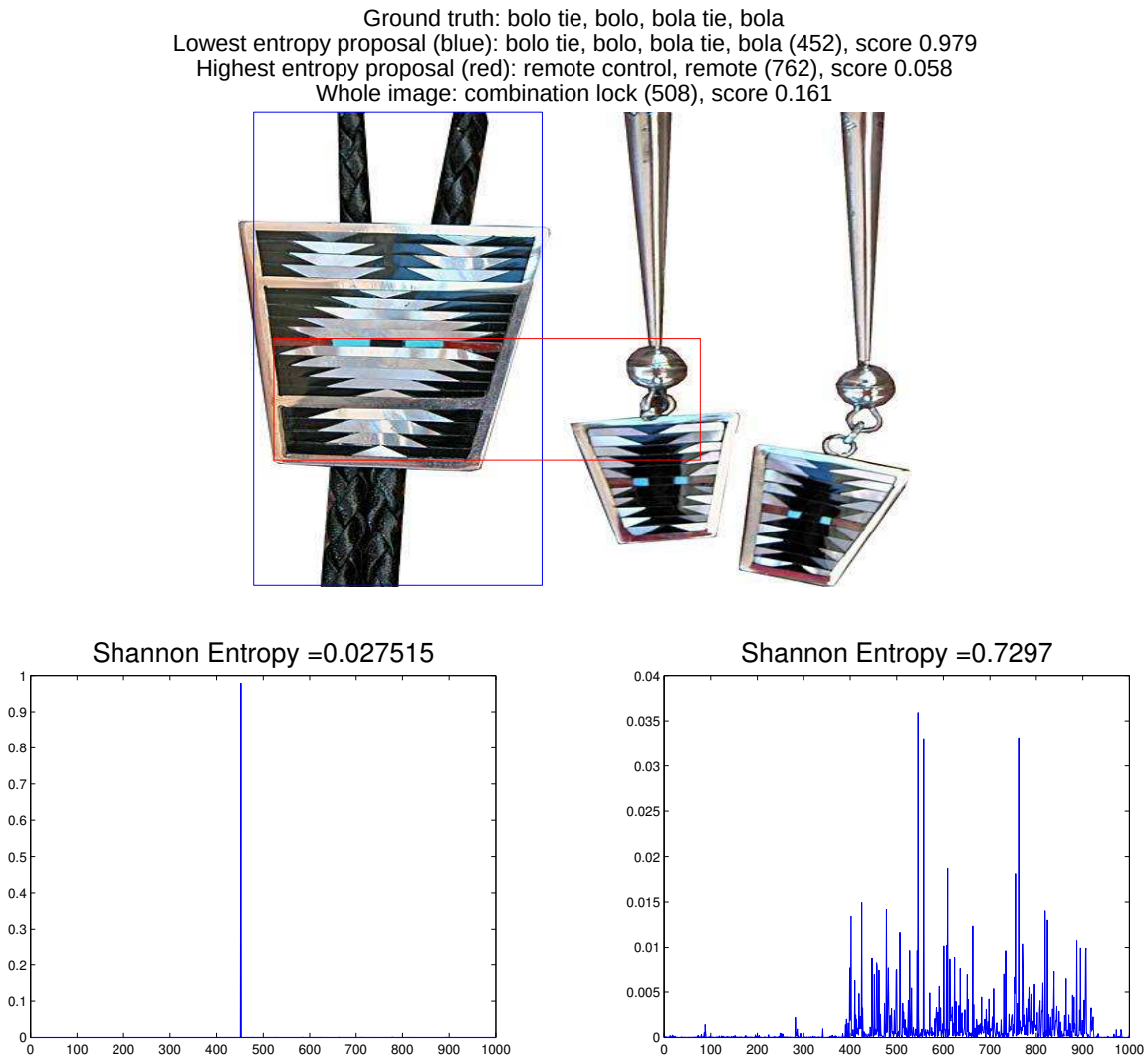


Figure 2.3: An ILSVRC image where the network is not confident and wrong when it is conditioned on the whole image, while the lowest entropy posterior makes the prediction correct with high confidence.

Our conjecture is that more discriminative class distributions tend to be more peaky with less ambiguity among the classes, and therefore lower entropy. In Fig. 2.3 we show an ILSVRC example where the proposal with the lowest-entropy posterior is classified correctly and with high confidence as opposed to conditioning the prediction on the whole image. In Fig. 2.4 we show how selecting a subset of image patches whose class posterior has lower entropy improves classification performance.

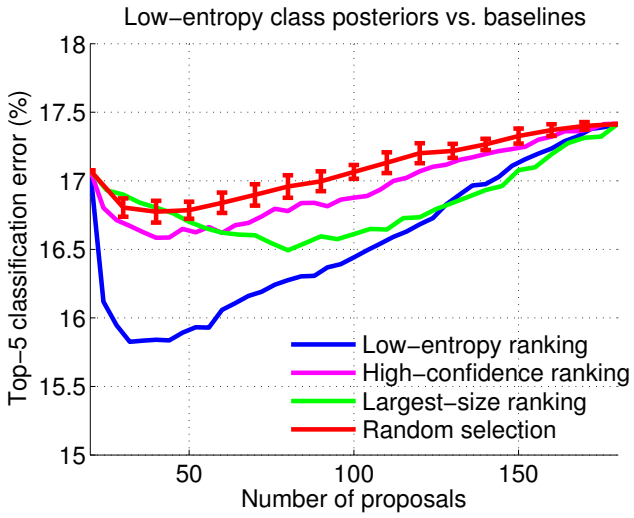


Figure 2.4: We show the top-5 error as a function of the number of proposals we average to produce the final posterior. Samples are generated with Algorithm 1 and classified with AlexNet. The blue curve corresponds to selecting samples with the lowest-entropy posteriors. We compare our method with simple strategies such as random selection, ranking by largest-size or highest confidence of proposals. The random sample selection was run 10 times and we visualize the estimated 99.7% confidence intervals as error-bars. We observe that the discriminative power of the classifier clearly increases when the samples are selected with the least Rényi entropy criterion.

We extract N candidate object proposals[3] [192] and evaluate the network for both the original candidates and their horizontal flips. Then we keep a small subset E, whose posterior distribution has the lowest entropy. We use Rényi entropy with relatively small powers ($\alpha = 0.35$), as we found that it encourages selecting regions with more than one highly-confident candidate object. While the parameter α increases, the entropy is increasingly determined by the events of highest probability. Larger α would be more effective for images with a single object, which is not the case in most images in ILSVRC.

Finally we introduce a weighted average of the selected posteriors as $\sum_r p(c|x_{|_r})p(x_{|_r})$, where $x_{|_r}$ is the support of sample r and $p(x_{|_r})$ is the weight of its posterior[2]. We try both uniform weights and weights proportional to the inverse entropy of the posterior $p(c|x_{|_r})$. The latter is expected to perform better, as it naturally gives higher weight to the most discriminative samples.

[3]We introduce a prior encouraging the largest proposals among the ones that the standard setting in [192] would give. To this end, we generate 200 proposals and keep the $N = 80$ largest ones (see Algorithm 1).

Algorithm 1 Regular & adaptive sampling in classification.

- *Object proposals.* We extract several object proposals from the image x (*e.g.,* 200 *Edge Boxes* [192] and keep the N largest ones). Among them we choose E proposals whose class posterior has the lowest *Rényi entropy* with parameter α. After hyper-parameter search, we choose $N = 80$, $E = 12$ and $\alpha = 0.35$.
- D *concentric* domain sizes around the center of x (including their horizontal flip). We use 5 sizes that are uniformly extracted in the normalized range $[0.6, 1]$, where 1 corresponds to the whole image ($D = 10$).
- C *crops.* Regular crops; *e.g.,* $C = 10$ or $C = 50$ in 1 or 3 scales, as in [91, 146, 158].
- The class conditionals are approximated as $\sum_r p(c|x_{|_r})p(x_{|_r})$, where $p(x_{|_r})$ is either uniform or equals to the inverse entropy of the posterior $p(c|x_{|_r})$.

All regular and adaptive sampling components are summarized in Algorithm 1 and are drawn in Fig. 2.2. These E proposals are classified by a Convolutional Neural Network and the multiple outputs are averaged element-wise in order to extract a single vector (of size $1,000 \times 1$ for Imagenet classification), which is our class posterior for the whole image.

Comparisons. To compare various sampling and inference strategies, we use the AlexNet and VGG16 models. All classification results in Table 2.2 refer to the validation set of the ILSVRC 2014 [136], except for the last row which demonstrates results on the test set. On the rows 2–5 we show the performance of popular multi-crop methods [91, 146, 158]. Then we compare them with strategies that involve concentric domain sizes (rows 6–8) and object proposals (rows 9–16).

Before extracting the crops and in order to preserve the aspect ratio of each single image, we rescale it so that its minimum dimension is 256. The proposals are extracted at the original image resolution and then they are rescaled anisotropically to fit the model's receptive field. Additionally, some multi-crop algorithms resize the image in S different scales and then sample C patches of fixed size 224×224 densely over the image. Szegedy et al. [158] use $S = 4$ scales and $C = 36$ crops per scale, which yields 144 patches in all. Following the methodology from Simonyan et al. [146], it is comparable to deploy $S = 3$ scales and extract $C = 50$ crops per scale (5×5 regular

Method			AlexNet			VGG16			eval-S	ave-S
# crops	# sizes	# proposals	top-1	top-5	t (10^3s)	top-1	top-5	t (10^3s)		
–	$D = 1$	–	43.00	19.96	0.5	33.89	13.24	2.8	1	1
$C = 10$	–	–	41.50	18.69	3.1	27.55	9.29	24	10	10
$C = 50$	–	–	41.01	18.05	33	27.44	9.12	67	50	50
$C = 10 \times 3$	–	–	40.58	17.97	7.9	27.23	8.88	63	30	30
$C = 50 \times 3$	–	–	40.41	17.55	41	27.14	8.85	174	150	150
–	$D = 10$	–	40.00	17.86	3.8	28.16	9.46	30	10	10
$C = 10$	$D = 10$	–	39.38	17.08	11	26.94	8.83	54	20	20
$C = 10 \times 3$	$D = 10$	–	39.36	17.07	23	26.76	8.68	94	40	40
–	–	$E = 40$	40.18	17.53	63	25.60	8.24	151	160	40
$C = 10$	–	$E = 20$	38.91	16.63		25.28	7.91		170	30
–	$D = 10$	$E = 12$	38.05	16.19	67	25.19	8.11	219	170	22
$C = 10$	$D = 10$	$E = 12$	**37.69**	**15.83**		**25.11**	**8.01**		180	32
$C = 10$	$D = 10$	$E = 12$ (fast)	37.71	15.88	47	25.12	8.08	185	180	32
–	$D = 10$	$E = 12$ (W)	37.98	16.12	64	25.23	8.10	190	170	22
$C = 10$	$D = 10$	$E = 12$ (W)	37.57	15.82		25.11	8.02		180	32
$C = 10$	$D = 10$	$E = 12$ (test)	**37.417**	**16.018**	–	**25.117**	**7.909**	–	180	32

Table 2.2: Top-1 and top-5 errors on the ImageNet 2014 classification challenge. The rows 2–5 include customary data augmentation strategies in the literature [91, 146, 158] (*i.e.,* regular sampling). The next three rows use concentric domain sizes that are uniformly sampled in the range $[0.6, 1]$ with 1 being the normalized size of the original image (cf. Fig. 2.2). In the rest of the rows we introduce adaptive sampling, which consists of a data-driven object proposal algorithm [192] and an entropy criterion to select the most discriminative samples on the fly based on the extracted class posterior distribution. 'W' denotes the methods that use weighted marginalization (rows 14 and 15). The last row shows results on the test set. $\#eval$ stands for the number of samples that are evaluated for each method, while $\#ave$ is the number of samples that are eventually element-wise averaged to produce one single vector with class confidences. The previous top-performing techniques with regular sampling and our results are shown in bold. In specific, we emphasize our top-performing method in the validation and its corresponding entry on the test set.

grid with flips), for a total of 150 crops over 3 scales (row 5 in Table 2.2).

The results, presented in Table 2.2, indicate as expected that scale jittering at test time improves the classification performance for both 10-crop and 50-crop strategies. Additionally, the 50-crop strategy is better than the 10-crop strategy for both models. The results on row 5 in bold are the lowest errors that can be achieved with these specific single models[4] using only regular crops.

Then we present our methods and observe that using the AlexNet network with $D = 10$ concentric domain sizes outperforms most multi-crop algorithms even if it only evaluates and averages 10 patches. Furthermore, combining it with 10 common crops achieves the best results for both networks, even without using 3-scale jittering. One interpretation for these improvements is that the concentric samples serve a natural prior for the majority of ILSVRC images, *i.e.,* the object of interest lies most probably at the center than at the image boundaries. This is a common assumption in the literature that also appears in large-scale video segmentation [89].

Following, we introduce the adaptive sampling mechanism with Algorithm 1 and reduce the top-5 error to 15.83% and 8.01% for AlexNet and VGG16 respectively. To set this in perspective, Krizhevsky et al. [91] report 16.4% top-5 error when they combine 5 models. We improve this performance with one single model. The relative improvement for the deployed instances of AlexNet and VGG16, compared to the data-augmentation methods used in [146, 158], is 9.9% and 9.4%, respectively. Rows 14 and 15 show results where the marginalization is weighted based on the entropy (notated as W), in contrast to methods that appear in rows 9–13, which use uniform weights (cf. Algorithm 1). At the last row we show results from the ILSVRC test server for our top-performing method (row 12).

We evaluate our method also with an instance of googLeNet [158]. We deploy the Princeton version of the model which is provided by Matconvnet. This instance does not achieve the top-performing results reported in [158], as a simpler training process is followed. Nevertheless our

[4]Specifically, we use the VGG16 model which is trained without scale jittering at training and appears on the first row of D area in Table 3 in [146]. Pre-trained models for both AlexNet and VGG16 are publicly available with the MatConvNet toolbox [165]. Simonyan et al. in their evaluation with 50 crops and 3 scales report 8.6% top-5 error on ImageNet 2014 validation. In contrast our implementation produces 8.85%, which can be attributed to using a different pre-trained model, as the initial weights are sampled from a zero-mean Gaussian distribution with standard deviation 0.01 and there might also be minor differences in the training process.

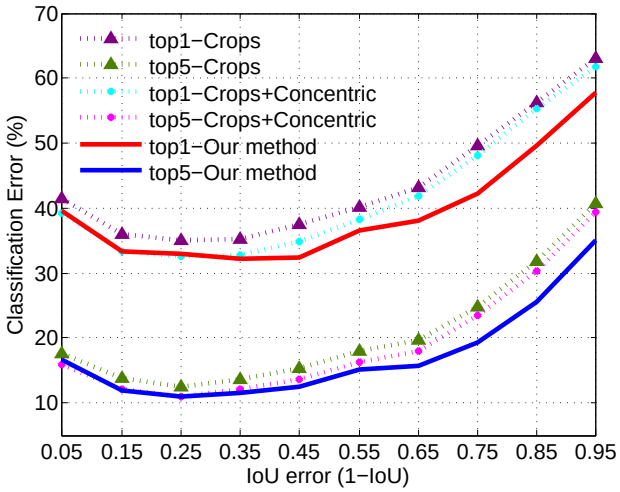


Figure 2.5: Classification error as a function of the IoU error between the objects and the regular and concentric crops.

primary focus is in the relative improvements compared to the baselines. We evaluate it using the whole images, with standard 50-crop augmentation at 3 scales [146] and using our top-performing variant; the top1/top5 error is $33.39/12.36$, $30.86/10.70$ and $29.22/9.67$, respectively. Thus, we have 9.6% relative top5 error reduction compared to using 50×3 crops.

Regular and concentric crops assume that objects occupy most of the image or appear near the center. This is a known bias in the ImageNet dataset. To analyze the effect of adaptive sampling, we calculate the intersection over union error between the objects and the regular and concentric crops, and show in Fig. 2.5 the performance of various methods as a function of the IoU error. The improvement of using adaptive sampling (via proposals) over only regular and concentric crops is increased as IoU error grows, indicating that objects occupy less domain or are far away from the center.

Time complexity. In Table 2.2 we show the number of evaluated samples ($\#eval$) and the subset that is actually averaged ($\#ave$) in order to extract a single class posterior vector. The sequential time needed for each method is linear to the number of evaluated patches $\#eval$. We run the experiments with the MatConvNet library and parallelize the load for VGG16 so that the testing

is done in batches of $B = 20$ patches. We report the time profile[5] for each method in Table 2.2. A few entries cover two boxes, as their methods are evaluated together. Extracting the proposals is not a major bottleneck if using an efficient algorithm [73], such as Edge Boxes [192]. In row 13 we report results of our faster version, where the Edge Boxes do not leverage edge sharpening and use one decision tree. Overall, compared to the 150-crop strategy, the object proposal scheme introduces only marginal computational overhead.

2.2.2 Wide-Baseline Correspondence

We test the effect of domain-size pooling in correspondence tasks with a convolutional architecture, as done by [43] for SIFT [107], using the datasets and protocols of [54]. This is illustrated in Fig. 2.2 (upper right), but here the domain sizes are centered around the detector. We expect that such averaging will increase the discriminability of detected regions and in turn the matching ability, similar to the benefits that we see on the last rows of Table 2.1.

We use maximally-stable extremal regions (MSER) [115] to detect candidate regions, affine-normalize them, align them to the dominant orientation, and re-scale them for head-to-head comparisons. For a detected scale σ at each MSER, the DSP-CNN samples D domain sizes within a neighborhood $[\lambda_1\sigma, \lambda_2\sigma]$ around it, computes the CNN responses on these samples and averages the posteriors. The deployed deep network is the unsupervised convolutional network proposed by [54], which is trained with surrogate labels from an unlabeled dataset (see the methodology in [44]), with the objective of being invariant to several transformations that are commonly observed in images captured from different viewpoints. As opposed to network-classifiers, here the task is correspondence and the network is purely a region descriptor, whose last two layers (3 and 4) are the representations.

In Fig. 2.6 (left) we show the comparison between CNN and DSP-CNN on Oxford dataset [119]. CNN's layer 4 is the representation for each MSER and DSP-CNN simply averages this layer's responses for all D domain sizes. We use $\lambda_1 = 0.7$, $\lambda_2 = 1.5$ and $D = 6$ sizes that are uniformly sampled in this neighborhood. There is a 15.1% improvement based on the matching

[5] We use a machine equipped with a NVIDIA Tesla K80 GPU, 24 Intel Xeon E5 cores and 64G RAM memory.

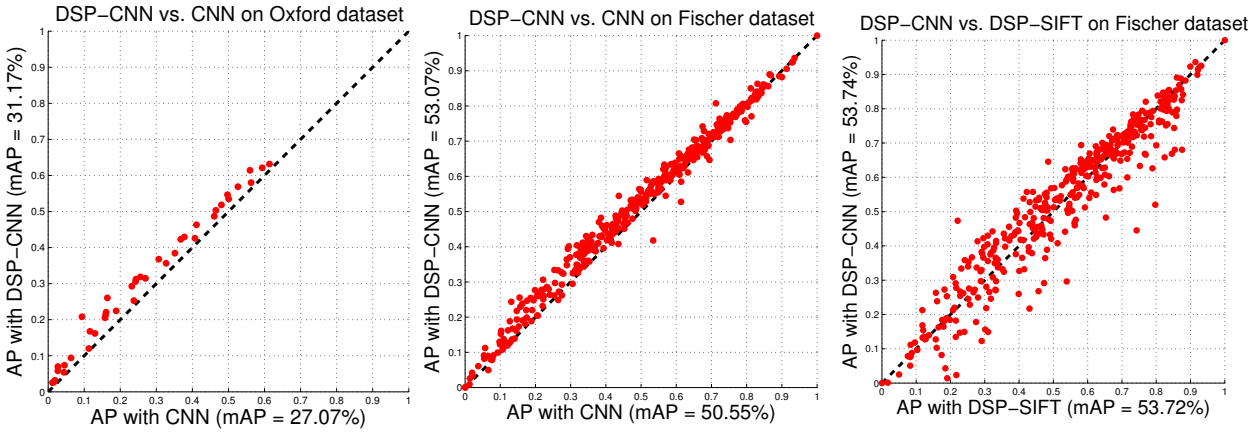


Figure 2.6: Head to head comparison between CNN and DSP-CNN on the Oxford [119] (left) and Fischer's [54] (center) datasets. The layer-4 features of the unsupervised network from [54] are used as descriptors. The DSP-CNN outperforms its CNN counterpart in terms of matching mAP by 15.1% and 5.0%, respectively. Right: DSP-CNN performs comparably to the state-of-the-art DSP-SIFT descriptor [43].

Method	Dim	mAP
Raw patch	4,761	34.79
SIFT [107]	128	45.32
DSP-SIFT [43]	128	**53.72**
CNN-L3 [54]	9,216	48.99
CNN-L4 [54]	8,192	50.55
DSP-CNN-L3	9,216	52.76
DSP-CNN-L4	8,192	53.07
DSP-CNN-L3-L4	17,408	**53.74**
DSP-CNN-L3 (PCA128)	128	51.45
DSP-CNN-L4 (PCA128)	128	52.33
DSP-CNN-L34 (concat. PCA128)	256	**52.69**

Table 2.3: Matching mean average precision for different approaches on Fischer's dataset [54].

mean average precision.

Fischer's dataset [54] includes 400 pairs of images, some of them with more extreme transformations than those in the Oxford dataset. The types of transformations include zooming, blurring, lighting change, rotation, perspective and nonlinear transformations. In Fig. 2.6 (center) and Table 2.3 we show comparisons between CNN and DSP-CNN for layer-3 and layer-4 representations and demonstrate 7.7% and 5.0% relative improvement. We use $\lambda_1 = 0.5$, $\lambda_2 = 1.4$ and $D = 10$ domain sizes. These parameters are selected with cross-validation. In Table 2.3 we show comparisons with baselines, such as using the raw data and DSP-SIFT [43]. After fine parameter search ($\lambda_1 = 0.5$, $\lambda_2 = 1.24$) and concatenating the layers 3 and 4, we achieve state of the art performance as shown in Fig. 2.6 (right), observing though the high dimensionality of this method compared to local descriptors.

Given the inherent high-dimensionality of CNN layers, we perform dimensionality reduction with principal component analysis to investigate how this affects the matching performance. In Table 2.3 we show the performance for compressed layer-3 and layer-4 representations with PCA to 128 dimensions and their concatenation. There is a modest performance loss, yet the compressed features outperform the single-scale features by a large margin.

2.3 Comparison between Marginalization and Max-out

In the task of classification, nuisance variability of factors such as translation, scale and aspect ratio is explicitly handled by the use of crops, concentric domains and proposals. Each of them represents an element g in the nuisance group G. Conditioned on g, a "Category" convolutional neural network returns a conditional posterior probability of the learned classes, $p(c|x, g)$ where x is the test image. To obtain a prediction independent of the nuisance G, one can either *marginalize*

$$p(c|x) = \int p(c|x, g) dP(g), \tag{2.1}$$

and extract the classes c with maximum posterior or *max-out*

$$\hat{c} = \arg\max_{g,c} p(c|x, g). \tag{2.2}$$

over all possible elements g and classes c.

The former has been extensively evaluated in Section 2.2.1. The latter has an additional benefit that allows to identify the nuisance element g which corresponds to the predicted class c via

$$\hat{g} = \arg\max_{g} p(c|x, g). \tag{2.3}$$

This helps to "localize" the object(s) of interest up to translation, scale and aspect ratio changes that are modeled by G. In this section, we evaluate the performance of max-out on the ILSVRC benchmark using again the same networks (AlexNet and VGG) as in Section 2.2.1.

As a comparison, we use the same number of crops, concentric domains and proposals as in row 12 of Table 2.2 ($C = 10, D = 10$ and $E = 12$). Instead of averaging the conditional posteriors, we find the maxima according to Eq. 2.2. Max-out achieves a top-1 error 40.22% and a top-5 error 17.44%. In Fig. 2.7, we show in blue lines the indices of images on which marginalization predicts the class label correctly but max-out does not, and in green lines the indices of images on which max-out wins. After inspecting the images where max-out fails, we observe that some of the failure cases are caused by the fact that ILSVRC only allows *one* class annotation while regions of proposals can contain other objects that are not considered the "ground-truth" class.

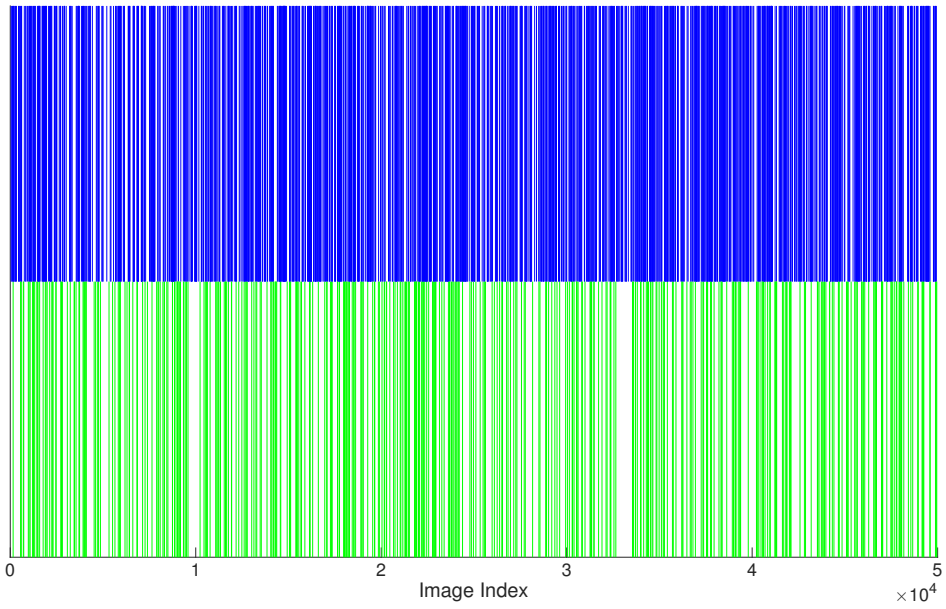


Figure 2.7: Comparison between Marginalization and Max-out. Blue lines show the images on which marginalization predicts the class label correctly but max-out does not. Green lines show the opposite.

2.4 Choosing the number of proposals

In Section 2.2.1 we describe that we use Edge Boxes [192] to generate object proposals, since they provide a good trade-off between recall and speed [73]. Here we describe the process that we followed to choose the hyper-parameter N which is the number of candidate proposals.

We use the ILSVRC validation set, where the ground-truth bounding box is given, and we run the proposal algorithm for various values N. Every time we search among all N proposals for the one with the highest overlap with the ground truth and evaluate the network's performance on this region. We use different values of N, and two criteria to define the overlap: at the first five rows in Table 2.4 we use the Intersection over Union (IoU), which is the standard overlap metric in Pascal VOC, and at the last row we select the smallest-area box that completely contains the ground truth region in R^2.

In Table 2.4 we show that using bigger values of N yields larger maximum overlap with the

Criterion	Proposals	top-1 error	top-5 error
highest IoU with the GT	10	39.49	16.09
highest IoU with the GT	20	38.97	15.41
highest IoU with the GT	40	38.25	15.00
highest IoU with the GT	100	37.73	14.63
highest IoU with the GT	200	37.83	14.69
Smallest bb around the GT	200	38.51	15.26

Table 2.4: Evaluation of the proposed Edge Boxes by calculating the classification performance when the ground truth is known and the best available bounding box is selected accordingly. We use the Intersection-over-Union (IoU) as overlap criterion. More Edge Boxes provide as expected better cover of ground-truth objects and subsequently higher classification accuracy. However, they add computational overhead to our algorithm, which is linear to the number of proposals. On the last row we use a slightly different selection criterion, i.e., the smallest bounding box that encloses the ground-truth region. If there is no such proposal, we choose the whole image. This criterion yields higher error.

ground truth and subsequently a higher classification accuracy, as we would expect. Using more than 100 proposals seems to give no more benefits, aside the computational cost which is linear to the number of proposals. The number in Table 2.4 are not directly comparable with the statistics in the rest of the paper, because they are produced in a subset of validation set (first $2,000$ images) but nevertheless they give an estimation for our algorithm. In the last row we see that the alternative criterion gives inferior performance for the same number of proposals ($N = 200$).

Since the Edge Boxes are not tuned for the categorization task, we noticed that introducing a prior toward larger regions improves the performance. In practice we extract 200 proposals per image and we keep the $N = 80$ largest ones. Nevertheless, our method is not dependent on specific proposal method and we expect that parameter tuning in the proposal algorithm could further improve the classification accuracy end-to-end.

2.5 CNN vs. DSP-CNN while varying the object scale and context (occlusions)

As shown in Table 2.2, domain-size pooling with $D = 10$ seems to be a good prior for Imagenet data, as it reduces AlexNet's top-5 classification error by 19% compared to using the whole image. In Fig. 2.8 we show how DSP-CNN performs better compared to a single-domain CNN for different domain sizes that are located around the image center. This plot suggests that leveraging multiple domain sizes yields better performance than selecting any single domain size. DSP-CNN is quite insensitive to marginalizing the posteriors of smaller domains, while the performance of the single-domain model quickly degrades.

Next we compare how the single-domain CNN and DSP-CNN perform in two cases: first, when the object of interest is rescaled and the amount of context is fixed, and second, when the image is fixed and the amount of context varies by rescaling the input domain sizes. We show results on the validation set of ILSVRC 2014 as in Section 2.2.1.

First, we deal exclusively with the object scale, while we fix the context level. For this task we consider the ground-truth (gt) bounding box padded with a 50px rim as the *object*, as this gives

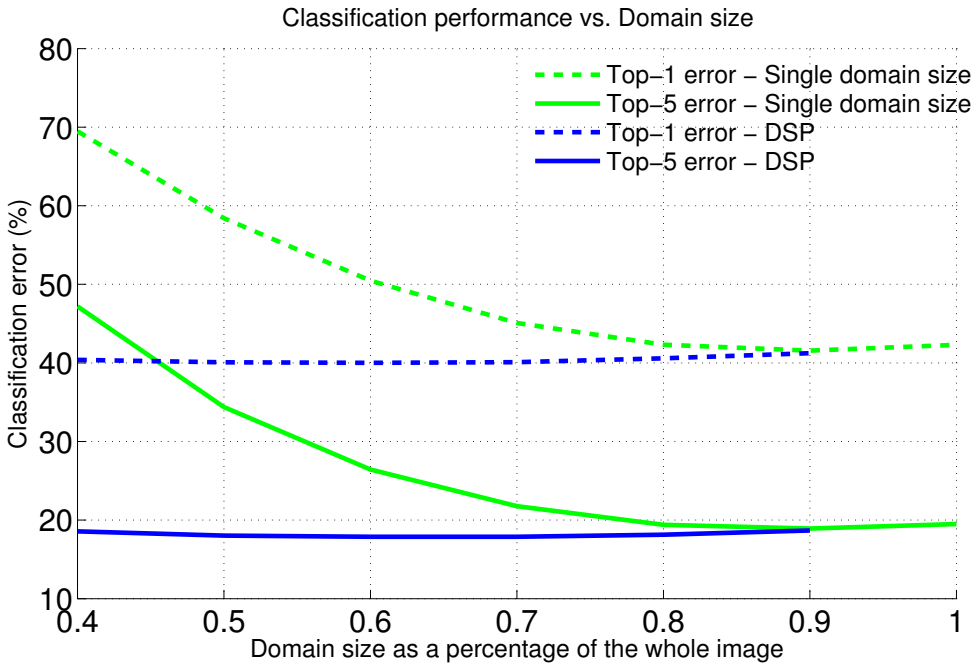
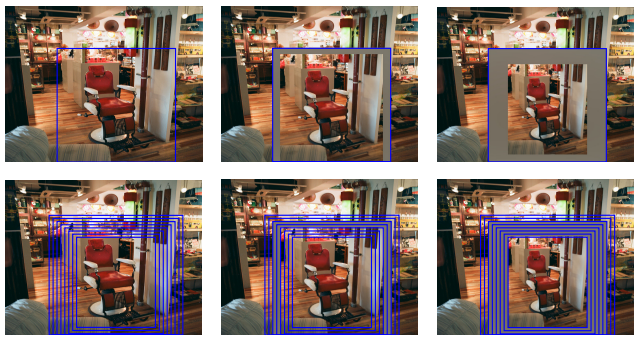


Figure 2.8: DSP on the whole image. We show the top-1 and top-5 classification error in Imagenet 2014 using various domain sizes which are located around the image center. The single domain sizes (green curves) are proportional to the whole image with ratio r, where $r \in [0.4, 1]$. The DSP method (blue curves) involves averaging of the posteriors while applying the network on $10 * (1 - r)$ domain sizes that are uniformly sampled in the range $[r, 1]$. We observe that the single-scale method has a fast diminishing accuracy when choosing smaller domain sizes, while DSP keeps yielding almost constant performance. The local minimum for a single domain size lies on $r = 0.9$ with top-1 and top-5 errors of 41.57% and 18.92%, while for DSP the best accuracy appears when sampling 5 domain sizes in $[0.6, 1]$ with 40.01% and 17.86% errors, respectively. This empirically validates our choice of using $D = 10$ (5 domain sizes and their horizontal flip) in Table 2.2. This experiment is agnostic to the location of objects within the image.

maximum classification accuracy as shown in Fig. 2.1. At the left of Fig. 2.9 we gradually shrink the *gt+50px* bounding box from full scale to *gt* size. At the right we compare the CNN with DSP-CNN for this range. For CNN we use the *gt+50px* model, which performs the best in Fig. 2.1. For DSP-CNN we use the model that samples 8 domain sizes in the $[0, 70]$ range, which is shown on the last row of Table 2.1. DSP-CNN outperforms CNN, and at the same time it is more insensitive as the object scale decreases. In the caption of Fig. 2.9 we describe the details of the evaluation.

Second, we show in Fig. 2.10 how varying the amount of context (or occlusions and clutter)



Original image 20 pixel shrinkage 50 pixel shrinkage

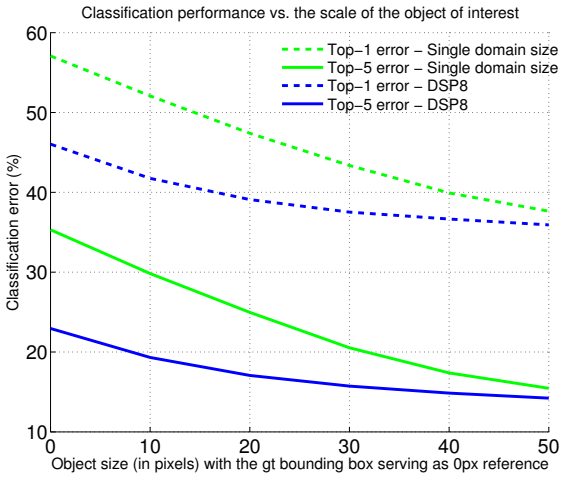


Figure 2.9: Object scale. Left: Shrinking the object in order to investigate the classification performance of CNN vs. DSP-CNN for various object scales. The object of interest for this task is defined as the ground-truth bounding box with 50px rim, as this provides the top accuracy (Fig. 2.1). Therefore, the object has 50 px rim in addition to the ground-truth size at its original scale, while the values between $[0, 50]$ pertain to its shrunk versions. The CNN is applied on the ground truth with 50px padding, as this gives empirically the higher classification accuracy (Fig. 2.1). Its DSP counterpart is applied on 8 domain sizes in $[0, 70]$, as it has been shown to be the top-performing method in Table 2.1. Right: The top-1 and top-5 classification error in Imagenet 2014 for increasing object scale (*i.e.,* the right value corresponds to the original scale). The background is not changing, while the freed space between the 50px rim and the receding object boundary is replaced by the average ILSVRC image in order to minimize any influence on the classifier. We observe that the DSP8 is more insensitive than the CNN for diminishing object scale.

influences the classification of the object of interest. Here the image is kept fixed, while the scale of the bounding box is changed proportionally to the ground truth. We use different domain sizes for CNN and a range around each of these sizes for DSP-CNN. The implementation details are described in the caption of Fig. 2.10. The samples are not augmented with horizontal flipping. Averaging the class posteriors achieves a better trade-off between context and nuisances and in turn produces a lower classification error.

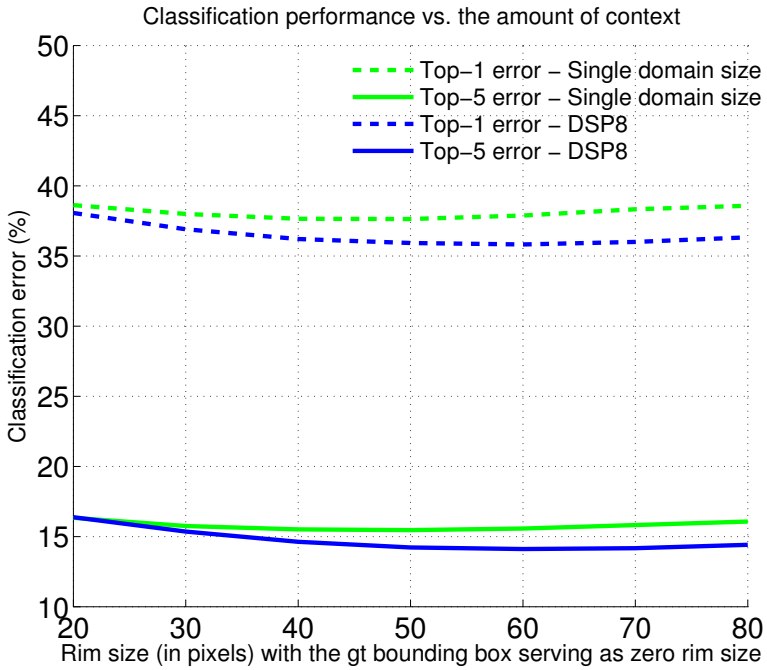


Figure 2.10: Occlusions. The top-1 and top-5 classification error in Imagenet 2014 for various domain sizes around the ground truth, while the image is kept fixed. The green curves pertain to testing with a single domain size with rim size r, while the blue curves correspond to averaging the posteriors of 8 domain sizes in the $[r-50, r+20]$ span. As for the single-scale case, this plot can be seen as a subset of Fig. 2.1, where the local minimum is on $50px$ for a 15.46% top-5 error. Here we show that the DSP8 consistently outperforms the single-scale method for various level of context (or occlusions). DSP's local minimum is on $r = 60$, i.e. averaging the posteriors of 8 domain sizes in $[10, 80]$, which gives top-5 error of 14.11%. This is marginally smaller than the error of DSP when it is sampled in $[0, 70]$ (Table 2.1).

2.6 Dense testing

In this section we want to investigate whether we can incorporate our sampling technique based on the posterior entropy in a dense testing setting and achieve performance gains in a fraction of time compared to the adaptive sampling scheme that is decribed in Section 2.2.1.

Following [146], we convert a standard convolutional neural network (e.g. [138, 146]) with convolutional and fully-connected layers into a fully-convolutional network. In practice [165, 79], since both fully-connected (FC) and convolutional (CONV) layers are computed as dot products, we can convert a FC layer to CONV either by shrinking the kernels or by providing larger feature

maps. Effectively this can be done by using a larger input image and setting the stride as 1 in the first FC layer. Therefore the FC kernel is computed densely like being a CONV operator and yields several posteriors. For example assume that instead of using 227×227 input images, we provide 419×419 images. Then the output of first FC layer will be $7 \times 7 \times 4,096$ instead of $1 \times 1 \times 4,096$. As we do not modify the remaining upper FC layers, they are effectively 1×1 CONV filters with stride 1 and depth $4,096$. Therefore, in the previous example, the network will output a 7×7 class score map.

On the top of the fully convolutional network we add one pooling layer in order to extract a single class posterior for the whole image. Posterior selection can be performed based on R$\acute{e}$nyi entropy, similarly as we did when using region proposals. Extension over scale is performed as well. Therefore, we use various images sizes and extract posterior maps of different sizes, which we aggregate before the entropy selection. In specific, we use image sizes 355×355, 419×419 and 483×483, which return 5×5, 7×7 and 9×9 class score maps respectively (which sum up to 310 posteriors in total).

Method	VGG16			eval-S	ave-S
	top-1	top-5	t(10^3s)		
Testing with regular crops [146]	27.14	8.85	174	150	150
Dense testing [146]	27.09	8.61	21	7x7x2	7x7x2
Testing with proposals	**25.11**	8.01	219	180	32
Dense Selective testing (translation)	26.37	8.10	24	80	98
Dense Selective testing (translation, scale)	25.93	**7.77**	**25**	310	200

Table 2.5: Top-1 and top-5 errors on the Imagenet 2014 classification challenge [136]. The rows 1-2 show previous methods in the literature, which serve as our baselines. Row 3 shows adaptive sampling as it is performed in Section 2.2.1, while next we demonstrate dense testing with posterior selection over translation (row 4) and over both translation and scale (row 5). *Eval-S* is the number of the evaluated posteriors and *ave-S* is the posterior vectors that are eventually averaged to produce one single prediction. Dense testing runs in a fraction of time, as several posteriors are extracted with one (or few) pass.

The results are shown in Table 2.5. We achieve comparable performance improvements as we received with adaptive proposal sampling in a fraction of time. The top5 error is further reduced, but in terms of the top1 error the adaptive sampling with proposals remains the best method. Strictly speaking, the complexity increases linearly to the number of convolutions on first FC layer for all testing scales and aspect ratios (plus the ratio of more CONV operations at the lower layers which adds a constant overhead). In practice, the speedup is large compared to multi-crop and testing with proposals, because the entire score map is computed in 1 pass (or as many passes as the number of tested scales and aspect ratios) given that the entire model fits in the GPU.

Someone may assume that dense testing renders the need for crops moot. However, it is possible that combining dense testing with adaptive samples (proposals) could yield further improvement. Additionally, the top-1 accuracy with proposals is still better than the one obtained with dense testing. It is worthwhile to stress that when applying a ConvNet to a crop, the convolved feature maps are padded with zeros, while in dense evaluation the padding for the same region naturally comes from the neighboring pixels (due to both the convolutions and spatial pooling), which substantially increases the effective network receptive field, and in turn the captured content.

2.7 Pascal VOC Detection

Next we perform comparisons on Pascal VOC 2007 detection challenge. Here the element of the group transformation (bounding box) is no longer the nuisance, but the object of interest. Therefore, we expect that averaging the class posteriors of neighboring regions will hurt the localization accuracy. However, there is a trade-off, as the domain-size pooling can improve the discriminability around the object of interest in terms of categorization. In the following, we put the challenge to the test. After, we show how searching the most "interesting" domain size in the scale-neighborhood of the proposals based on the entropy of the class posteriors can improve the detection performance end to end.

We use Regions with CNNs [60] as a baseline. This algorithm includes three main steps: first, generic object proposals are extracted using Selective Search [164], then a CNN is used to classify each proposal, and finally an optional regression step improves the localization of the

Method	mAP	mAUC
Regions with CNNs [60]	54.16	54.30
Domain-size averaging on the proposals	41.47	40.33
Domain-size averaging with entropy selection	51.21	51.08
Domain-size selection with entropy criterion	**54.36**	**54.70**

Table 2.6: Mean Average Precision (mAP) and mean Area Under the Curve (mAUC) for R-CNN's [60] variants on Pascal VOC 2007.

output predictions. There are many intermediate steps that are equally important for the algorithm to perform well, such as greedy non-maximum suppression for the candidate bounding boxes in order to avoid duplicate predictions. We keep all factors constant and use the pre-trained models for CNN and Support Vectors Machines, as they are provided by the authors of [60]. All variants are compared without the optional regression step at the end. We use the test set and the evaluation protocol of Pascal VOC 2007 challenge.

The mean Average Precision and mean Area Under the Curve achieved by [60] are shown on the first row in Table 2.6. We attempt domain-size pooling to classify the proposals and a criterion for domain size selection based on the entropy, similar in principle to the one that we use in classification (Section 2.2.1). We experiment with each method using three domain sizes for each proposal: the proposed bounding box with Selective Search padded with $16px$ as in [60] and two concentric bounding boxes to it with $15px$ and $30px$ additional padding. On the second row we show that domain-size average pooling lowers the average precision for detection, which is to be expected as it hurts the localization. On the third row we average the posterior over the base domain size and any additional domain size that has lower entropy than the base one, if any. This is still inferior compared to using no pooling at all. These methods may help the discriminability around the object of interest, but they deteriorate the localization accuracy and reduce the end-to-end performance. Finally, on the forth row we show how selecting the neighbor with the lowest entropy for each proposal improves the detection performance compared to the baseline. This is to be expected as it can be seen as a method for refining the proposal algorithm. Although we sample

only three elements from the scale group, our method can be easily extended to include several samples from the location-scale or even the affine group.

2.8 Performance profile of DSP-CNN vs. CNN for wide-baseline correspondence

The Fischer dataset [54] includes 400 pairs of images, some of them with more extreme transformations than those in the Oxford dataset [119]. The types of transformations include zooming, blurring, lighting change, rotation, perspective and nonlinear transformations. In Fig. 2.11 we present the matching performance for different magnitude of various transformations. Finally, we show the 5 best and the 5 worst pairs (among all 400 pairs in Fischer data) in terms of DSP-CNN vs. CNN relative performance.

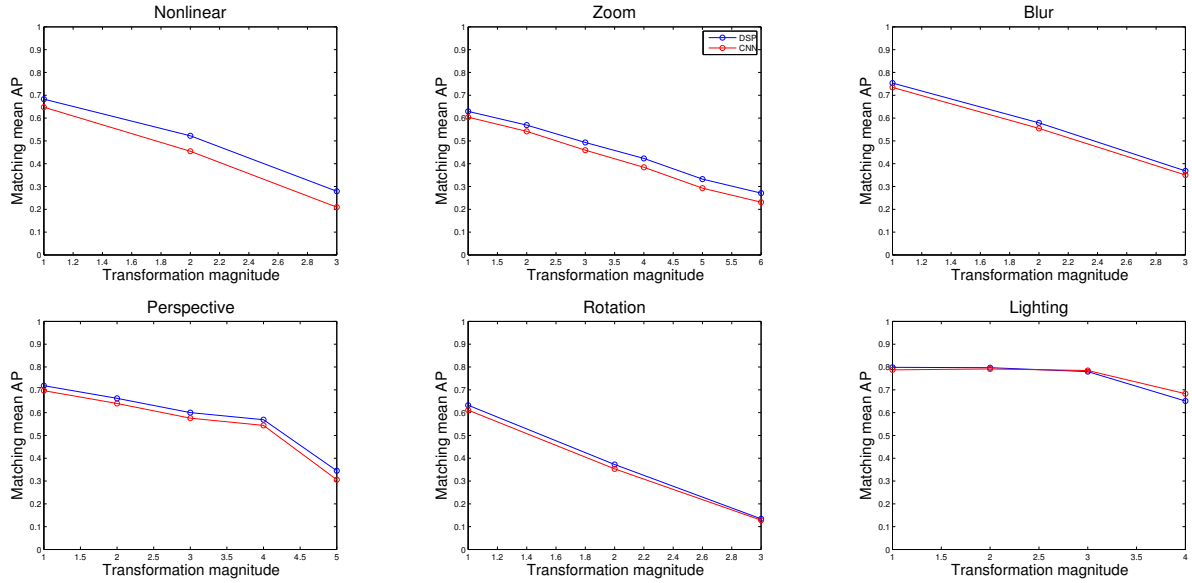


Figure 2.11: Matching mean Average Precision (mAP) for different magnitude of transformations in the Fischer dataset. The largest benefits of deploying domain-size pooling appear for nonlinear transformations, while there is consistent improvement for zoom, blur, perspective and rotation. Finally, as it should be expected, this technique does not help with illumination variation. Actually, averaging the class posteriors slightly reduces the discriminability for large lighting changes.

Figure 2.12: Pairs with the best improvement of DSP-CNN over CNN in Fischer data. For each pair over the arrow we write the transformation and its corresponding maginitude. Under the arrow is the absolute mAP increase. DSP-CNN is especially robust with non-linear local deformations.

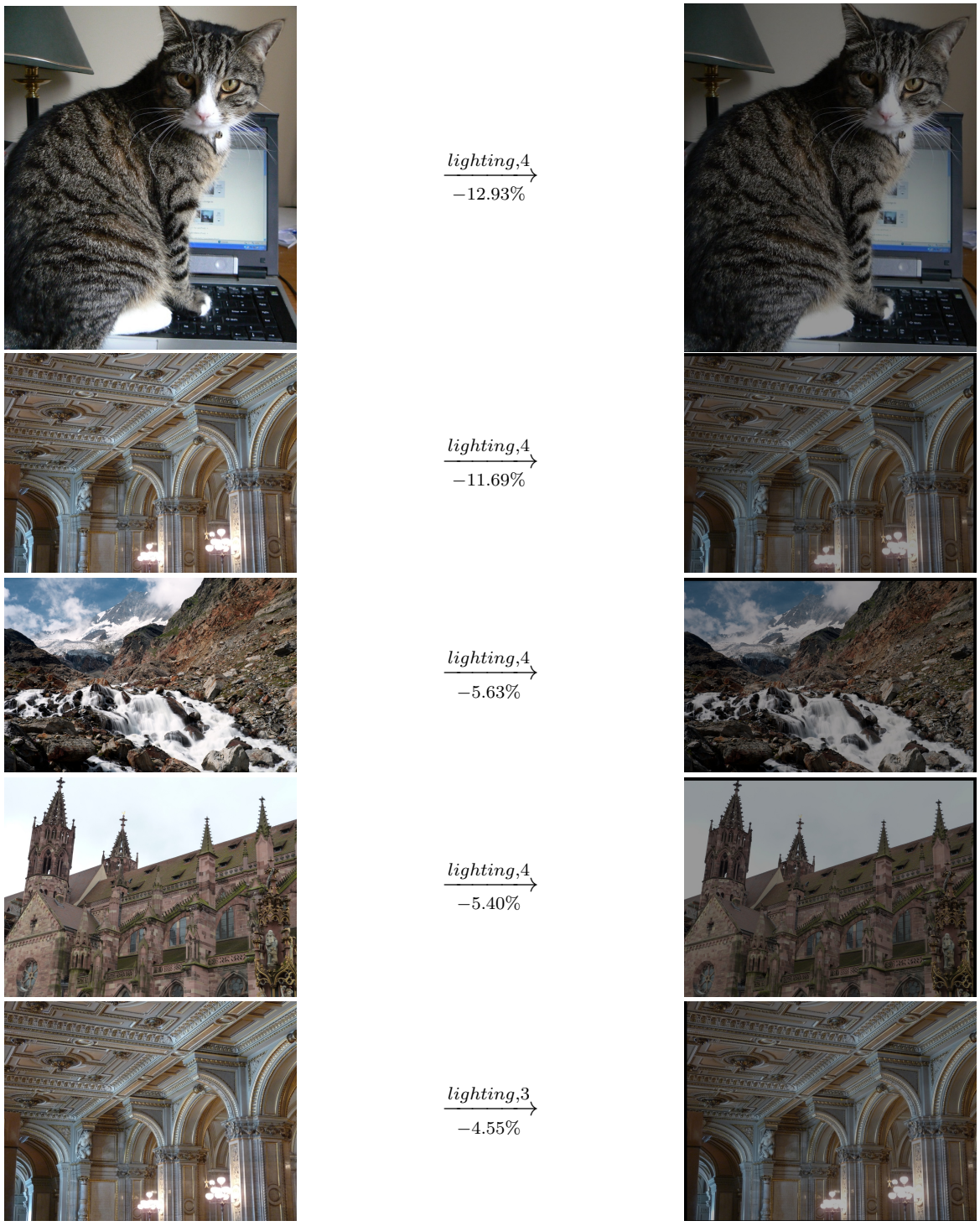


Figure 2.13: Pairs where DSP-CNN performs the worst compared to CNN in Fischer data. For each pair over the arrow we write the transformation and its corresponding magnitude. Under the arrow is the absolute mAP decrease. It is expected that the domain-size pooling does not help with illumination variation, which is confirmed by Fig. 2.11.

2.9 Extended Discussion

This section expands on the discussion of our conclusions in the previous sections.

Is a CNN really (meant to be) computing class posteriors? A CNN produces, at its penultimate layer, just before the classifier, relative scores for each class $c = [c_1, \ldots, c_N]$ in response to a particular instance of the data I. This can be interpreted as the likelihood of each class $L(c) \propto P(I|c)$. Once weighted by the prior probability of each class and normalized, this yields the class posterior $P(c|I)$. For the purpose of testing the hypothesis underlying our investigation, which concerns the marginalization of nuisance variables, class priors are irrelevant (since nuisances do not enter the prior), so we consider equivalently the class likelihoods $P(I|c)$ or class posteriors $P(c|I)$.

Is a CNN really (supposed to be) marginalizing nuisance groups? The class posterior (or likelihood) links the data I to the class c. However, a class c manifests itself in the data through a particular *instance* of object of class c, imaged under particular imaging conditions – including position, scale, aspect ratio and other unknown (nuisance) variables. There is, therefore, a Markov-chain dependency between the class c, the particular object instance, and the data I, mediated by nuisance variables g. Since the CNN produces an estimate or approximation of $P(I|c)$, such nuisance variability has to be managed *somehow* by the CNN. The key question, then, is exactly *how* the CNN manages it. For *planar translation*, the CNN narrative suggests that the *structure* of the network is designed to (approximately) marginalize it, by averaging scores computed equi-variantly at each location. For everything else, the assumption is that nuisance variability is *learned away* by the network through supervision (instances that share the same class but different nuisances are labeled as the same, which shapes the residual surface of the CNN).

What do you mean by conditionals? Posteriors? Marginals? As discussed above, the CNN returns class scores, that can be interpreted as either class likelihoods or, after normalization, class posteriors $P(c|I)$. When referring to "conditionals" or "marginals" we always refer to the variables of interest in this paper, that are nuisance variables. Thus, by condition-

als we mean $P(c|I,g)$ or $P(I|c,g)$, where $g \in G$ is the (nuisance) conditioning variable. Similarly, by marginals, we mean the probabilities that marginalize $g \in G$, for instance $P(I|c) = \int_G P(I|c,g)dP(g)$ for a continuous group G.

Averaging scores of few crops is a lousy approximation of proper marginalization. Indeed, if a network trained on the whole image is thought to approximate $p(c|x)$, when tested on a proposal $r \subseteq x$ determined by a reference frame g_r, it computes $p(c|x_{|r})$ (x restricted to r), which is different from $p(c|g_r, x)$, as correctly pointed out by the Reviewer. Then, explicit marginalization computes $\frac{1}{|r|}\sum_r p(c|x_{|r})$ which is different from $\frac{1}{|r|}\sum_r p(c|g_r, x)$ which in turn is different from $\sum_r p(c|g_r, x)p(g_r|x)$, which would be needed to approximate the proper marginal $p(c|x) = \int p(c|g,x)dP(g|x)$. This approach is therefore, on average, a lower bound on proper marginalization, and the fact that it would outperform implicit marginalization is surprising indeed and worth investigating empirically.

No conclusion *in expectation* can be drawn from finite sample averages. Indeed marginalization entails the computation of a continuous integral, which can be only *approximated* by a CNN. Nevertheless, all other factors being equal (weights of the CNN, training set, etc.), the fact that the CNN restricted to, and average across, few bounding boxes performs better indicates that the approximation computed by the CNN is not very effective.

Where is averaging performed to approximate marginalization? At the output of the network (penultimate layer), so as to averaged are the class-conditional distributions.

Why averaging discriminants? That is counterproductive. Indeed, in general, averaging reduces discriminative power, increasing false alarms. Averaging *along directions spanned by nuisance variables*, corresponding to marginalization, however, reduces sensitivity to nuisance transformation, thus reducing missed detections. So averaging trades off discriminative power with insensitivity to nuisance variability. Our purpose is indeed to quantify such a tradeoff.

In practice, our results show that such averaging has negligible effects on discriminative power. This may be due to the fact that the ambient space of most descriptors is so high-

dimensional that even multiple categories (such as those in ImageNet) are so far from each other that local averaging causes no false alarms. Instead, local marginalization is generally beneficial to reduce sensitivity to nuisance factors for a constant number of training samples.

Reducing context should reduce performance. Fig. 2.1 measures the tradeoff between *conditioning* (feeding the CNN the "true" location, scale and aspect ratio of the object of interest, which should *improve* performance, by reducing the entropy of the class *on average*), and *context* removing which should *reduce* performance. The fact that restricting to a bounding box reduces performance is obvious; that this happens even with the *ground truth* bounding box (for AlexNet) is less obvious and indicative of the trade-off between context and nuisance variability (e.g., the classification accuracy improves 34% compared to the ground truth for AlexNet when we pad with 50 pixels). The fact that padding the bounding box with 50 pixels *improves* performance by a large margin (27% compared to using the whole image for AlexNet) is non-obvious and indicative of the inability of the CNN to capture context beyond a few pixels.

This is not obvious as a CNN, in principle, has the ability to capture co-occurrence statistics on the entire image domain, since the "receptive field" (regions of the image plane) subtended by filters at higher layers encompass a large area of the image. However, the experiments conducted indicate that the CNN is not effectively leveraging such context. This is shown in three steps: First, the baseline performance is *comparable* (slightly lower for AlexNet, slightly higher for VGG16) to restricting the image to a bounding box containing the object of interest. Second, the baseline performance *increases* if the image is restricted to the bounding box plus a small rim around it, suggesting that the network indeed can leverage some context. Third, continuing to increase the rim size only hurts the classification accuracy. Fig. 2.1 shows results for different padding sizes.

Entropy reduction and mutual information between the class and the data. Throughout the paper, "entropy" refers to the entropy of the posterior, that is of the class c conditioned on the data I, $\mathbb{H}(c|I)$. Reduction of this conditional entropy is equivalent to an increase in mutual information between I and c, for $\mathbb{I}(c; I) = \mathbb{H}(c) - \mathbb{H}(c|I)$.

CNNs are not designed for wide-baseline matching. Indeed, although they have been used for that purpose [54], so an objective benchmark is available. The deployed network is trained to be invariant to several transformations that are commonly observed in images captured from different viewpoints. Therefore it approximates the properties of an invariant descriptor via learning, as opposed to networks which are trained on the classification task and aim at grouping all instances of same classes, thus totally losing their generative power at their last layer.

For that task, denoising auto-encoders, or RBMs, would seem better suited, and indeed have been used to face matching tasks [157]. However, [41] shows that Gated RBMs perform worse than local descriptors in wide-baseline matching.

How do bounding boxes define group transformations? The center of the bounding box defines a position on the pixel lattice (u, v), assumed to belong to the continuum after interpolation, $(u, v) \in \mathbb{R}^2$. The two sides define units around the coordinate axes (σ_1, σ_2), with $\sigma_1, \sigma_2 > 0$. These four-parameters can be considered as an element of the anisotropic location-scale group, that transforms every point on the plane $x = (x_1, x_2)$ via $gx = y$ where $y_1 = \sigma_1 + u$ and $y_2 = \sigma_2 + v$. The group has a null element $u = v = 0; \sigma_1 = \sigma_2 = 1$ and an inverse on the real plane, regardless of whether the image is defined there: $g^{-1}(y) = x$ where $x_1 = (y_1 - u)/\sigma_1$ and $x_2 = (y_2 - v)/\sigma_2$. The image can be extended to the plane by zero-padding. This construction extends to rotated bounding boxes (similarity group), parallelograms (affine) and arbitrary convex quadrilaterals (projective group). Thus, sampling bounding boxes on the plane corresponds to sampling elements of one of these groups, depending on the geometry of the bounding boxes. In our case, we are restricting ourselves to the location-scale (anisotropic) group, hence corresponding to sampling rectangular, axis-aligned bounding boxes.

How is averaging bounding boxes "anti-aliasing"? We use this term as characterized by [43]. Anti-aliasing generally refers to the removal of "aliases", or spurious extrema in the signal reconstructed from samples that are not present in the original (pre-sampling) continuous signal. In signal processing, under the assumptions of ℓ^2 integrability and (at least local)

stationarity, so one can talk about "frequency", anti-aliasing is usually performed by convolving the signal with a "low-pass" filter that removes higher frequencies thus limiting the signal to a frequency "band". Such convolution is a local average of samples, weighted by a kernel that can be designed to have sharpest frequency cut-off (*e.g.,* the sinc function), or other criteria such as optimal trade-off between spatial and frequency support (*e.g.,* Gaussian kernels). In our context there is no assumption of stationarity, and we do not design the averaging procedure for optimality, but instead perform local averaging with respect to a uniform kernel, a crude version of anti-aliasing that is, however, sufficient to improve performance thus supporting *a fortiori* the conclusions on its effects. Note that what is being averaged, or anti-aliased, is not the samples from the posterior, but the posterior itself, thus this is a somewhat unusual (generalized) sampling scenario where each sample is an integrable function.

Why PCA? The reduction of dimensionality performed to compare the CNN to a small-dimensional descriptor such as SIFT could be performed in a number of ways. PCA is the simplest, not necessarily the best, as it does not capture the discriminative subspace, but the representative subspace instead.

2.10 Final remarks

Our empirical analysis indicates that CNNs, that are designed to be invariant to nuisance variability due to small planar translations – by virtue of their convolutional architecture and local spatial pooling – and learned to manage global translation, distance (scale) and shape (aspect ratio) variability by means of large annotated datasets, in practice are less effective than a naive and in theory counter-productive practice of sampling and averaging the conditionals based on an ad-hoc choice of bounding boxes and their corresponding planar translation, scale and aspect ratio.

This has to be taken with the due caveats: First, we have shown the statement empirically for *few* choices of network architectures (AlexNet and VGG), trained on *particular* datasets that are unlikely to be representative of the complexity of visual scenes (although they may be representative of the same scenes as portrayed in the test set), and with a specific choice of *parameters* made by their respective authors, both for the classifier and for the evaluation protocol. To test the

hypothesis in the fairest possible setting, we have kept all these choices constant while comparing a CNN trained, in theory, to "marginalize" the nuisances thus described, with the same applied to bounding boxes provided by a proposal mechanism. To address the arbitrary choice of proposals, we have employed those used in the current state-of-the-art methods, but we have found the results representative of other choices of proposals.

In addition to answering the question posed in the introduction, along the way we have shown that by framing the marginalization of nuisance variables as the averaging of a *sub-sampling* of marginal distributions we can leverage of concepts from classical sampling theory to *anti-alias* the overall classifier, which leads to a performance improvement both in categorization, as measured in the ImageNet benchmark, and correspondence, as measured in the Oxford and Fischer's matching benchmarks.

Of course, like any universal approximator, a CNN can in principle capture the geometry of the discriminant surface by "learning away" nuisance variability, given sufficient resources in terms of layers, number of filters, and number of training samples. So in the abstract sense a CNN *can* indeed marginalize out nuisance variability. The analysis conducted show that, at the level of complexity imposed by current architectures and training set, it does so less effectively than ad-hoc averaging of proposal distributions.

This leaves researchers the choice of investing more effort in the design of proposal mechanisms [60], subtracting duties from the Category CNN downstream, or invest more effort in scaling up the size and efficiency of learning algorithms for general CNNs so as to render the need for a proposal scheme moot.

CHAPTER 3

Boosting Convolutional Features for Robust Object Proposals

3.1 Introduction

Convolutional neural networks (CNNs), the de-facto standard for object detection in images, can in principle manage nuisance variability due to (planar) position, scale and aspect ratio [91]. However, the quest for top performance in [136] has led researchers away from letting the CNN manage all such variability, favoring instead split pipelines whereby the image is first pre-processed to yield *proposals*, which are subsets of the image domain (bounding boxes) to be tested for the presence of an object class by a "Category CNN." The output of a proposal algorithm is a collection of bounding boxes, or equivalently a sampling of the (anisotropic) translation-scale group of transformations,[1] each corresponding to a proposal, or hypothesis, of object.

The best proposal algorithm is the one that densely samples the group. However, even for small-dimensional transformations such as the anisotropic translation-scale group, a single image could yield billions of proposals. As in any sampling procedure, the goal is to trade off performance with complexity. To this end, *adaptive sampling* schemes can be employed to select proposals based on the data. Then a proposal algorithm takes as input an image, and produces as output sample transformations, in our case bounding boxes, using a binary classifier for each value of the transformation. Such a selection amounts to a *premature decision* that can only decrease the performance in the overall task, which we accept in exchange for gains in run-time. To minimize damage, a proposal should yield few if any missed detection (if an object is not proposed, it will

[1]The translation component is the center of the bounding box, and the two scales are the sides of the box. More general groups could be considered, such as similarity, affine, or projective.

never be found[2]), even if at the cost of many false alarms. In other words, the classifier is tasked not so much with selecting regions that are highly likely to contain objects, but with discarding as many regions as possible that, with high confidence, do not. A further caveat in developing proposals is that the Category CNN discards the image outside the proposal window, thus possibly forgoing side information or "context."

Since a proposal algorithm involves the design of a classifier, whose results are to be fed to another classifier (a Category CNN), it seems natural to want to leverage on the latter to design the former. For instance, if the Category CNN computes a multi-class posterior with pose, scale and aspect-ratio marginalized, we could recycle its powerful components to produce a binary distribution (object vs. not) for a *given* pose, scale and aspect ratio, by marginalizing the classes. This is done simply by feeding the image restricted to each sample transformation (a bounding box) to the Category CNN, followed by averaging. This raises a conundrum: If the Category CNN could indeed marginalize them, conditioning on (estimated) transformations would be detrimental, a consequence of the Data Processing Inequality [34]. The fact that Regions-with-CNN pipelines outperform CNNs alone suggests that the latter may not be as effective at marginalizing nuisances, which has been also shown by Karianakis et al. [85]. If we accept this as a fact, then most of the effort should focus on the proposal algorithm, since that is tasked with removing the most challenging variability in visual data [155, 130]. This is what we do.

We introduce a proposal scheme (Fig. 3.1) that employs features already learned by a Category CNN to design a boosting binary classifier that labels bounding boxes sampled densely in location, and coarsely in scale and aspect ratio, as "object" or "background" (Sect. 3.2). We further apply linear regression to refine the location of the top-scoring bounding boxes. In Sect. 3.3 we benchmark our scheme against several proposal algorithms, using standard evaluation protocols, in both performance and computational cost. We also test our scheme on an end-to-end task using ImageNet's localization challenge [136], by swapping a state-of-the-art proposal scheme with ours, with measurable improvements.

[2]For instance, Selective Search proposes on average $2,403$ regions per image in [60] with 91.6% recall, meaning that even with an oracle detection algorithm, 8% of all object instances would go undetected.

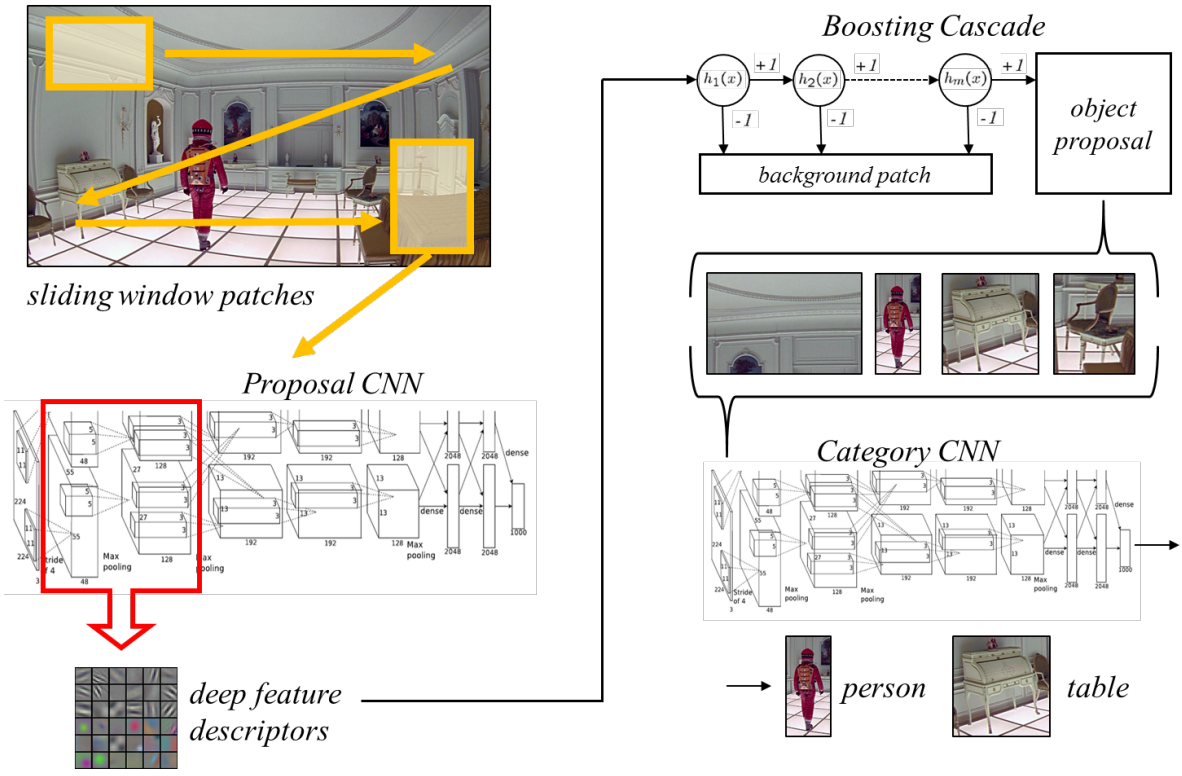


Figure 3.1: Processing pipeline for boosting convolutional features. Regions corresponding to objects and background are extracted from training images. Convolutional responses from first layers of a Proposal CNN are used to describe these patches, and fed to a boosting model to learn an object/background classifier. Finally a Category CNN is employed to classify each proposal into one of many object categories.

3.1.1 Prior work

We briefly review some representative methods which are evaluated in detail in [73].

Selective Search [164], which is currently the most popular algorithm, has its features and score functions engineered on Pascal VOC [48] and ILSVRC [136] so that low-level superpixels [51] are gradually merged to represent high-level objects in a greedy fashion. It achieves very high localization accuracy due to the initial over-segmentation at a time overhead. *RandomizedPrim's* [111] is similar to Selective Search in terms of features and the process of merging superpixels. However, the weights of the merging function are learned and the whole merging process is randomized.

There is a family of algorithms which invest significant time in a good high-level segmenta-

tion. *Constrained Parametric Min-Cuts (CPMC)* [21] generates a set of overlapping segments. Each proposal segment is the solution of a binary segmentation problem. Up to $10,000$ segments are generated per image, which are subsequently ranked by objectness using a trained regressor. Rantalankila et al. [133], similar in principle to [164] and [21], merge a large pool of features in a hierarchical way starting from superpixels. It generates several segments via seeds like CPMC does. Endres et al. [46] combine a large set of cues and deploy a hierarchical segmentation scheme. Additionally, they learn a regressor to estimate boundaries between surfaces with different orientations. They use graph cuts with different seeds and parameters to generate diverse segments similar to CPMC. *Multiscale Combinatorial Grouping (MCG)* [10] combines efficient normalized cuts and CPMC [21] and achieves competitive results within a reasonable time budget.

In the literature several methods attempt to quantify *Objectness* [3, 132] based on a combination of saliency, color contrast, edge density, location and size statistics, and the overlap of proposed regions with superpixels. Data-driven Objectness [84] is a practical method where the likelihood of an image corresponding to a scene object is estimated through comparisons with large collections of example object patches.

Binarized Normed Gradients for Objectness (BING) [29] is a simple linear classifier over edge features and is used in a sliding window manner. In stark contrast to most other methods, BING takes on average only 0.2s per image on a CPU. *EdgeBoxes* [192] is similar in spirit to BING: A scoring function is evaluated in a sliding window manner, with object boundary estimates and features which are obtained via structured decision forests.

Scalable, High-quality Object Detection [159] is also data driven, as an evolution of [47] that integrates region proposals and classification in one step. By deploying an ensemble of models with robust loss function and their newly introduced "contextual features", they achieve state-of-the-art performance on the detection task.

3.2 Methodology

Features and Boosting: We use binary boosting to train a classifier with output $y_i \in \{1, -1\}$ (object/background) for each image patch $i \in \{1, \ldots, N\}$. The input samples x_i are feature vectors

which describe an image patch i. The features x_i are a subset of convolutional responses $conv_j^{k_j}$ from a Proposal CNN (cf. Fig. 3.1), where j pertains to convolutional layer $j \in \{1, \ldots, L\}$ and k_j spans the number of feature maps for this layer (e.g., *alexNet* [91] uses $L = 5$ and $k_1 \in \{1, \ldots, 96\}$). Our Proposal CNN is the *VGGs* model from [26], whose first-layer convolutional responses are 110×110 pixels, double the resolution of alexNet.

Aggregate-channel features from [38] are used, where convolutional responses serve the role of channels, while we deploy a modified version of the fast setting provided by [9]. Thus, efficient AdaBoost [167] is used to train and combine $2,048$ depth-two trees over the $d \times d \times F$ candidate features (channel pixel lookups), where d is the baseline classifier's size and F is the number of convolutional responses, *i.e.,* the patch descriptors (*e.g.,* VGGs architecture has $F = 96$ kernels in the first layer). The convolutional responses from all positive and negative patches which are extracted from the training set are rescaled to a fixed $d \times d$ size (in our case $d = 25$) before they serve as input to the classifier. In practice, classifiers with various d can be trained to capture different resolutions of these representations. On testing all classifiers are applied to the raw image and their detections are aggregated and non-maximally suppressed jointly.

Mining samples: We train the classifier with positive and negative samples extracted from Pascal 2007 VOC [48]. Positive samples are the ones that correspond to the ground truth annotated objects, while negatives are defined as rectangular samples randomly extracted from the training set at different scales and aspect ratios, having less than 0.3 intersection-over-union (IoU) overlap with the positives. For testing, we use patches sampled from the validation sets of VOC 2007 and ImageNet 2013 (detection), both exhaustively annotated.[3] There is some margin for improvement by more sophisticated sampling of negatives on VOC and ILSVRC, since their annotation does not include all possible object classes that can appear in an image.

While using hierarchical feature responses, annotations have to be mapped to corresponding regions in each filter map while taking into consideration the padding parameters and kernel sizes for specific network architecture.

[3]All object instances of C classes are fully annotated, with $C = 20$ for Pascal, and $C = 200$ for ImageNet Detection, so we can test whether a negative candidate impinges on actual objects.

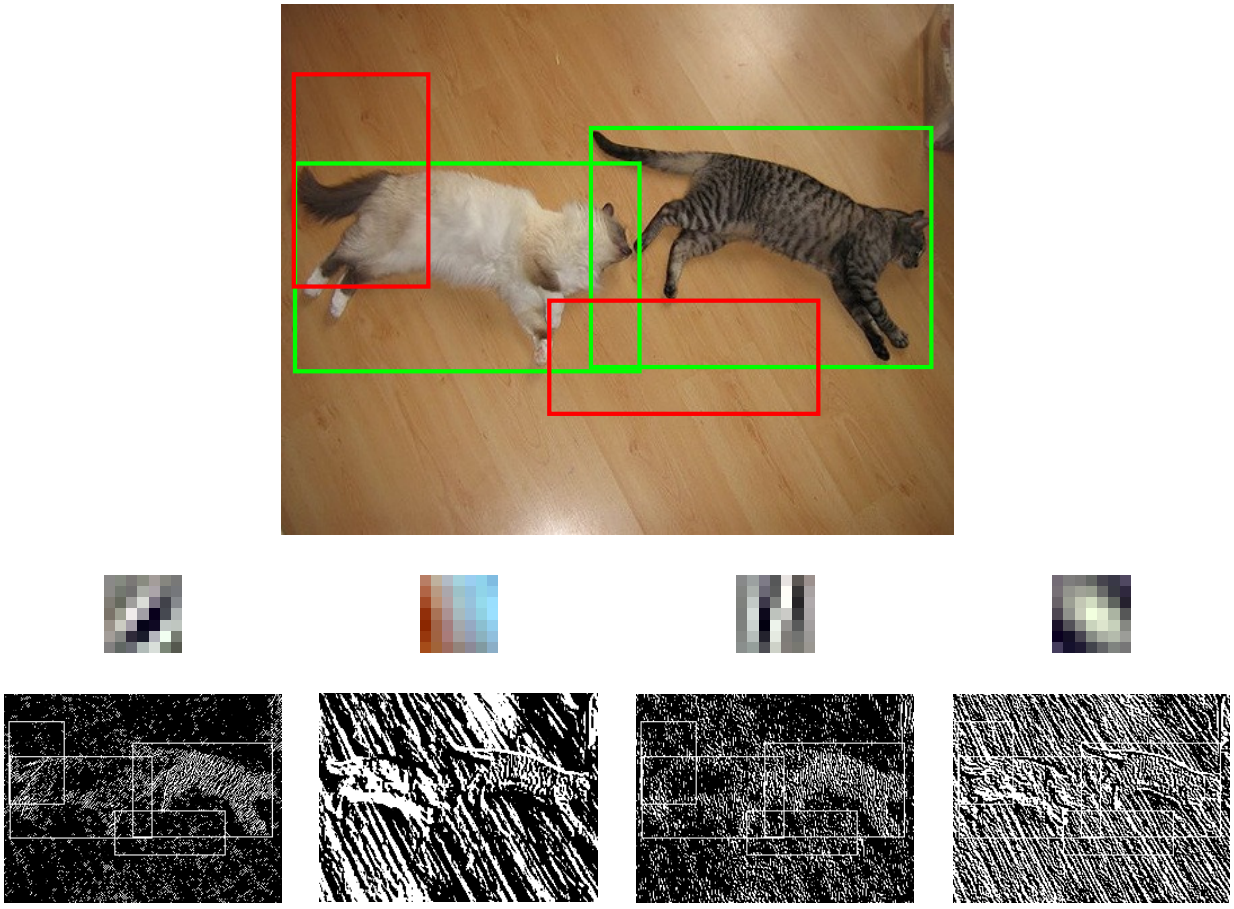


Figure 3.2: An image from Pascal VOC and its convolutional responses with a subset of first-layer filters. In order to classify object candidates, a binary boosting framework is trained with positive (green) and negative (red) samples which are extracted from CNN's lower layers.

Connection with EdgeBoxes [192]: After testing filter responses from several layers, we have found that using only the first convolutional responses before any spatial pooling is applied yields the best performance. Given this choice and the nature of first-layer filters (Fig. 3.1), our method ends up relating to BING [29] and EdgeBoxes [192], which use edge features, and methods that use color similarity (e.g., Selective Search). This family of methods provides quick detection, as the time-consuming high-level segmentation is avoided. On the other side, omitting a segmentation step has the drawback that it is less likely to generate proposals which are well-aligned with the object boundaries. Nevertheless, a subsequent regression step which leverages on image information and features from the upper convolutional layers can alleviate this shortcoming.

Testing: We apply the learned classifier on a sliding window centered at every pixel, with S different scales and R aspect ratios. We choose $S = 12$ scales and $R = 3$ aspect ratios. Non-maximum suppression (NMS) is used to reject detections with more than U IoU overlap for every (scale, aspect ratio) combination. Finally, after detections from all scales and aspect ratios are aggregated, another joint NMS with V IoU is applied. We experimented with different parameters and after cross validation we use $U = 63\%$ and $V = 90\%$ to report results in Fig. 3.3.

Bounding-Box Regression: After extracting the proposals, a regression step can be applied to refine their location. As proposed by [60], a linear regressor is used with regularization constant $\lambda = 1,000$. For training we use all ground truth annotations G^i and our best detection P^i per ground truth for all training images $i, i \in \{1, \ldots, N\}$ from Pascal VOC 2007. The best detection is defined as the one with the highest overlap with the ground truth. We throw away pairs with less than 70% IoU overlap. The goal of the regressor is to learn how to shift the locations of P towards G given the description of detected bounding box ϕ. The transformations are modeled as linear functions of $pool_5$ features, which are obtained by forward propagating the P regions through the Proposal CNN.

3.3 Experiments

We adopt ImageNet 2013 (detection) [136] as a test benchmark. Note that the boosting classifier and the regressor are instead trained on Pascal 2007 VOC. We follow the evaluation protocol proposed by [73] and report the performance vs. localization accuracy (Fig. 3.3) and vs. number of candidates per image (Fig. 3.4). Specifically, we calculate the recall of ground-truth objects for various localization thresholds using the IoU criterion, as it is customary on Pascal VOC. In Fig. 3.3 we report performance compared to state-of-the-art methods and three baselines, as evaluated by [73]. Each algorithm is allowed to propose, on average, up to $10,000$ regions per image. The methods are sorted based on the Area-Under-the-Curve (AUC), while in parentheses we report the average number of proposed regions per image. A small subset of images have been blacklisted in the evaluation process per ILSVRC policy.

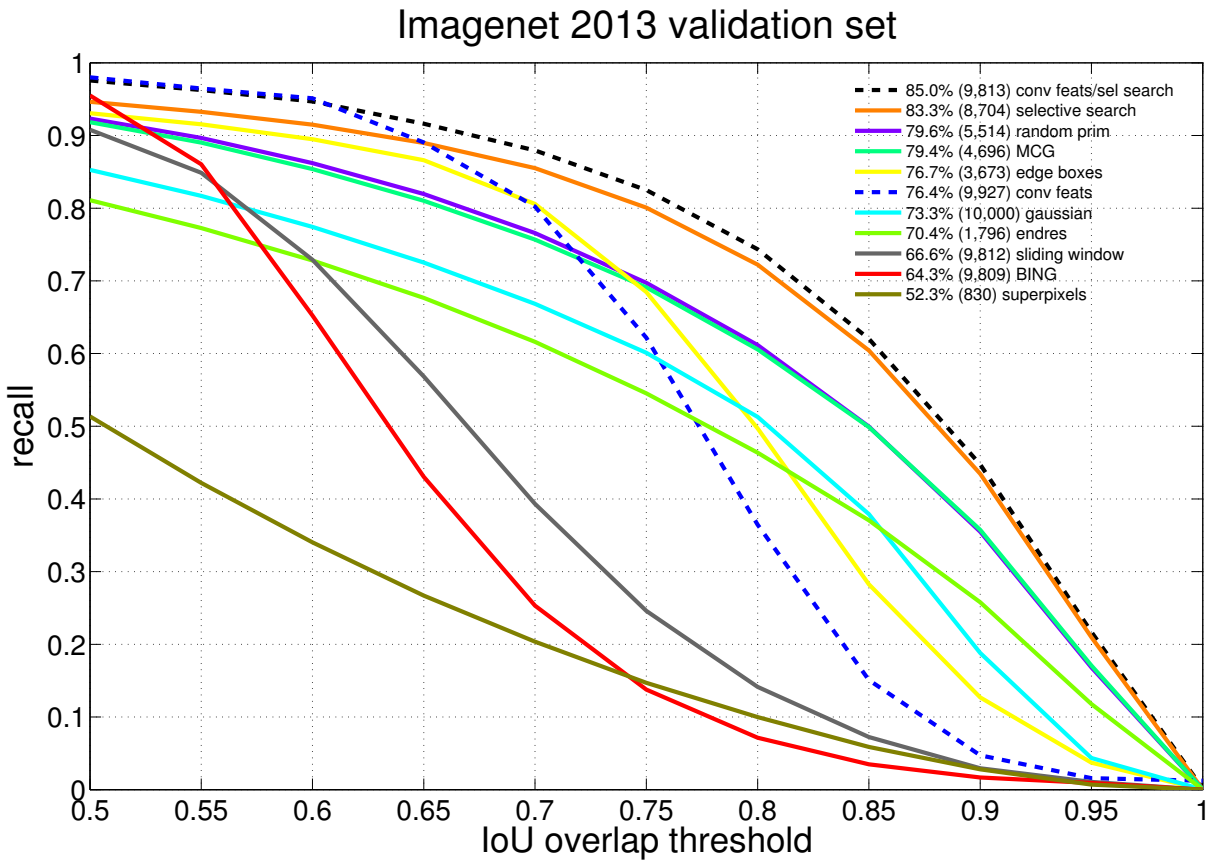


Figure 3.3: Proposals quality on ImageNet 2013 validation set when at most 10,000 regions are proposed per image. On recall versus IoU threshold curves, the number indicates area under the curve (AUC), and the number in parenthesis is the obtained average number of proposals per image. Statistics of comparison methods come from [73]. Our curves are drawn dashed.

Recall vs. localization: Our method belongs to the family of algorithms with fast and approximate object detection, such as BING and EdgeBoxes. These algorithms provide higher recall rate but poorer localization compared to methods that use segmentation, such as Selective Search. The latter are considerably slower but more accurate in localizing the objects. In Table 3.1 we provide the recall rate for varying localization accuracy measured by IoU. Our method provides the highest recall until around 65% IoU overlap. We also provide in Fig. 3.3 and Table 3.1 the gain in performance when we jointly use Selective Search and our method while still constraining the number of proposals to be less than $10,000$. The two approaches are complementary, as Selective Search provides better localization, while our algorithm sports higher recall, *i.e.,* higher retrieval rate of

Recall (%) for various IoU thresholds	$IoU \geq 0.5$	$IoU \geq 0.65$	$IoU \geq 0.8$	Testing time (s)
Selective Search [164]	94.6	89.0	**72.2**	10
Randomized Prim [111]	92.3	82.0	61.2	1
MCG [10]	91.8	81.0	60.6	30
Edge Boxes [192]	93.1	86.6	49.7	0.3
Boosting Convolutional Features	**98.1**	**89.4**	38.7	2
Endres 2010 [46]	81.1	67.7	46.4	100
BING [29]	95.5	43.0	7.2	0.2
Boosting Conv Features and Selective Search	**97.7**	**91.9**	**75.3**	12
Gaussian	85.3	72.5	51.3	0
Sliding window	90.8	56.9	14.1	0
Superpixels	51.3	26.7	10.0	1

Table 3.1: Comparison of our method against various category-independent object detectors on the Validation set of ImageNet 2013 (detection). We compare recall for various overlap thresholds. To be consistent with published literature, Pascal VOC's intersection-over-union (IoU) criterion is used. Methods are sorted according to the AUC, similar to Fig. 3.3. In bold font the top-2 methods per IoU threshold. Representative testing times are shown in the last column.

ground truth objects for localization accuracy less than 65% IoU.

Time complexity: Test time is linear in the number of deployed classifiers, scales, aspect ratios, and image size, with other parameters held constant. In Table 3.1 an estimate of average test time is shown in the last column for our framework compared to others as evaluated by [73]. Extracting convolutional responses for the validation image set of ImageNet 2013 takes only a few minutes with Caffe [79] on a single $K40$ GPU (in specific $2ms$ per image, mostly consumed for saving the features). Training the boosting framework [9], now included in Dollar's Matlab toolbox [37], on Pascal VOC 2007 (train-val and test: $9,963$ images with $24,640$ annotated objects) with a high-end multi-core CPU takes about three hours. This consists of training on all positives and $20k$ negatives, and additionally three rounds of bootstrapping, when at each round $20k$ more negatives are extracted among the classifier's false positives. The training time increases for larger

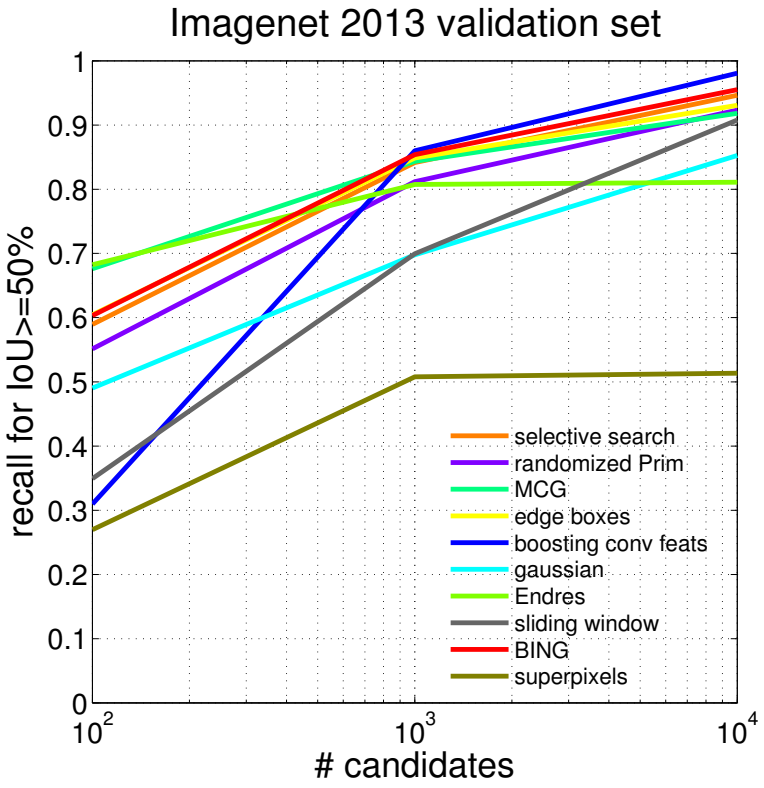


Figure 3.4: Proposals quality on ImageNet 2013 validation set in terms of detected objects with at least 50% IoU for various average number of candidates per image. Compared to all other methods from [73], our method is the most effective in terms of ground truth object retrieval when at least $1,000$ regions are proposed and accurate localization is not a major concern.

values of baseline detector's size, such as $d = 40$. But this still does not affect test time.

At test time, the classifier is applied densely on a sliding window on $20,121$ validation images from ImageNet (detection). $S = 12$ different scales and $R = 3$ different aspect ratios are used. Greedy non-maximal suppression is performed, where bounding boxes are processed in order of decreasing confidence, and once a box is suppressed it can no longer suppress other boxes. Separate and joint NMS are deployed with $U = 63\%$ and $V = 90\%$ IoU thresholds, correspondingly. Testing on a multi-core CPU takes about $2s$ per image.

In Fig. 3.4 we compare performance for at least 50% IoU recall for different number of proposed regions. Our scheme is the most effective when at least $1,000$ regions are proposed. For a smaller number of proposals, our performance degrades significantly.

3.4 ImageNet detection challenge

To investigate the end-to-end effectiveness of our proposal scheme, we evaluate the performance on ImageNet 2013 Detection. We fit our algorithm within the "Regions-with-CNN" detection framework [60] by replacing the output of Selective Search [164] with our regions instead.

In Table 3.2 we show the mean and median average precision on a subset of the validation set. We use the $\{val1, val2\}$ split, as in [60]. We deploy their pretrained CNN and SVM models as category CNN, which are trained on $\{val_1, train_{1k}\}$, *i.e.,* 9,887 validation images and 1,000 ground truth positives per class from the classification set. The Proposal CNN is the *VGGs* model from [26], pretrained on ILSVRC2012. Images are rescaled to 900 pixels width while preserving the aspect ratio. In that case, Selective Search proposes on average $5,826$ regions per image. It is wortwhile to mention that in [60] all images are rescaled to have 500 pixels width, which yields 29.7 and 29.2 mean and median AP for $2,403$ regions on average, respectively. For our method we used the model in Figs. 3.3 and 3.4, which generates $9,927$ proposals on average.

Average Precision (AP)	Mean AP	Median AP	Number of regions
Selective Search [164]	31.5	30.2	5,826
Boosting Convolutional Features	34.0	32.5	9,927

Table 3.2: Mean and median average precision on the ImageNet 2013 detection task. We employ the Regions-with-CNN (R-CNN) framework to compare regions by swapping Selective Search with our method. This comparison is without post-processing regression step.

3.5 Discussion

We have introduced a proposal scheme that leverages on convolutional features from lower layers of CNNs to train a boosting classifier to label each bounding box in a sliding window as "object" or "background". The former are then fed to a Category CNN for classification. In addition to comparing recall and computational cost against other leading proposal schemes, we have shown

improvement end-to-end on the ImageNet detection challenge, which can be ascribed to two factors: Higher recall within roughly $50 - 70\%$ IoU thresholds, and a larger number of proposals. Coarse localization is corrected to some extent from subsequent steps of R-CNN as the slack is absorbed by the CNN. Further improvement can be had with class-specific regression on top of prediction, so that bounding boxes better wrap the objects. Finally, an ensemble of models along with more sophisticated architectures (e.g., GoogLeNet [158], MSRA PReLU-nets [67], very-deep nets [146], etc.) would improve the entire pipeline. However, absolute performance is of no relevance here, as we use end-to-end scores as a means to evaluate the impact of our method in comparison with competing proposal schemes. Our method, when combined with Selective Search, is state-of-the-art when one is willing to use a number of proposals in the order of 1000 or more per image. While in some applications this may be too high a cost, the gain in performance may be worthwhile in other applications.

Our work is based on the premise that a CNN is not as effective in dealing with simple group transformations as its architecture would suggest, which is derived by the empirical success of Regions-with-CNN approaches in the current benchmarks in use in the community. Of course, empirical tests involve a large number of parameters and design choices that confound the comparison, so it is possible that improvements in the design of CNNs, for instance by allowing them to manage convolutions with respect to larger groups of transformations [32], would render the use of proposals moot. On the other hand, it is possible that the training cost of marginalizing known classes of transformations such as location, scale, aspect ratio, in terms of size of the data set, may be too high for current architectures, even for convolutional networks that are carefully designed to manage such variability. A more desirable course of academic action than empirical evaluation, with the ensuing escalating size of the datasets and number of parameters, would be to analyze the representational properties of convolutional architectures to determine the extent in which they can effectively marginalize nuisance variability *by design*, without the need to learn away nuisance variability that is known to exist and well understood. DSP-CNN is a step toward this direction, which is presented in Sect. A.3.4.

CHAPTER 4

Person Depth ReID: Robust Person Re-identification with Commodity Depth Sensors

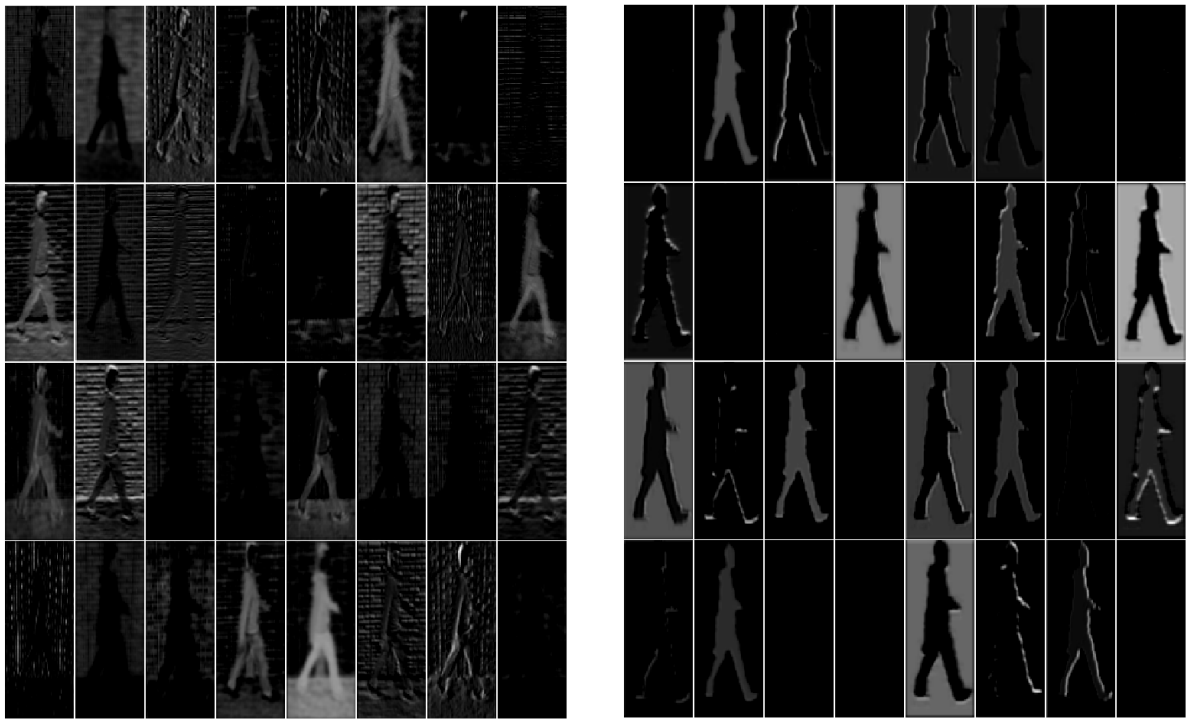
4.1 Introduction

Person re-identification is a fundamental problem in automated video surveillance and has attracted significant attention in recent years [57, 166, 62]. When a person is captured by cameras with non-overlapping views, or by the same camera but over many days, the objective is to recognize them across views among a large number of imposters. This is a difficult problem because of the visual ambiguity in a person's appearance due to large variations in illumination, human pose, camera settings and viewpoint. Additionally, re-identification systems have to be robust to partial occlusions and cluttered background. Multi-person association has wide applicability and utility in areas such as robotics, multimedia, forensics, autonomous driving and cashier-free shopping.

4.1.1 Related work

Existing methods of person re-identification typically focus on designing invariant and discriminant features [64, 49, 109, 95, 188, 176, 23, 102], which can enable identification despite nuisance factors such as scale, location, partial occlusion and changing lighting conditions. In an effort to improve their robustness, the current trend is to deploy higher-dimensional descriptors [102, 105] and deep convolutional architectures [101, 178, 1, 173, 169, 153].

In spite of the ongoing quest for effective representations, it is difficult to deal with very large variations such as ultra wide-baseline matching and dramatic changes in illumination and image resolution, especially when having limited training data. As such, there is vast literature in learning



Person Color ReID [173] Person Depth ReID

Figure 4.1: Convolutional filter responses from "conv3" layer using the same frame from the TUM GAID data as input for both Person Color ReID [173] and the feature encoder f_{CNN} of Person Depth ReID, which is drawn in Fig. 4.3.

discriminative distance metrics [93, 100, 191, 110, 118, 161, 102, 127, 114, 36] and discriminant subspaces [128, 105, 174, 102, 131, 185]. Other approaches alleviate the problem of pose variability by explicitly accounting for spatial constraints of the human body parts [27] or by predicting the pose within a multi-shot setting [30]. Similarly, adjacency constrained salient region matching [187] can help tackle the misalignment caused by large viewpoint and pose variation. In order to reduce the intra-class variance while preserving the intrinsic graphical structure, Shi et al. [142] mined positive and negative samples of different difficulty and built graphical relationships to approximate geodesic distance for training convolutional neural networks. Kodirov et al. [92] followed unsupervised methodology to formulate a graph regularized dictionary learning model and efficient optimization algorithm for cross-view matching.

However, a key challenge to tackle within both distance learning and deep learning pipelines is the *small sample size* problem [28, 185], which is attributed to the lack of large-scale person

re-identification benchmarks. New datasets have been released recently, such as CUHK03 [101] and MARS [189], a video extension of the Market-1501 dataset [190]. Their training sets are in the order of $20,000$ positive samples, i.e. two orders of magnitude smaller than Imagenet [136] which has been successfully used for object recognition [91, 146, 158].

The small sample size problem is especially acute on the person re-identification algorithms which leverage temporal sequences [66, 175, 22, 116], as the feature dimensionality increases linearly in the number of frames that are accumulated compared to the single-shot representations. On the other hand, explicitly modeling temporal dynamics and using multiple frames help algorithms to deal with noisy measurements, occlusions, adverse poses and lighting.

Adding regularization, such as *Dropout* [151], to the layers where most parameters are concentrated like the fully-connected ones, is one step towards reducing the parameter space and allow learning models to have higher generalization capability. Xiao et al. [173] achieved state-of-the-art accuracy on many person re-identification benchmarks by designing a deep convolutional network, similar in nature to *GoogleNet* [158], and training it on the union of several available datasets. Additionally, they further improve their performance on individual datasets by introducing "domain-guided dropout", where the dropout rate for each neuron is adaptively set as a function of its activation rate in the training set.

Haque et al. [66] introduced a carefully designed *glimpse* layer in order to compress their 4D spatiotemporal input representation of 500-frame video from $\approx 2.5 \times 10^9$ elements to a feature vector size in the order of 1×10^6 elements. They provide the compressed vector as input to a 4D convolutional encoder, while the decision for the next glimpse location is made using a sparsification technique with a reinforcement learning objective within a recurrent attention framework. However, designing such a downsampling mechanism with the objective of minimizing the large input size without losing much information can be challenging. Our algorithm has several key differences with this work: First, we do not use any glimpse layer and there is no locator module, as our input module detects the human silhouette region, which is used in its entirety. Second, instead of a $4D$ convolutional autoencoder, our encoder is 3-dimensional and its input is one frame. Third, we design a temporal attention unit to estimate the weight of each frame, which regularizes the recurrence and affects the multi-shot evaluation.

Some recent works in natural language processing [108, 24] explore temporal attention in order to keep track of long-range structural dependencies. Yao et al. [177] in video captioning use a soft attention gate inside their Long Short-term memory decoder, so that they estimate the relevance of current features in the input video given all the previously generated words. Our algorithm is different from these approaches as we use a *hard attention* unit, which is not differentiable but can be learned with reinforcement learning.

In the literature there are RGB-based approaches which extract the binary silhouettes and estimate geodesic distances between body parts [82, 183, 112]. Also, depth-based methods that use measurements from 3D human skeleton data have emerged in order to infer anthropometric and human gait criteria [125, 121, 2, 5, 45]. In an effort to leverage the full power of depth data, recent methods use 3D point clouds to estimate motion trajectories and the length of specific body parts [76, 186]. It is worthwhile to point out that skeleton information is not always available. For example, the skeleton tracking in Kinect SDK can be ineffective when a person is in side view or the legs are not visible.

4.1.2 Motivation

On top of the above-mentioned challenges, RGB-based methods are challenged in scenarios with significant lighting changes and when the individuals change clothes. These factors can have a major influence on the effectiveness of a system that, for instance, is meant to track people across different areas of a building over several days where different areas of a building may have very different lighting conditions, the cameras may have different color balance, and a person may wear clothes of different colors. This is our *key motivation* for investigating representations that are insensitive to color information such as silhouettes from depth.

4.1.3 Contributions

Our contributions can be summarized as follows:

i) We explore the use of depth sensors for person re-identification under adverse conditions, such as cases where the subjects appear with different clothes over time, while still being ro-

bust to viewpoint variation, human pose and partial occlusion. We construct representations from depth, so that we enable feature learning in end-to-end fashion with deep convolutional neural networks. The learned representations are different in nature from those learned with RGB models (see Fig. 4.1). Our experiments (e.g., see Fig. 4.5) suggest that depth is an effective modality for this task.

ii) We tackle the small sample size problem in various ways: first, we customize the optimization for the depth modality and deploy dropout in the convolutional encoder and the recurrent element. Second, we use the time as regularizer, as the agent is a recurrent neural network. Third, we design a temporal hard attention unit, whose weights are learned with a reinforcement learning objective, and enables scalability over longer sequences. Fourth, initializing the encoder with a pre-trained RGB ReID model [173] provides multimodal data augmentation.

iii) We conduct an empirical study using three re-identification datasets. The TUM-GAID database [71] is the largest one, including 305 persons. In the scenario where 32 subjects appear with different clothes after three months, our model achieves 6.2% higher top-1 and 23.6% higher top-5 accuracy compared to the top-performing RGB-based ReID method [173]. Next, we show further performance improvements when deploying recurrence and temporal attention, along with the head color information. We use the DPI-T dataset [66] with views from top to compare with Haque et al., who released this dataset and used an attention model with spatial glimpse layer. Finally, in order to evaluate the effectiveness of our algorithm with more challenging partial occlusions, viewpoints, and human poses, we introduce the FaceBody dataset. It involves 57 subjects that walk and operate in a realistic meeting room scenario.

4.2 Our Method

4.2.1 Input Representation

The input for our system is raw depth measurements from the Kinect V2 Sensor. The input data are depth images $\mathbf{D} \in \mathbb{Z}^{512\times424}$, where each pixel $D[i,j], i \in [1,\ldots,512], j \in [1,\ldots,424]$, contains the Cartesian distance, in millimeters, from the image plane to the nearest object at the particular (i,j) coordinate. In "default range" setting, $[0, 0.4m)$ and $(8.0m, \infty)$ ranges are classified as

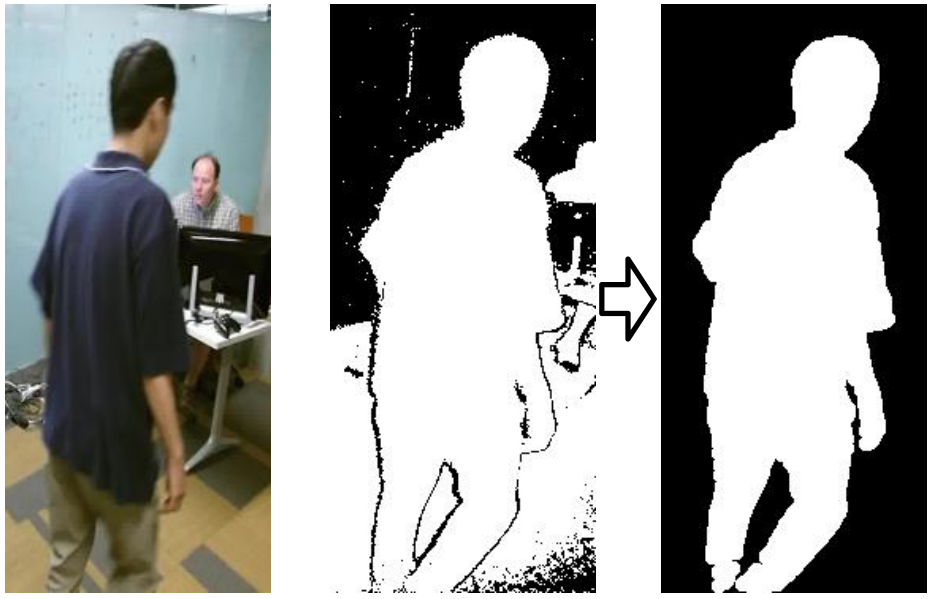


Figure 4.2: The cropped color image (left), the grayscale depth representation $\mathbf{D}_\mathbf{p}^\mathbf{g}$ (center) and the result after background subtraction (right) using the body index information $\mathbf{B}_\mathbf{p}$ from skeleton tracking.

unknown measurements, $[0.4, 0.8)[m]$ as "too near", $(4.0, 8.0][m]$ as "too far" and $[0.8, 4.0][m]$ as "normal" values. We have a dedicated algorithm to crop a rectangle that surrounds the person. When skeleton tracking is effective, the *body index* $\mathbf{B} \in \mathbb{Z}^{512\times424}$ is provided by the Kinect SDK, where 0 corresponds to background and a positive integer i for each pixel belonging to the person i. Therefore, when the Body Index is available, there is no need to use tracking in order to effectively crop the person (see Sec. 4.3.6).

After extracting the person region $\mathbf{D}_\mathbf{p} \subset \mathbf{D}$, the measurements within the "normal" region are normalized in the range $[1, 256]$, while the values from "too far" and "unknown" range are set as 256, and values within the "too near" range as 1. In practice, in order to avoid a concentration of the values near 256, whereas other values, say on the floor in front of the subject, span the remaining range, we introduce an offset $t_o = 56$ and normalize in $[1, 256 - t_o]$. This results in the "grayscale" representation $\mathbf{D}_\mathbf{p}^\mathbf{g}$. When the skeleton information is available, the body index $\mathbf{B}_\mathbf{p} \subset \mathbf{B}$ is used as binary mask for background subtraction on the person depth region $\mathbf{D}_\mathbf{p}$ before range normalization (see Fig. 4.2). Assuming that we crop person i, each pixel of $\mathbf{B}_\mathbf{p}$ with body index value different from i is set to 256.

We also consider the binary representation $\mathbf{D_p^b}$, as "black-and-white silhouette", by thresholding on $t_b = 128$:

$$D_p^b(i,j) = \begin{cases} 1, & \text{if } D_p^g(i,j) < t_b \\ 128, & \text{if } D_p^g(i,j) >= t_b \end{cases} \tag{4.1}$$

for $(i,j) \in [1,512] \times [1,424]$. The average image is computed over the training set and is subtracted from each testing image.

4.2.2 Model

The problem is formulated as *sequential decision process* of an agent that performs human recognition from a partially observed environment via video sequences. At each time step, the agent observes the environment via depth camera, calculates a feature vector based on a deep Convolutional Neural Network (CNN) and actively infers the importance of the current frame for the re-identification task via a temporal attention unit. The weight that is estimated by the attention unit determines whether the hidden representation is updated or not, which subsequently affects the classification. This hidden representation is computed by a recurrent module, which is meant to model the temporal dynamics. Finally, the agent receives a reward based on the success or failure of its action at each step. The agent's objective is to maximize the sum of rewards over time. The agent, as well as its comprising modules, are described in the following paragraphs. An outline of the model is shown in Fig. 4.3.

4.2.2.1 Agent

Formally, the problem setup is a Partially Observable Decision Process (POMDP). The true state of the environment is unknown. The agent learns a stochastic policy $\pi((w_t, c_t)|s_{1:t}; \theta)$ with parameters $\theta = \{\theta_g, \theta_w, \theta_h, \theta_c\}$ that, at each step t, maps the history of past information $s_{1:t} = I_1, w_1, c_1, \ldots, I_{t-1}, w_{t-1}, c_{t-1}, I_t$ to a distribution of actions. Both actions contribute to the recognition task via the estimated frame weight w_t and class posterior c_t. The weight w_t is computed by the temporal attention unit, which takes the current frame encoding g_t as input, while the classifier is attached on the RNN output h_t. The vector h_t maintains an internal state of the environment as

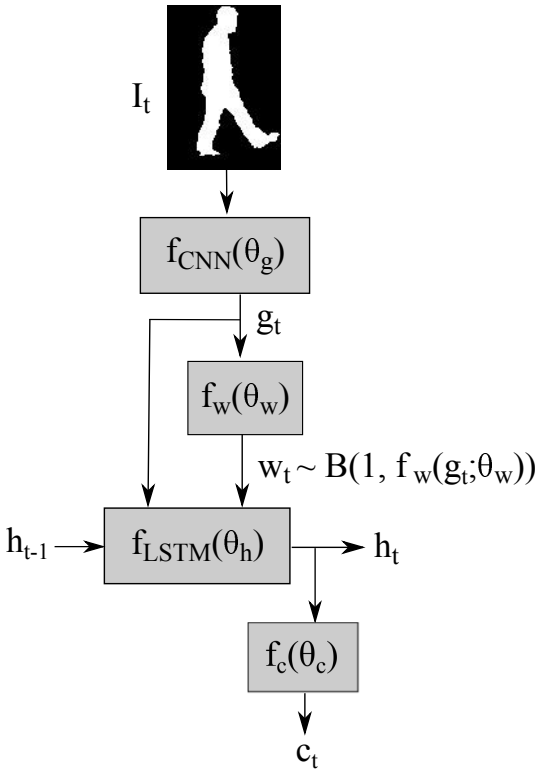


Figure 4.3: Model architecture: a recurrent deep neural network with temporal attention.

a summary of past observations and is updated by the recurrent module $f_{LSTM}(\theta_h)$. Note that, for simplicity of notation, the input image at time t is denoted as I_t, but the exact input representation is the grayscale region D_p^g as described in Sec. 4.2.1. At each time step t, the agent receives a reward r_t, which equals to 1 when the frame is correctly classified and 0 otherwise.

4.2.2.2 Feature encoder $f_{CNN}(\theta_g)$

The first design choice pertains to choosing features that are robust to various image and human shape variations due to camera viewpoint, human pose, light conditions, noisy measurement and partial occlusion. Recent investigation for the best architecture for person re-identification [101, 178, 1, 173, 169, 153] has shown that the deep convolutional network introduced by Xiao et al. [173] has outperformed other approaches on several public datasets. This network uses batch normalization [77] and includes 3×3 convolutional layers [146], followed by 6 Inception modules [158], and 2 fully connected layers. We adopt this architecture because in addition to its

effectiveness in RGB-based person re-identification, it allows us to initialize the parameters of Depth ReID with a pre-trained model, as a form of data augmentation.

We introduce two modifications in this network. We replace the top layer with a $256 \times N$ fully connected layer, where N is the number of subjects and depends on the dataset. The weights of this layer are initialized at random from a zero-mean Gaussian distribution with standard deviation 0.01. Additionally, we add dropout regularization between the fully-connected layers.

The model is trained to recognize the *identity* of a person by minimizing its cross-entropy loss, as is customary in other large-scale recognition tasks, such as face identification [154]. Afterwards, we remove the model's top layer and use the 256×1 vector as our feature encoding g_t.

4.2.2.3 Recurrent module $f_{LSTM}(\theta_h)$

We use Long Short-Term Memory (LSTM) element units as described in [180], which have been shown by Donahue et al. [39] to be effective in dealing with the vanishing and exploding gradients problem and modeling long-term dynamics for computer vision tasks. Assuming that $\sigma()$ is sigmoid, $g[t]$ is the input at time frame t, $h[t-1]$ is the previous output of the module and $c[t-1]$ is the previous cell, the implementation corresponds to the following updates:

$$i[t] = \sigma(W_{gi}g[t] + W_{hi}h[t-1] + b_i) \tag{4.2}$$

$$f[t] = \sigma(W_{gf}g[t] + W_{hf}h[t-1] + b_f) \tag{4.3}$$

$$z[t] = tanh(W_{gc}g[t] + W_{hc}h[t-1] + b_c) \tag{4.4}$$

$$c[t] = f[t] \odot c[t-1] + i[t] \odot z[t] \tag{4.5}$$

$$o[t] = \sigma(W_{go}g[t] + W_{ho}h[t-1] + b_o) \tag{4.6}$$

$$h[t] = o[t] \odot tanh(c[t]) \tag{4.7}$$

where W_{sq} is the weight matrix from source s to target q for each gate q, b_q are the biases leading into q, $i[t]$ is the input gate, $f[t]$ is the forget gate, $z[t]$ is the input to the cell, $c[t]$ is the cell, $o[t]$ is the output gate, and $h[t]$ is the output of this module. Finally, $x \odot y$ denotes the element-wise product of vectors x and y. Note that this LSTM does not use peephole connections between cell and gates.

4.2.2.4 Temporal attention unit $f_w(\theta_w)$

At each time step t the attention unit calculates the weight w_t of the image frame I_t, as the latter is represented by the feature encoding g_t. This module consists of a linear layer which maps the 256×1 vector g_t to one scalar, followed by Sigmoid non-linearity which squashes real-valued inputs to a $[0, 1]$ range. Next, the output of the module is defined by a Bernoulli random variable with probability mass function:

$$f(w_t; f_w(g_t; \theta_w)) = \begin{cases} f_w(g_t; \theta_w), & \text{if } w_t = 1 \\ 1 - f_w(g_t; \theta_w), & \text{if } w_t = 0 \end{cases} \tag{4.8}$$

During training, the weight w_t is chosen *stochastically* to be a binary value in $\{0, 1\}$. When $w_t = 1$, the current input g_t is forwarded through the LSTM. In case $w_t = 0$, the recurrent module is bypassed and the hidden representation from the previous frame is propagated to the current frame ($h_t := h_{t-1}$). During testing, the temporal unit acts deterministically and therefore $w_t = f_w(g_t; \theta_w)$.

This stochastic procedure introduces noise at the frame level during training, which is analogous to dropout regularization, but with a data-driven Bernoulli parameter instead. The probability of dropping a frame is controlled by the parameter $p = f_w(g_t; \theta_w)$, which ensures learning better models, as shown empirically in Sec. 4.3.7. Frames that the encoder is more confident to classify correctly are less likely to be dropped, as opposed to frames with a low-confidence encoder. This behavior is learned via reinforcement learning as explained in Sec. 4.2.3.2. An example sequence with the inferred Bernoulli parameter p for each frame is shown in Fig. 4.6.

4.2.2.5 Classifier $f_c(\theta_c)$

The classifier consists of a fully connected layer and Softmax, which map the 256×1 hidden vector h_t to the posterior class vector c_t with length N depending on the dataset. We use dropout regularization between the hidden vector and the classifier.

4.2.3 Training

In our experiments we pre-train the parameters of the feature encoder θ_g before attaching the RNN and the attention module and train the whole model in end-to-end-fashion. However, the entire architecture can be trained from scratch end to end. In the following subsections we describe the training process for the encoder and the recursive model with attention using a hybrid supervised loss.

4.2.3.1 Training the encoder $f_{CNN}(\theta_g)$

Deploying popular training techniques [18] with depth data needs careful consideration regarding the optimization process, as the dataset size is typically limited and the representations are of different nature than those that are color-based (see Fig. 4.2). We found empirically that stochastic gradient descent with modest base learning rate and low momentum can consistently converge to a good local minimum.

Optimization. Formally, stochastic gradient descent updates the model's weights w using a linear combination of the negative gradient of the loss $Q(z, w)$ for input z with respect to the weights w and the previous weight update v. The learning rate γ and the momentum μ are the coefficients of these two terms, respectively. At time t the update is:

$$v_{t+1} = \mu v_t - \gamma_t \nabla_w Q(z_t, w_t) \tag{4.9}$$

$$w_{t+1} = w_t + v_{t+1} \tag{4.10}$$

We choose base learning rate as low as $\gamma_0 = 3 \times 10^{-4}$ and momentum 0.5 in order to achieve convergence. The learning rate is reduced by a factor of 10 throughout training every time the loss reaches a "plateau". More details regarding the learning policy for each experiment are provided in Sec. 4.3.

Initialization. Initially, the network weights and bias were randomly initialized using the "Xavier" algorithm [61] which automatically determines the variance of initialization for each layer based

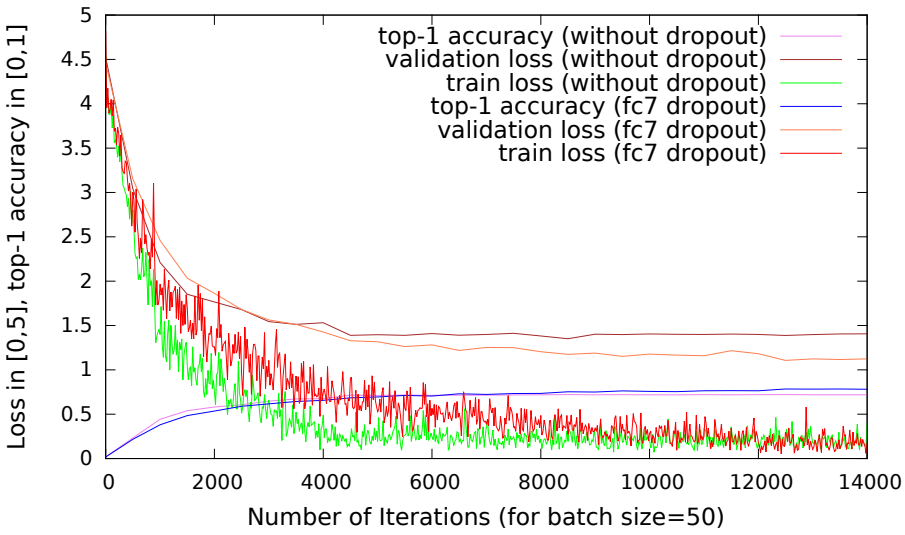


Figure 4.4: The encoder convergence on FaceBody data.

on the number of input and output neurons. Since learning the parameters of such a large model demands a significant amount of data, we found that multimodal data augmentation can significantly improve the performance. To this end, we initialized parameters θ_g with a pre-trained RGB-based person re-identification model that has been trained on the union of several ReID datasets (*JSTL-DGD* model from [173]). In that case, only the parameters of the added fully-connected layer for training the encoder are initialized at random. The learning rate multipliers for the learnable parameters of that layer are set 10 times larger than all multipliers for the rest of the network.

Regularization Given the data sparsity, regularizing the model weights is very important for identifying discriminative regions in depth images for person re-identification. We explore two different methods. First, we use the original model without the regularizer. Next, we introduce dropout between the two fully-connected layers ("fc7 dropout"), where most parameters are concentrated. In Fig. 4.4, we show the benefits of adding noise between layers "fc7" and "fc8", both in terms of top-1 accuracy and generalization ability.

4.2.3.2 Training the attention unit and the RNN

The parameters of our model $\theta = \{\theta_g, \theta_w, \theta_h, \theta_c\}$ are learned so that the agent maximizes its total reward over time $R = \sum_{t=1}^{T} r_t$ under the distribution of all possible sequences $p(s_{1:T}; \theta)$. This involves calculating the expectation $J(\theta) = \mathbb{E}_{p(s_{1:T};\theta)}[R]$ over a very big number of sequences, which can quickly become intractable. As proposed by Williams [172] and recently deployed successfully on recurrent models of spatial visual attention [120, 66], a sample approximation of the gradient, known as the REINFORCE rule, can be applied as follows:

$$\nabla_\theta J = \sum_{t=1}^{T} \mathbb{E}_{p(s_{1:T};\theta)}[\nabla_\theta \log \pi(u_t | s_{1:t}; \theta)(R_t - b_t)] \tag{4.11}$$

$$\approx \frac{1}{M} \sum_{i=1}^{M} \sum_{t=1}^{T} \nabla_\theta \log \pi(u_t^i | s_{1:t}^i; \theta)(R_t^i - b_t) \tag{4.12}$$

where s^i's sequences are obtained after M episodes.

In our case, as REINFORCE is applied on the output of Bernoulli stochastic unit with $p = f_w(g_t; \theta_w)$ and probability mass function $\log f(u; p) = u \log p + (1 - u) \log(1 - p)$, the gradient approximation is given by:

$$\nabla_\theta J \approx \frac{1}{M} \sum_{i=1}^{M} \sum_{t=1}^{T} \frac{u_t^i - p_t^i}{p_t^i(1 - p_t^i)}(R_t^i - b_t) \tag{4.13}$$

where $R_t^i = \sum_{t'=1}^{t} r_{t'}^i$ is the cumulative reward obtained following the execution of action u_t^i. Please note that this is a *biased* estimate of the gradient in order to achieve lower variance, as a baseline reward b_t is used. Consistent with [120], we set $b_t = \mathbb{E}_\pi[R_t]$, which is computed as the mean square error between R_t^i and b_t is minimized by backpropagation. This way, the baseline reward is learned at the same rate as the rest of the model.

All in all, a *hybrid* supervised loss is used to train the attention unit and the RNN's classification output. At each step, the agent takes an action w_t and the reward signal R_t^i is the supervision for evaluating the value of the action for the classification task. The REINFORCE rule increases the log-probability of an action that results in higher accumulated reward than the expected (baseline) total reward (i.e. by increasing $f_w(g_t; \theta_w)$). Otherwise, the log-probability decreases. Finally, in order to backpropagate the gradients through the classifier that is attached on the LSTM unit

backward through the whole network, we minimize the cross-entropy loss as is customary in supervised learning. The objective is to maximize the conditional probability of the true label given the observations I_t, i.e. we maximize $\log \pi(c_t^*|s_{1:t};\theta)$, where c_t^* corresponds to the ground-truth class for time step t.

4.3 Experiments

4.3.1 Depth-based Datasets

TUM-GAID Most existing depth-based datasets for person re-identification contain a small number of subjects. *IIT PAVIS* [13], *BIWI* [123] and *IAS-Lab* [124] contain 79, 50 and 11 persons, respectively. On the other hand, *TUM-GAID* database [71] is one of the largest to date. It contains RGB video, depth and audio for 305 people in three variations. A subset of 32 people is recorded a second time after three months with different clothes. Cropped versions for both the RGB and the depth image sequences are provided by the authors. The skeleton data is not available for this dataset.

FaceBody In spite of its large number of subjects, the persons in TUM-GAID always appear from the side view with fixed viewpoint and distance. As we want to explore the robustness of our method with varying vantage point, human pose, scale, and partial occlusions, we introduce a new dataset, which we name *FaceBody*. It includes 57 subjects appearing from 2 camera viewpoints walking into a meeting room in different walking patterns. Each person executes 6 walking

Dataset	IDs	Training Images	Testing Images	Appearances
TUM-GAID [71] (N-train)	150	34,881	17,625	1 (2 for 16 IDs)
TUM-GAID [71] (N-test)	155	35,454	17,776	1 (2 for 16 IDs)
FaceBody (t1, t4)	57	18,178	14,984	2
DPI-T [66]	12	3,740	4,010	25 (5 different clothes)

Table 4.1: Statistics of the datasets.

sequences, amounting to 12 sequences in total. We simultaneously collect the color and depth sequences with the Kinect V2 Sensor. For a subset of the data sequences where skeleton tracking is successful by Kinect SDK (e.g. when the face is visible at some point or when there are no large body occlusions), we also have the skeleton information, which is the 3D location of 25 human joints and pixel-wise *body index* per person.

Depth-based Person Identification from Top (DPI-T) Haque et al. [66] recently introduced DPI-T for person re-identification from depth. The new dataset contains 12 persons in 300 training and 355 testing sequences for a total of $3,740$ training and $4,010$ testing images, respectively. It is different from previous datasets in many ways. First, more diverse observations per individual are included, as the subjects appear in a total of 25 sequences across many days. The individuals wear 5 different set of clothes on average and walk at variable speeds. Second, unlike most publicly available datasets, the subjects appear from the top. This is a common scenario in automated video surveillance, where the camera is attached near the ceiling looking down. Third, the individuals are captured in daily life situations where they hold objects such as handbags, laptops and coffee. This data imposes new challenges in person re-identification and is used as the third benchmark. Table 4.1 provides a summary of statistics for the three datasets.

4.3.2 Evaluation Metrics

Top-k accuracy equals the percentage of test images or sequences for which the ground-truth label is contained within the first k model predictions. Plotting the top-k accuracy as a function of k gives the Cumulative Matching Curve (CMC). Integrating the area under the CMC curve and normalizing for the number of IDs produces the normalized Area Under the Curve (nAUC).

We evaluate our method in both "single-shot" and "multi-shot" mode by testing on individual images and sequences, respectively.

4.3.3 Experimental Settings

The encoder f_{CNN} is trained using Caffe [79]. Based on the input size of the deployed convolutional architecture, we rescale the input depth images to be 144×56 and subtract the mean depth image. We train our model using stochastic gradient descent with mini-batches of 50 images for training and testing. We set the momentum as low as 0.5, as higher values cause the model to diverge. The momentum μ effectively multiplies the size of the updates by a factor of $\frac{1}{1-\mu}$ after several iterations, so lower values result in smaller updates. The weight decay is set $2 * 10^{-4}$, as is common in Inception type of architecture [158].

The rest of the model in Fig. 4.3 is implemented in Torch/Lua [33]. We implemented our own customized conversion scripts from Caffe to Torch for the pretrained encoder, as the architecture is not standard. As for training Depth ReID, the batch size is 50 images, the momentum is 0.9 and the learning rate linearly decreases from 0.01 to 0.00001 in 400 epochs up to 500 epochs maximum duration. For the RNN history of $rho = 3$ frames is used, unless otherwise stated.

The experiments are conducted on a modern machine with NVIDIA Tesla $K80$ GPU, 24 Intel Xeon E5 cores and $64G$ RAM memory. The code implementing our method and the pretrained models necessary to reproduce the evaluation will be distributed publicly upon completion of the anonymous review process.

4.3.4 Baselines

Color model. The model designed by Xiao et al. [173] has been shown to outperform other methods on various public datasets. For instance, they achieve 13.2% higher CMC top-1 accuracy than the previous top-performing method [127] on large CUHK03 [101]. Therefore, we choose this method as our RGB-based baseline.

Motion model. We also compare our method to a motion-based method, as motion is also insensitive to appearance changes. Castro et al. [22] demonstrated competitive results on TUM-GAID, although they used a resolution of 80×60, which is 8 times lower than the original resolution of 640×480 for these sequences. By comparison, our model's input is 144×56. Additionally,

although we can make one-shot predictions, Castro et al. built a representation on subsequences of 25 frames. They extract dense optical flow between consecutive frames, crop and stack the flow channels, which are then passed through a convolutional neural network to obtain gait signatures for the entire subsequence.

Depth model. The Recurrent Attention Model (RAM) introduced by Haque et al. [66] relies only on depth images like our method. They introduced the DPI-T dataset, which we use for comparisons.

4.3.5 TUM-GAID database

Evaluation protocol. TUM-GAID depth data includes 12 "normal" sequences (N), 4 sequences with a backpack (B) and 4 sequences with coating shoes (S). We use the N setting, where sequences $n01$–$n06$ are from session 1, and sequences $n07$–$n12$ are from session 2, where the subjects have changed clothes. In half of the sequences the persons walk from left to right, while in the other half they walk from right to left. Of the 305 persons that appear in session 1, only 32 of them participate in session 2. Based on the official protocol, we use sequences $n1$–$n4$, $n07$–$n10$ for training, and sequences $n5$–$n6$ and $n11$–$n12$ for validation and testing, respectively. The subjects are partitioned into 150 training and 155 testing subjects, where the split is even for individuals participating in session 2.

Preprocessing. The tracked RGB and depth data are conveniently provided by the creators of TUM-GAID. Since the skeleton data are not available, we do not perform background removal. This has minor influence, as the background is identical for all sequences, filled in with a plain wall.

Task 1: Training on multiple clothes. First, we use all training sequences where the individuals appear in two sets of clothes. For this experiment we exclusively benchmark the f_{CNN} module. It is pre-trained on the training subjects, and afterwards fine-tuned on the training sequences of the testing subjects. Small base learning rate of 5×10^{-4} is used for pre-training. For fine-tuning the

Resolution	Method	top-1 accuracy (%)
640×480	*Gait Energy* [71]	44.0
	SVIM [171]	65.6
	Fisher Motion [23]	78.1
	SDL [183]	96.9
80×60	*Gait Signatures* [22]	62.5
144×56	*Depth ReID* (TL)	92.7
	Binary Depth ReID	95.4
	Depth ReID	**97.0**

Table 4.2: Comparisons on TUM-GAID for Task 1.

base rate is set 1×10^{-3}, as the network has adapted to depth data. A multistep policy is adopted where the learning rate decreases by a factor of 10 after $8k$ and $12k$ iterations and the training converges by $16k$ iterations. Since the Color ReID network [173] is already pre-trained on RGB-based datasets, we directly fine-tune it on the testing subjects. Finally, we train a depth model with the same protocol on binary representations $\mathbf{D}_{\mathbf{p}}^{\mathbf{b}}$ (see Sec. 4.2.1).

In Table 4.2 we provide comparison with other methods. Since the motion-based baseline [22], which also uses a deep convolutional architecture, allows fine-tuning only the top layer on the testing IDs, we also evaluate Depth ReID with this constraint. This method is presented at rows 6 and is notated as "TL". The deployed resolution that different methods use is noteworthy. Most methods under comparison use the data in their original resolution, which is 640×480. Our method and Castro et al. [22] that are based on convolutional networks downsample the images by a large factor in order to match the model input. Despite its lower resolution, Depth ReID outperforms the other methods. Additionally, even when fine-tuning only the last layer, the depth features are well-transferable [179] to the new set of persons.

Task 2: Constrained training on one set of clothes. Our objective is to examine whether a color-insensitive representation such as depth can offer more accurate person re-identification

Method	top-1	top-5	nAUC
RGB ReID, single-shot [173]	41.8	64.4	74.3
Depth ReID, single-shot	48.0	88.0	85.0
Depth, single-shot+RGB ReID	48.6	83.0	81.9
Head RGB ReID	59.2	78.4	79.4
Depth, single-shot+Head RGB ReID	65.4	85.9	85.2
Depth ReID, multi-shot with RNN	56.3	87.5	87.5
Depth ReID, multi-shot with RNN and attention	59.4	93.8	89.6
Head RGB ReID+Depth ReID, multi-shot with RNN and attention	71.9	93.8	89.9

Table 4.3: Recognition accuracy (%) and normalized area under the curve (%) on TUM-GAID (normal sequences) for Task 2.

when the subjects change clothes. To that end, we fine-tune on the training sequences $n01$–$n04$ of the testing IDs, using the sequences $n05$–$n06$ for validation. Therefore, this model has no access to training data from session 2. Next, the model is evaluated on sequences $n11$–$n12$. We make the assumption that the 32 subjects that participate in the second recording are known.

In Table 4.3 we show that Depth ReID is more robust than the corresponding RGB model, presenting 6.2% higher top-1 and 23.6% higher top-5 accuracy in single-shot mode. Note that Depth ReID achieves 97.0% accuracy (cf. Table 4.2) when sequences from both set of clothes are available during training. This is a critical problem to deal with as training data are not always available for new clothes.

As large variations in color and texture can be distracting for verification purposes, we attempt to rely more on the head region, which is less sensitive to day-by-day changes. To that purpose, we fine-tune the RGB-based pre-trained model [173] on the upper body part, which we call "Head RGB ReID". In order to remove the foreground, we extract a binary mask from depth by thresholding the depth representation. Given that the subjects in color and depth images are not perfectly pixel aligned, we apply morphological dilation on the binary mask with a circular disk of radius 4 to ensure that the foreground region includes the whole body in RGB. Then we crop the head

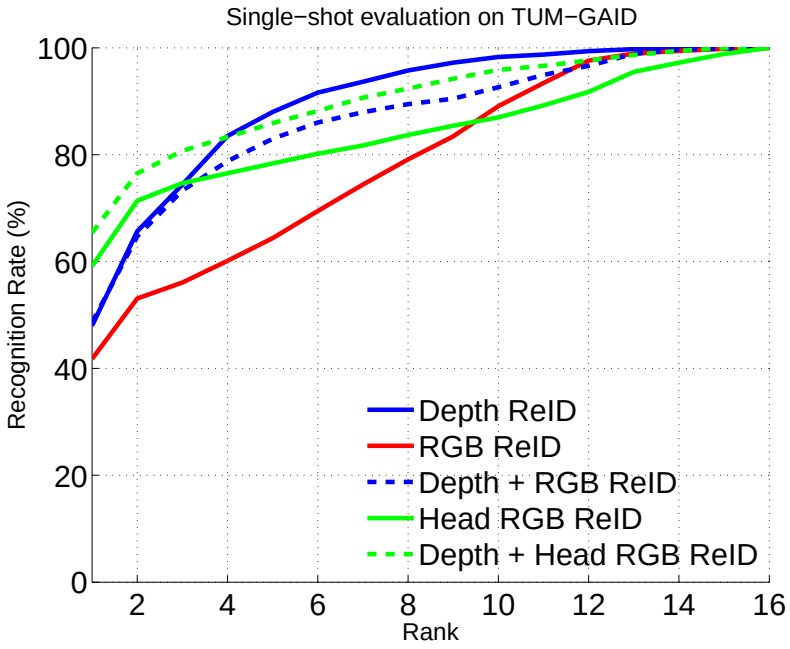


Figure 4.5: Cumulative matching curves for Task 2 on TUM-GAID. For rank-k (x axis), the y axis denotes recognition accuracy, if the ground truth label is within the method's top-k predictions.

region using the circumscribed rectangle around the top $1/4$ of the foreground region. In Table 4.3 we see the improvement in top-1 accuracy using Head RGB ReID, individually and jointly with depth information. Finally, we show the accuracy of Depth ReID with LSTM units and temporal attention, while evaluating each sequence in multi-shot mode.

In Fig. 4.5 we visualize the CMC curves for single-shot setting. Depth ReID scales better than its counterparts, which is validated by the normalized Area Under the Curve (nAUC) in Table 4.3. Intuitively, when the face is well-visible, Head RGB ReID is expected to be reliable, which explains the higher top-1 accuracy. On the other hand, when the face is mostly occluded, more guesses are not likely to improve the re-identification rate more quickly than body models.

4.3.6 FaceBody dataset

Evaluation protocol. The new dataset contains 6 sequences, $t1$–$t6$, of 57 subjects in a realistic meeting room scenario, as captured by two different viewpoints with Kinect V2 Sensors. The persons enter the room, walk in various paths, write on the board, and then exit the room. The

Method	top-1	top-5
RGB ReID, single-shot [173]	62.7	90.6
Depth ReID, single-shot	78.6	91.4
Depth ReID, Multi-shot with RNN	91.1	98.3
Depth ReID, Multi-shot with RNN and attention	92.9	98.8

Table 4.4: Re-identification accuracy (%) on FaceBody.

data, in addition to RGB and depth images, includes the skeleton tracking, i.e. the body index information, which is pixel aligned to the depth images and the $3D$ location of 25 pre-determined joints [144]. The body index is a reliable way to crop the persons in all frames, while sparing the need to deploy a tracker. However, the body index is available only when the skeleton tracking works successfully. In order to ensure the quality of extracted detections, we use the sequences $t1$ and $t4$ from each camera that have skeleton data for all 57 subjects. Let us denote the two cameras $c1$ and $c2$. The sequences $t1/c1$ and $t1/c2$ are used for training and the sequences $t4/c1$ and $t4/c2$ for testing, which sum up to $18,178$ training and $14,984$ validation images.

Preprocessing. We follow the process as described in Sec. 4.2.1 to obtain the depth crops $\mathbf{D_p^g}$. As there is no perfect alignment between the depth and the RGB data, we do not mask out the background for the RGB images. Therefore, the background is a nuisance for Color ReID on FaceBody. However, all sequences are recorded in the same room, so the background should have limited effect. Instead, for RGB images, we use the body index to extract a rectangular region around the person and add a 20–$pixel$ margin to ensure that the person's silhouette lies within the bounding box.

Comparisons (Table 4.4). Although FaceBody poses new challenges, as the subjects present pose variation and partial occlusions, Depth ReID is consistently more reliable than Color ReID. Part of this improvement can be attributed to the precise background subtraction based on body index in case of depth, which yields very accurate global shape information.

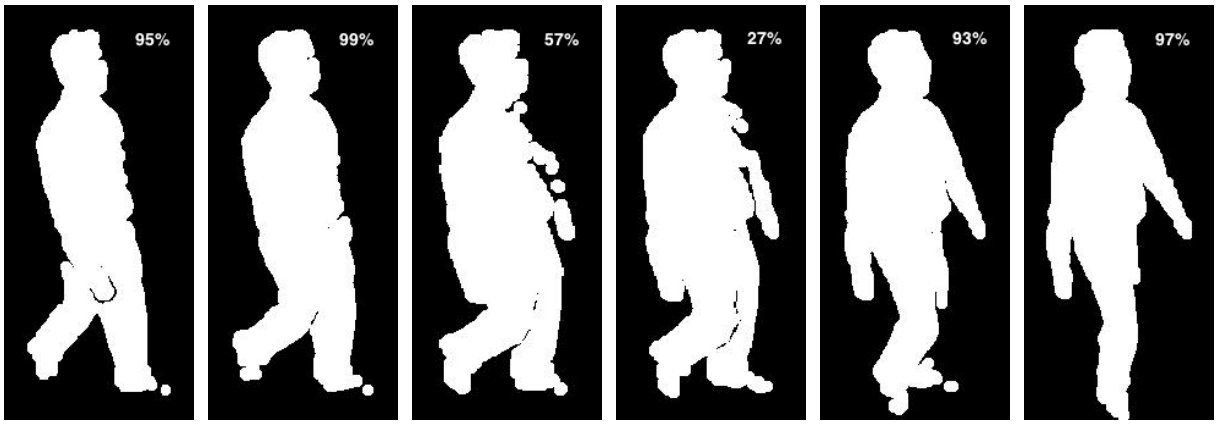


Figure 4.6: Example sequence along with the inferred Bernoulli parameter $p = f_w(g_t; \theta_w) \in [0, 100](\%)$ using a trained Depth ReID model with attention on FaceBody. Frames that are characterized by noisy measurements, uncommon pose and partial occlusions are likely to contribute less in multi-shot prediction, based on the estimated weight by the temporal attention unit.

Multi-shot evaluation. Following, we leverage on multiple frames from each sequence to make k-shot predictions. In order to make the evaluation more challenging, we allow to use only $k = 3$ consecutive frames per evaluation. This is a realistic scenario where a tracked person can be occasionally occluded or there is lack of motion. Most testing sequences have length N in the order of 200 frames. Our protocol is to perform 100 runs for each sequence where the start frame is chosen uniformly at random in $\{1, \ldots, N - k + 1\}$ range. In Table 4.4 we show the performance of Depth ReID with LSTM units and temporal attention.

Inspecting the temporal attention unit. After inspecting the estimated Bernoulli parameter $p = f_w(g_t; \theta_w)$ on unknown testing data, we observed that large variations are possible within one sequence, even between neighboring frames. Lower values are usually associated with noisy frames as in the example sequence in Fig. 4.6, or with challenging human pose and partial occlusions which are not well represented in the training set.

4.3.7 DPI-T dataset

Depth ReID is trained on DPI-T following the procedure described in Sec. 4.2.3 and the official evaluation protocol. For the multi-shot setting all available frames are used for a single sequence prediction. Although the persons on DPI-T have many more appearances (5 different sets of clothes on average), the sequences are shorter than in the other two datasets (approximately 16 frames per sequence).

In Table 4.5 we demonstrate our model's performance compared to Haque et al. [66]. For single-frame predictions we use only the encoder f_{CNN} with its attached classifier. For multi-shot mode with averaging in row 4, we simply calcuate the average of f_{CNN} outputs over each sequence frame. Next in rows $5-7$ we show results with LSTM units attached on the encoder. As for the last row, each sequence's class posterior is computed as the weighted sum of the model's outputs c_t for the sequence length K, based on the inferred weights $w_1, \ldots, w_K$. In rows 5 and 6 all frames contribute equally. Note that the RNN with constant Bernoulli $p = 0.5$ performs worse than the model which learns p. It is to be expected that learning the parameter p via the attention unit enforces learning better models, as frames are preserved or dropped out based on how likely they are to increase the accumulated reward and not uniformly at random such as when $p = 0.5$.

Setting	Method	top-1	top-5
Single-shot	*3D RAM* [66]	47.5	—
	Depth ReID, single-shot	62.3	93.6
Multi-shot	*4D RAM* [66]	55.6	—
	Depth ReID, averaging	72.6	96.4
	Depth ReID, RNN with Bernoulli p=0.5	73.9	96.4
	Depth ReID, RNN with learned Bernoulli p	75.9	96.0
	Depth ReID, RNN and attention	77.5	96.0

Table 4.5: Re-identification accuracy ($\%$) on DPI-T [66].

4.4 Discussion

We have presented a novel framework for person re-identification in the absence of RGB information, hence in the dark. Our pipeline leverages grayscale encodings from depth measurements, normalized, offset and masked using skeleton information and morphology, in order to learn depth representations with a recurrent deep convolutional architecture. We tackle the small sample size problem with regularizers and by introducing a temporal attention unit that allows efficient and scalable training with video sequences. The entire model can be trained end to end with a hybrid supervised loss under the principles of maximizing the conditional probability of the true class identity and the REINFORCE rule. Note that the model can be extended to calculate spatio-temporal attention regions, albeit not necessary in our pipeline as we use skeleton data to detect the region of interest, *i.e.,* the person.

CHAPTER 5

Learning to Discriminate in the Wild: Representation-Learning Network for Nuisance-Invariant Visual Comparison

5.1 Introduction

Representation-learning architectures have shown the ability to learn class-specific variability despite significant nuisance variability [56, 96, 134, 157, 162]. The problem of *nuisance variability* is particularly acute in Computer Vision, where even the same object or scene can yield a large variety of images depending on vantage point, illumination and partial occlusion, which can be nuisance factors for certain tasks [147]. This point has been recently emphasized by Poggio [130], who set forth the hypothesis that much of the ventral stream is tasked with managing the infinite amount of nuisance variability, and by Sundaramoorthi et al. [155], who showed that the intrinsic variability of objects in images is infinitesimal compared to nuisance variability. These theses would seem to challenge the possibility that nuisance variability in images can be *learned away* by even powerful learning architectures. In this manuscript, we put this challenge to the test by establishing two visual classification tasks, and deploying a fairly simple *representation-learning* architecture to tackle them.

The first task we select is the determination of *co-visibility*. This is a binary decision where, given two video frames, we wish to determine for each pixel whether or not back-projects onto the same point in physical space. This completely eliminates intrinsic variability, because the underlying scene is known to be the same and the diversity between images of the same scene is entirely attributable to nuisance factors such as different vantage points and illumination. We deploy a scheme based on a factored *Gated Restricted Boltzmann machine* [117] and joint *Superpixels* [122] of different scales to learn away such nuisance variability. The model is trained with

pairs of random images which are related by specific transformations. During testing violation of co-visibility occurs in regions of an image where corresponding patches between two frames are not recognized as sufficiently similar according to the model.

The second problem we deal with is segmentation in a single image, which is also cast as binary classification. Class variability makes nuisance elimination more challenging, but we show that a Gated RBM coupled with *Normalized Cuts* [143] are able to yield a semantic segmentation. The general framework is presented in Sec. 5.2, Sec. 5.3 demonstrates the experimental setting and comparative results on each problem and Sec. 5.4 consists of our conclusions. The upshot is that, even though in theory nuisances account for almost all the variability in the data [155], in practice the finite cardinality of data space acts as a regularizer, and since the classification occurs in data space, nuisance variability can be learned away.

5.1.1 Related work

The determination of co-visibility is related to the general problem of *correspondence*, that underlies a significant portion of Computer Vision research [140]. When correspondence is trivial, for instance when multiple images of the same scene are taken from a stationary camera at different time instants, this problem is known as *background subtraction* [129] and violations of co-visibility are due to the presence of moving foreground objects. In the more general setting, the determination of co-visibility is entangled with correspondence, so this problem relates to *optical flow*, another broad concern in the Computer Vision literature [7, 11, 12, 69, 152, 156]. Occlusion detection is often formulated as classification problem, where motion estimation is performed in a discrete setting ([94, 103]), which is a well-known difficult problem. Occlusion detection is closely related with occlusion boundary detection, where estimations are performed in video sequences [69, 78, 106, 152, 156] or single images [72, 137]. Martin et al. [113] fuse multiple cues from local image measurements to precisely infer the object boundaries in natural scenes. We compare with the occlusion regions learning work of Humayun et al. [74], who use various hand-crafted visual features, a subset of which is selected for each testing pair within a Random Forest-based framework. We also compare with the optical flow estimation of Ayvaci et al. [11].

5.1.2 Contributions

By choosing an appropriate training set, the network becomes insensitive to variability due to certain factors ("nuisances") like rotation and illumination changes, so the residual is informative for the rest factors, such as co-visibility in our occlusion detection setting or interclass variability for segmentation. We use a large training set which has been generated by applying certain transformations to random binary images. In our occlusion detection method there is no constraint regarding the order of the frames or the baseline range. There are no strict assumptions as for rigid motion or regarding the orientation of occlusion boundaries and the shape of occluded regions. However, discriminating between occlusions and disocclusions has a small post-processing overhead compared to flow algorithms. When taking the superpixel maps into account, our occlusion detection algorithm often outperforms recent methods based on optical flow and miscellaneous visual features. Although training may take hours depending on the size of the training set (~ 5 hours for $30,000$ image pairs with size 13×13 on a standard laptop), it is performed once and offline, and then the testing (e.g., 640×480 image pairs) takes only a few seconds. Finally, we propose applying our network on image segmentation and demonstrate how invariance and pairwise patch comparisons can yield a semantically meaningful segmentation.

5.2 Framework for Nuisance-Invariant Visual Comparison

Boltzmann machines are probabilistic bidirectionally connected networks that capture important information of an unknown distribution based on samples from this distribution. However, their learning is computationally consuming. *Restricted Boltzmann machines* impose the probabilistic restriction of statistical independence between variables of same layer given the state of variables of all other layers and simplifies the learning process. The 2-layer architecture can be modelled as bipartite undirected graph.

5.2.1 Gated Restricted Boltzmann machine

A *Gated Restricted Boltzmann machine* is a parametrized generative model representing a probability distribution. Given some observations (i.e., the training data), learning means adjusting the parameters so that the represented distribution fits the training data as well as possible. The Gated RBM consists of 3 layers of binary variables: two layers of visible units that correspond to the observations and one hidden layer, which encodes dependencies between two observable layers. Therefore, this model can capture the relationship (modulo a set of factors that it is trained to be invariant to) and in turn "similarity" between two images.

A Gated RBM consists of K hidden units $\mathbf{H} = (H_1, \ldots, H_K)$ that capture the dependencies between two layers of observed variables with units $\mathbf{X} = (X_1, \ldots, X_I)$ and $\mathbf{Y} = (Y_1, \ldots, Y_J)$. Adopting binary random variables, ($\mathbf{X}$, $\mathbf{Y}$, $\mathbf{H}$) takes values ($\mathbf{x}$, $\mathbf{y}$, $\mathbf{h}$) $\in \{0, 1\}^{I+J+K}$. The image transformations do not include arbitrary motions of individual pixels, so the three-way interactions among the layers can be modeled as the product of all possible two-way interactions with F factors [117]. Thus, the joint probability distribution is $p(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{x},\mathbf{y},\mathbf{h};\theta)}$ with energy

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}; \theta) = -\sum_{f=1}^{F}(\sum_{i=1}^{I} u_{if}x_i)(\sum_{j=1}^{J} v_{jf}y_j)(\sum_{k=1}^{K} w_{kf}h_k) - \sum_{i=1}^{I} a_i x_i - \sum_{j=1}^{J} b_j y_j - \sum_{k=1}^{K} c_k h_k, \quad (5.1)$$

where $\theta = \{\mathbf{U}, \mathbf{V}, \mathbf{W}, \mathbf{a}, \mathbf{b}, \mathbf{c}\}$ are the model parameters and $Z(\theta) = \sum_{\mathbf{x},\mathbf{y},\mathbf{h}} e^{-E(\mathbf{x},\mathbf{y},\mathbf{h};\theta)}$ is the *partition* function. Instead of $I \times J \times K$ interaction tensor, three matrices with sizes $I \times F$, $J \times F$ and $K \times F$ are factorized in a common product. Hence, the order of parameter complexity decreases from cubic to square. For all $i \in \{1, \ldots, I\}$, $j \in \{1, \ldots, J\}$, $k \in \{1, \ldots, K\}$ and for all $f \in \{1, \ldots, F\}$, u_{if}, v_{jf} and w_{kf} are real-valued weights associated with the f factor and i, j or k unit, correspondingly. Weight matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ consist of "filters" $\{\mathbf{u}_f, \mathbf{v}_f, \mathbf{w}_f, f = 1 \ldots F\}$. Additionally, a_i, b_j and c_k are real-valued bias terms associated with the ith and jth visible units and kth hidden unit, respectively. The model is illustrated in Fig. 5.1.

Intuitively the first energy term represents a similarity score, as its high value coincides with co-occurrence of high projection scores of images $\mathbf{x}$, $\mathbf{y}$ and some subset of hidden variables $\mathbf{h}$ on F factors. The filters' shape and the semantics of similarity inferred by the model depend on the training set. For example, after training with pairs of images which are related by affine transfor-

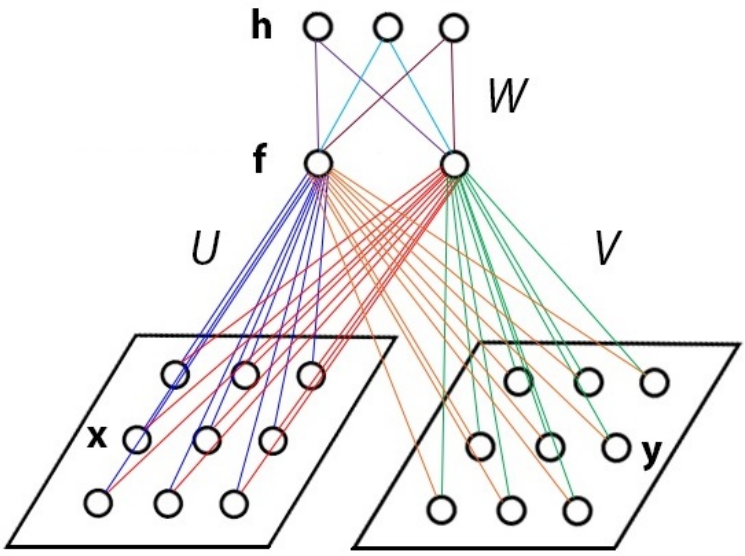


Figure 5.1: Graphical representation of a Gated Restricted Boltzmann machine (RBM).

mations, the hidden variables capture "elementary" dependencies between the observed variables like translational shifts, planar rotations and other small-dimensional (local) group transformations. In that case, two testing images will be considered as "similar" by the model when they are almost identical or parts of them are related by affine transformations (of magnitude similar with these ones appearing on the training data).

The *symmetric* model is a special case where the weights of both visible layers are equal, that is $\{u_{if} = v_{if}, i = 1 \dots I, I = J\}$. It operates a complex transformation that is determined by the hidden layer and maps a set of representations of one visible layer to the other. The representations are projections on a common space, which topologically "compensates" the transformations that appear on the training set. More generally, the non-symmetric model is essentially a mapping induced by the hidden layer between different, but related (according to the training set) representations of the observable layers. In Fig. 5.2, we show the observed layers' filters $\{\mathbf{u}_f, \mathbf{v}_f, f = 1 \dots F, F = 100\}$ when the model is trained exclusively with shifted and scaled image pairs, respectively. The non-symmetric Gated RBM can be applied on image pairs of different size ($I \neq J$), while the numbers of hidden variables K and factors F can be selected by the user. We mainly experimented with values: $I = J = 13 \times 13 = 169$ and $I = J = 26 \times 26 = 676$, $K = 50$–200, $F = 100$–200.

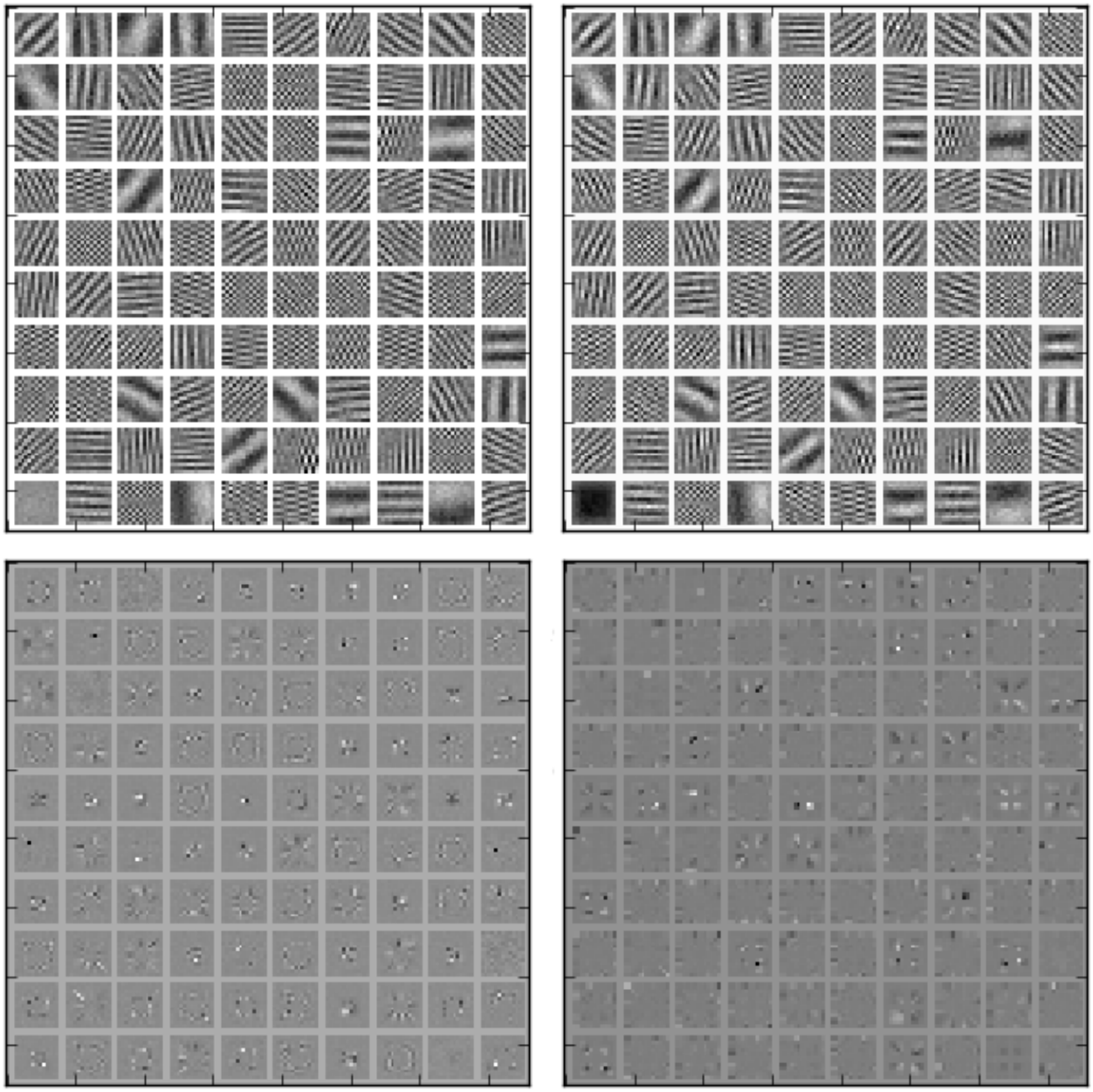


Figure 5.2: Filters generated when training exclusively with shifted (top) and scaled (bottom) random binary images.

5.2.2 Conditionals and Marginals

The complexity advantage of Restricted Boltzmann machines is that all variables of one layer are independent given the state of all other layers' variables. Thus, the joint conditional distribution is the product of all conditional distributions and calculations can be done in parallel. The conditional

distributions can be factorized as:

$$p(\mathbf{x}|\mathbf{y},\mathbf{h}) = \prod_{i=1}^{I} \mathcal{B}(x_i; \sigma[\sum_{f=1}^{F} u_{if}(\sum_{j=1}^{J} v_{jf} y_j)(\sum_{k=1}^{K} w_{kf} h_k) + a_i])$$
$$p(\mathbf{y}|\mathbf{x},\mathbf{h}) = \prod_{j=1}^{J} \mathcal{B}(y_j; \sigma[\sum_{f=1}^{F} v_{jf}(\sum_{i=1}^{I} u_{if} x_i)(\sum_{k=1}^{K} w_{kf} h_k) + b_j])$$
$$p(\mathbf{h}|\mathbf{x},\mathbf{y}) = \prod_{k=1}^{K} \mathcal{B}(h_k; \sigma[\sum_{f=1}^{F} w_{kf}(\sum_{i=1}^{I} u_{if} x_i)(\sum_{j=1}^{J} v_{jf} y_j) + c_k]) \tag{5.2}$$

where $\mathcal{B}(x; p)$ is the pdf of a Bernoulli random variable x with parameter which is function of the other two layers and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function.

The distribution over an image pair ($\mathbf{x}$, $\mathbf{y}$) is taken by marginalizing the joint distribution over $\mathbf{h}$:

$$p(\mathbf{x},\mathbf{y}) = \sum_{\mathbf{h}\in\{0,1\}^K} p(\mathbf{x},\mathbf{y},\mathbf{h}). \tag{5.3}$$

The number of possible $\mathbf{h}$ increases exponentially with the number K of hidden variables, making the computation intractable for reasonable values. However, approximating the unknown distribution with Gibbs sampling allows us to work only with the conditionals. This fact, along with the conditional independence among variables in each layer of Gated RBM given the other two layers, make computational cost reasonable. Additionally, a GPU-based implementation[1] of the model speeds up the training process by an order of magnitude.

5.2.3 Maximum Likelihood Learning

Given a set of *i.i.d.* training examples $D = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})\}$, the model parameters θ are learned via an unsupervised learning framework. The log-likelihood given observed training pairs D is maximized:

$$\max \ \log \mathcal{L}(\theta|\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{n=1}^{N} \log \ p(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}; \theta). \tag{5.4}$$

[1] Publicly available online from R. Memisevic and J. Susskind at *http://learning.cs.toronto.edu/ rfm/code/gbmcuda.py*.

Algorithm 2 Training with 3-way Contrastive Divergence

Input: Gated RBM $(\mathbf{X}, \mathbf{Y}, \mathbf{H})$, training batch D.

Output: Weights update $\Delta\theta$.

Initialize all weights $\Delta\theta = 0$.

for all $(\mathbf{x}, \mathbf{y}) \in D$ **do**

$(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}) \leftarrow (\mathbf{x}, \mathbf{y})$

for $t = 0, \ldots, k-1$ **do**

in random order:

$\forall k = 1, \ldots, K$ sample $h_k^{(t)} \sim p(h_k|\mathbf{x}^{(t)}, \mathbf{y}^{(t)})$

$\forall i = 1, \ldots, I$ sample $x_i^{(t+1)} \sim p(x_i|\mathbf{h}^{(t)}, \mathbf{y}^{(t)})$

$\forall j = 1, \ldots, J$ sample $y_j^{(t+1)} \sim p(y_j|\mathbf{h}^{(t)}, \mathbf{x}^{(t)})$

end for

for all weights **do**

$\Delta\theta = \Delta\theta - \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}^{(0)}, \mathbf{y}^{(0)}) \frac{\partial E(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}, \mathbf{h})}{\partial \theta} + \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \frac{\partial E(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}, \mathbf{h})}{\partial \theta}$

end for

end for

For a single training pair $(\mathbf{x}, \mathbf{y})$ the log-likelihood gradient w.r.t. a single model parameter θ is:

$$\begin{aligned}
\frac{\partial \log \mathcal{L}(\theta|\mathbf{x}, \mathbf{y})}{\partial \theta} &= \frac{\partial \log(\frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{x}, \mathbf{y}, \mathbf{h}; \theta)})}{\partial \theta} \\
&= \frac{\partial(\log \sum_{\mathbf{h}} e^{-E(\mathbf{x}, \mathbf{y}, \mathbf{h})} - \log \sum_{\mathbf{x}, \mathbf{y}, \mathbf{h}} e^{-E(\mathbf{x}, \mathbf{y}, \mathbf{h})})}{\partial \theta} \\
&= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}, \mathbf{y}) \frac{\partial E(\mathbf{x}, \mathbf{y}, \mathbf{h})}{\partial \theta} + \sum_{\mathbf{x}, \mathbf{y}, \mathbf{h}} p(\mathbf{x}, \mathbf{y}, \mathbf{h}) \frac{\partial E(\mathbf{x}, \mathbf{y}, \mathbf{h})}{\partial \theta}. \qquad (5.5)
\end{aligned}$$

By combining Eqs. 5.4 and 5.5, the mean of this derivative over the training set can be expressed as:

$$\frac{1}{N} \sum_{n=1}^{N} \frac{\partial \log \mathcal{L}(\theta|\mathbf{x}, \mathbf{y})}{\partial \theta} = \left\langle \frac{\partial E(\mathbf{x}, \mathbf{y}, \mathbf{h})}{\partial \theta} \right\rangle_{p(\mathbf{h}|\mathbf{x}, \mathbf{y}) q(\mathbf{x}, \mathbf{y})} - \left\langle \frac{\partial E(\mathbf{x}, \mathbf{y}, \mathbf{h})}{\partial \theta} \right\rangle_{p(\mathbf{x}, \mathbf{y}, \mathbf{h})}. \qquad (5.6)$$

The second term in Eq. 5.6 is intractable, as it is computed over all configurations $(\mathbf{x}, \mathbf{y}, \mathbf{h})$ (increases exponentially with number of units $I + J + K$). However, Gibbs sampling of the unknown

distribution gives a tractable approximation. Samples are drawn alternatingly from the conditional distributions $p(\mathbf{h}|\mathbf{x},\mathbf{y})$, $p(\mathbf{x}|\mathbf{h},\mathbf{y})$ and $p(\mathbf{y}|\mathbf{h},\mathbf{x})$ in random order and the sampling process is terminated in k steps ($k = 1$ works well in practice [70]). Given the tri-partite structure of the model, the learning process has been called as *3-way Contrastive Divergence* [157] and is summarized in Alg. 2.

5.2.4 Distance function

The model can be trained with pairs of images related by many transformations. This process makes it invariant over all these transformations, so an appropriate similarity score given by the model can potentially discriminate between similar/non-similar images modulo these factors. The log-likelihood that is assigned to a testing pair $(\mathbf{x},\mathbf{y})$ is:

$$log\ p(\mathbf{x},\mathbf{y}) = -logZ + \sum_{i=1}^{I} a_i x_i + \sum_{j=1}^{J} b_j y_j + \sum_{k=1}^{K} log(1 + e^{c_k + \sum_{f=1}^{F} w_{kf} (\sum_{i=1}^{I} u_{if} x_i)(\sum_{j=1}^{J} v_{jf} y_j)}). \quad (5.7)$$

The normalizing term $logZ$ is intractable, as it includes marginalization over $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$. Fortunately, when we compare pairs of images, this term is common and can be eliminated. However, to use the *unnormalized* likelihood as distance of two images would be problematic, as a single pair $(\mathbf{x},\mathbf{y})$ could be made to have arbitrarily small likelihood, e.g., by rescaling both images with some constant. To deal with that the following distance function is used:

$$d(\mathbf{x},\mathbf{y}) = -logp(\mathbf{x},\mathbf{y}) - logp(\mathbf{y},\mathbf{x}) + logp(\mathbf{x},\mathbf{x}) + logp(\mathbf{y},\mathbf{y}), \quad (5.8)$$

as was first proposed in [163] for a RBM and was also deployed in [157]. The normalizing terms are eliminated, and the likelihood of any single image is normalized for both observable layers. Strictly speaking, d is a *semi-metric*, as the triangle inequality is not guaranteed to hold among three testing images.

5.3 Experiments

After training with a large dataset of image pairs which are related by a specific set of transformations, the model is invariant with respect to them and the distribution of distances calculated by

Eq. 5.8 for a set of testing image pairs can be informative for determining other factors, such as *occlusions* and *interclass variability*.

5.3.1 Occlusion Detection

The model is trained using Alg. 2 with pairs of random images related by affine transformations, shifts and rotations, scale and illumination variation. The first factors intend to deal with different vantage points where these images are captured from, while the latter one with different lightning conditions. In our experiments the Gated RBM's observable layers have size either 13×13 or 26×26 and the range of pixel-wise transformations on the training set varies between $3 - 6$ pixels. In order to obtain the results of this section, $F = 200$ filters and $K = 100$ hidden variables were used. The model was trained over $10,000$ epochs, where the training set included $10,000$ purely shifted, $5,000$ purely rotated images, $5,000$ general affine transformations, $5,000$ illumination variant and $5,000$ scaled pairs. Affine transformations are applied to random, binary training images, which empirically proved to give equally effective model compared to when training with patches cropped from natural images. Applied transformations is all that counts instead of specific information of any single image/visible layer. Illumination-variant training images are extracted from PHOS dataset [168]. Batch size $D = 100$–$1,000$ and $5,000$–$10,000$ epochs were used in these experiments.

During testing, two frames were partitioned into $d \times d$ densely overlapping patches ($d = 13$ or $d = 26$) and Eq. 5.8 was used to estimate the "distance" of corresponding patches according to the model. After training over all these factors, which are nuisances in our setting, $d(\mathbf{x}, \mathbf{y})$ provides a score to quantify co-visibility in a testing pair $(\mathbf{x}, \mathbf{y})$ because occlusions are the main cause of disagreement. Thresholding the distance map yields the binary occlusion map. Comparisons are made at the patch level, but the resulting distance is applied only to the central pixel of each patch. Overlapping patches are deployed, while all comparisons can be performed in parallel. Our framework was tested on sequential video frames taken from Berkeley Motion Segmentation [19], Middlebury [12] and UCL Optical Flow [7] datasets.

A *baseline* algorithm can be built where simple differences of average intensities over $13 \times$

Figure 5.3: Occlusion detection between frames 7 and 8 of "Cars8" sequence of the Berkeley Motion Segmentation dataset. The occlusion (and disocclusion) areas are displayed on both frames. The image pair on top is obtained with the baseline algorithm, which gives many false alarms on turbulent and with variant lightning scene areas, such as the road and the car's front surface. The image pair below displays our detection having used the aggregate superpixel distance from Eq. 5.9 and $m = 8$ superpixel maps.

13 patches are extracted and thresholded. This procedure yields a large number of false alarms, because any movement or lightning change in the background or the occluder affects the "naive" patch distance. On the other hand, our network's invariance over all these transformations provides background/foreground subtraction and the residual is mostly occlusions. Fig. 5.3 demonstrates this concept.

Toward superpixels: Training the model with bigger visible layers offers invariance over larger transformations, but it typically gives less accurate predictions close to the boundaries of the oc-

clusion regions, as the model needs to examine a bigger patch and deal with more nuisances. It can drive occlusion detection though by providing a mask that offers subtraction of the larger nuisances and then testing with smaller visible layers refines the occluded regions. Moreover, a deeper architecture would not be especially helpful, because the primary purpose of this network is to perform image comparison modulo small deformations. It does not intend to simulate complicated transformations like facial expressions or body poses. However, empirical work suggests that the network *per se* can give decent, but not competitive results, mainly because of computational resources limitations. Training a very large Gated RBM with thousands of hidden variables and filters over all possible transformations in theory could give an oracle that could eliminate all possible nuisances and in turn discriminate occlusions with infinitesimal classification error. In practice, though, for a computationally tractable solution that yields competitive occlusion detection, we turn to *superpixels*.

Superpixels are basically regions of near-uniform intensity on the image domain and our conjecture is that with high probability their pixels back-project to points in the scene that belong to the same object. Therefore, it is natural to resolve for entire superpixels whether they are co-visible or not. Averaging the model's distances over the whole superpixel is a mechanism that is robust against outliers and gives accurate occlusion boundaries. However, superpixel partitions on any single image are not useful when working with image pairs. Therefore, we design a mechanism of *jointly* extracting superpixels in two images (one common superpixel partition) in order to have pixel groups that faithfully "follow" the boundaries on both frames and share common appearance/texture. The superpixel code [122] is based on the Boundary Detector from [113], and in order to extract joint superpixels we modified it adopting as edge probability map the maximum of the two images' probability maps and choosing as angle θ for every pixel (i, j) the corresponding angle $\theta_1(i, j)$ or $\theta_2(i, j)$ of the image with dominant gradient there. In order to be less dependent on the algorithm's randomness and process in different scales, the maximum number of pixels per superpixel, the number of eigenvectors and other parameters, m superpixel partitions for each testing image pair are extracted according to various values for the above-mentioned parameters and

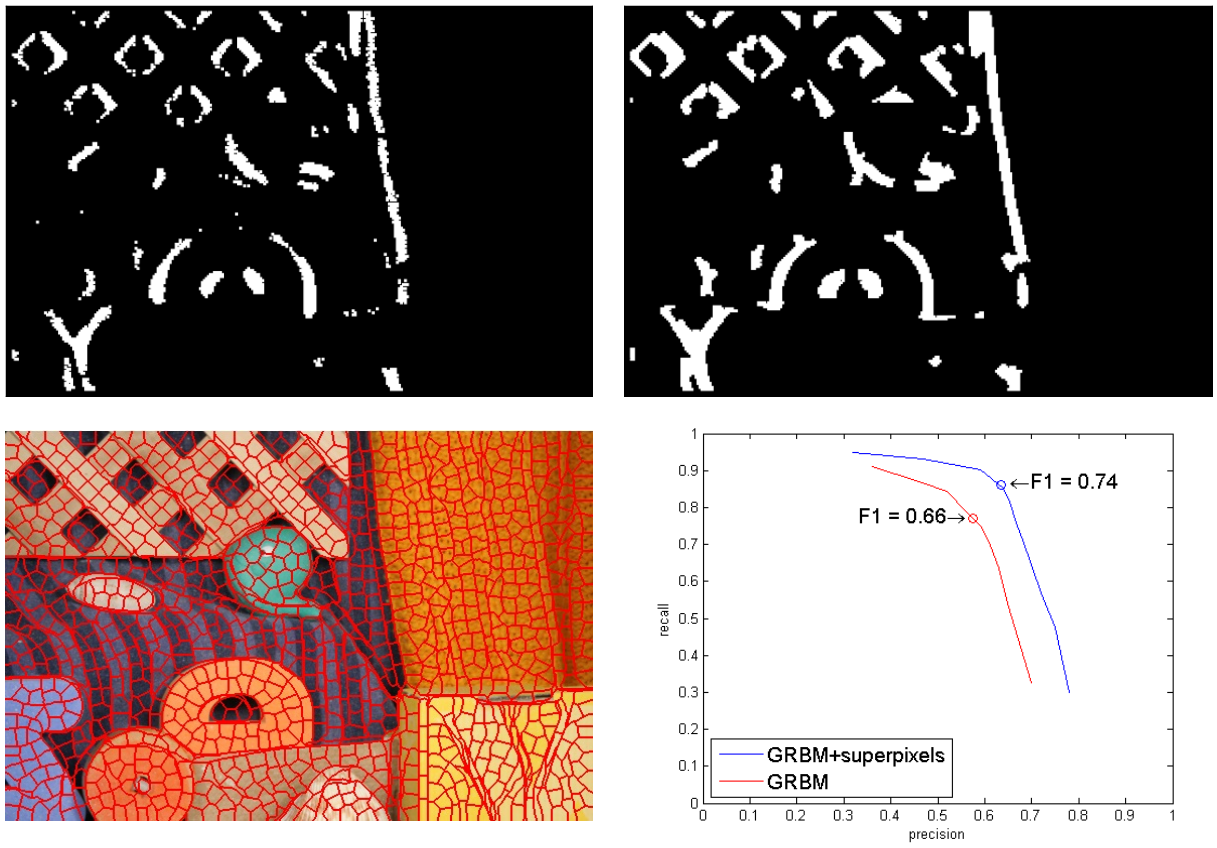


Figure 5.4: This figure demonstrates the influence of superpixel information in the occlusion detection task. It shows results from the "RubberWhale" sequence of Middlebury dataset (frames 7 and 14). At the first row we see the occlusion detection without and with superpixel information considered, respectively. The occlusion regions consist of fewer, more compact connected components, have fewer outliers, and fit better on the occluders' boundaries. At the second row, the first image displays the joint superpixel partition, while next the PR curves are illustrated.

averaged the distance scores over all. The *aggregate* superpixel score is defined as:

$$d_{aggregate}(i) = \frac{1}{m} \sum_{m\ maps} dist_m(i) \quad \forall i \in I, \tag{5.9}$$

where $dist_m(i)$ is the average score over the superpixel in map m that contains pixel i.

Figs. 5.4 shows via PR curves how occlusion detection improves when superpixels are deployed. Fig. 5.5 includes comparisons with [11] and baseline algorithms based on a one-layer RBM and a 2-layer perceptron, which are trained on concatenation of same training pairs that are

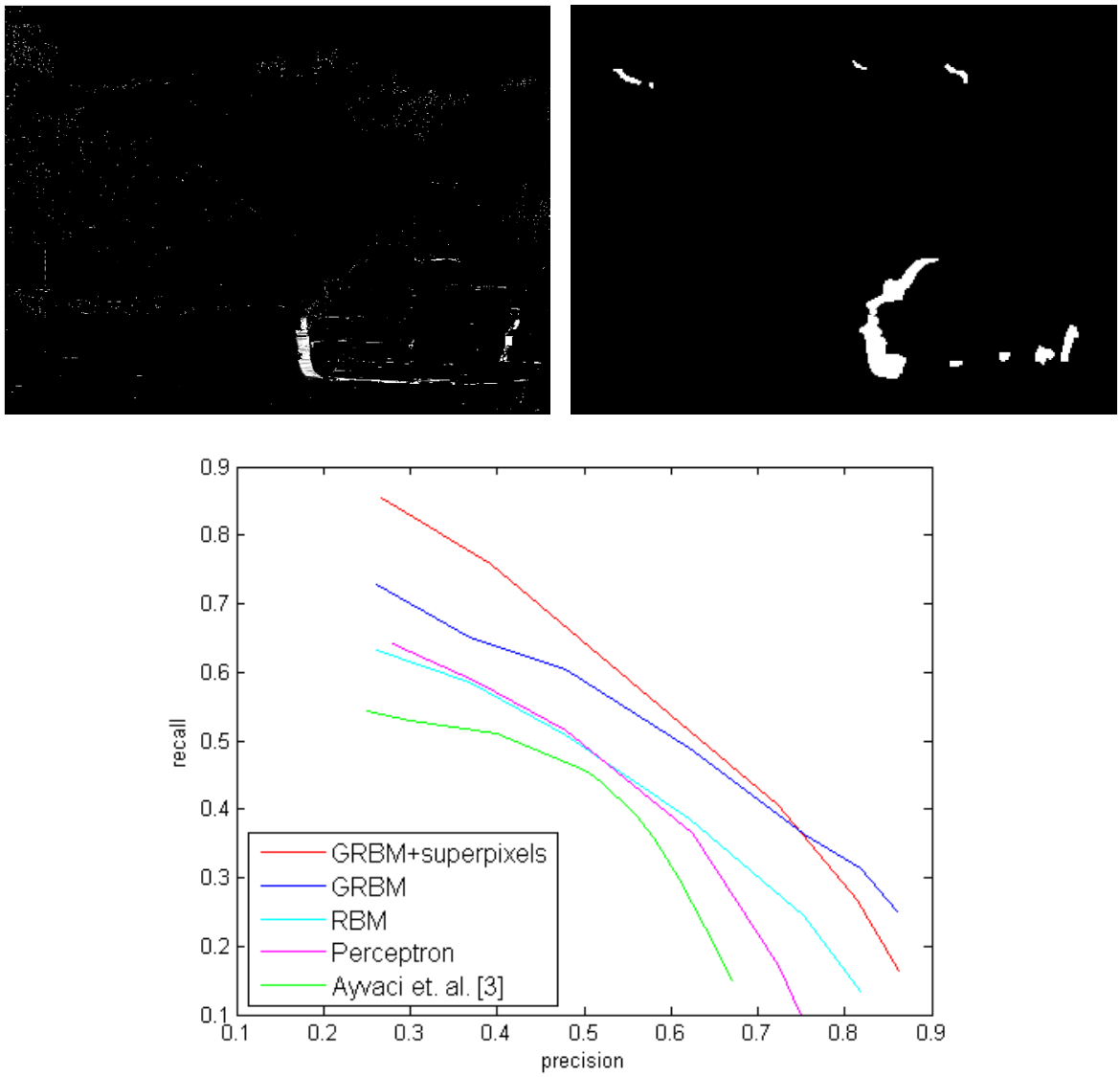


Figure 5.5: Above we compare our algorithm with an optical flow algorithm [11] on the "Cars8" sequence. Our method (right) gives accurate occlusion detection, especially on areas with varying illumination, such as the windscreen and the shadow of the car. Below, the PR curves (extracted on 3 pairs of consecutive frames from different sequence instances) demonstrate improved detection with superpixels, compared to [11] and baseline algorithms based on a one-layer RBM and a two-layer perceptron.

deployed in GRBM's training. The perceptron is trained discriminatively in a binary prediction task (i.e., either the concatenated pair consists of patches with the same appearance except for local affine transformations and illumination variation or not, which is a strong cue for occlusion),

Figure 5.6: The left image pair compares our method with an algorithm that considers both flow and boundary features [74] on the hard, short-baseline "Venus" sequence from the Middlebury dataset. Our method (right) is able to disregard most edges which are not occlusion boundaries. However, although superpixels drive the occlusion boundaries, flow features still occasionally display better behavior on boundaries. In the right image pair we compare our algorithm with a state-of-the-art optical flow algorithm [11] on the "Cars8" sequence. Our method (right) is more accurate, especially on areas with varying illumination, such as the windscreen and the car shadow.

	Venus	RubberWhale	Tsukuba	Text1	BrickBox1t1
r [94]	0.60	0.23	0.44	0.82	0.51
p [94]	0.63	0.31	0.58	0.68	0.49
p [74]	0.69	0.47	0.85	0.88	0.96
p (ours)	0.75	0.81	0.86	0.91	0.92

Table 5.1: Comparison of our occlusion detection algorithm with [94] and [74] on Middlebury and UCL Optical Flow sequences. The comparison is in terms of precision (p) for the *same* recall values (r).

where random image pairs are used as negative samples. It seems that explicitly modeling 3-way interactions without learning any information of included individual images performs better on this specific task.

In Table 5.1 we present a quantitative comparison of our occlusion detection with [94] and [74] at sequences from Middlebury and UCL Optical Flow datasets in terms of precision and recall statistics. Kolmogorov and Zabih [94] designed an algorithm to detect occlusions in stereo image pairs, and therefore, unsurprisingly, their method can not effectively deal with transformations more complex than the horizontal translations, which commonly appear in these sequences. We

Baseline based on differences of intensity averages patchwise

Gated RBM trained on shifts

Gated RBM trained on shifts and rotations

Gated RBM trained on shifts, rotations, affine, scale and illumination variation

Gated RBM trained on shifts, rotations, affine, scale and illumination variation, plus considering superpixel maps

Figure 5.7: Our occlusion detection algorithm for different transformations and against a baseline algorithm between frames 7 and 8 of "Cars8" sequence of Berkeley Motion Segmentation dataset.

use it as a baseline algorithm like in Humayun et al. [74]. The latter ones leverage various flow and appearance features within a learning framework. However, they have many false alarms on edges that are not occlusion boundaries, which originate from the fact that edge detection is one of their component. This behavior becomes obvious also through the qualitative comparison in Fig. 5.6. Finally, in Fig. 5.7 we plot the performance of our occlusion detection algorithm against an increasing number of transformations deployed during the training stage.

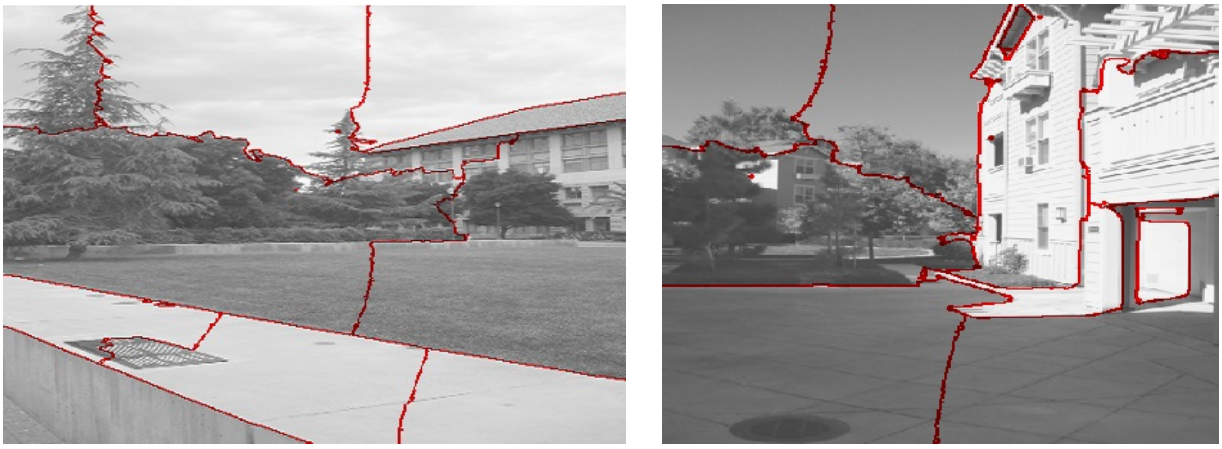
5.3.2 Image Segmentation

In an effort to further investigate network's capability, we challenge it in a binary classification task with intrinsic class variability, image segmentation from a single frame. The distance function from Eq. 5.8 is now used as an estimator of dissimilarity between neighboring patches in a single image. The similarity "discontinuities" (i.e., pairs that have a lower similarity score compared to others) is a cue of object boundaries. After thresholding the distance map, the task becomes a binary decision problem. When a pair is dissimilar according to our comparison framework, their common boundary is considered as object boundary.

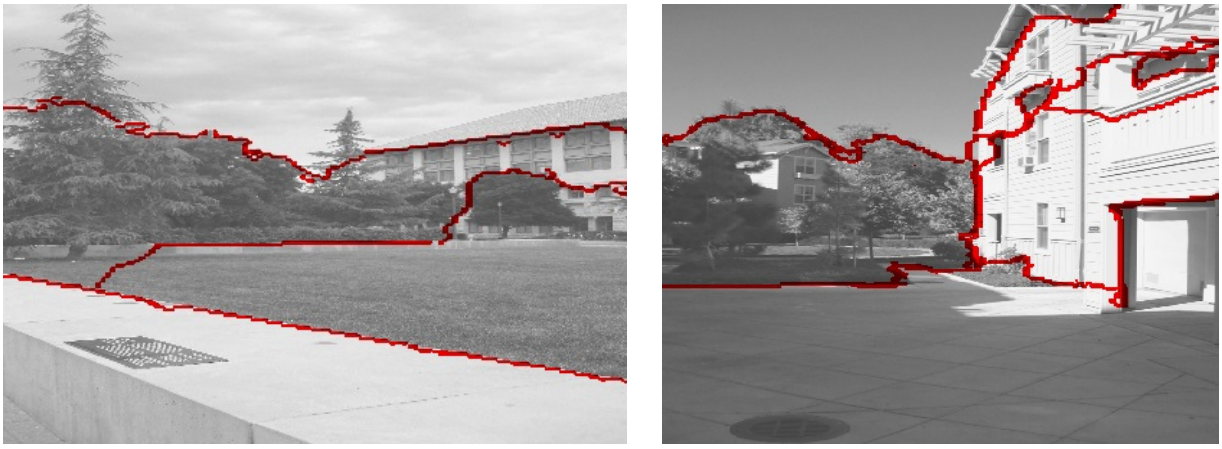
The model is trained over the same spectrum of transformations that were used in occlusion detection (shifts, rotations, affine, illumination, scale) in the same manner as before. After obtaining a binary map of similar/dissimilar neighboring patch pairs, the Normalized Cuts algorithm [143] is used for the final segmentation, where instead of using boundary, brightness or spatial information in the input matrix W, we use Gated RBM's dissimilarity scores:

$$\begin{aligned} &\forall p(i_1, j_1), p(i_2, j_2) \in \mathcal{P} : \\ &w_{12} = \begin{cases} d_{12}, & \text{if } |i_1 - i_2| = 0, r, 2r, 3r \text{ or } |j_1 - j_2| = 0, r, 2r, 3r \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{5.10}$$

where $\mathcal{P}$ is an image partition in overlapping patches, r is the patch size ($r = 13$ in these experiments) and $p(i, j)$ is patch centered on pixel (i, j). Our network is nuisance invariant, thereby capable of ignoring shadow and changing illumination effects and detecting similar textons at different scales, positions and angles. The final segmentation is semantically sensible, in view of the fact that mainly object boundaries are detected instead of other edges which are false alarms in this



(a) Normalized cuts



(b) Normalized cuts guided by nuisance-invariant patch similarity

Figure 5.8: These figures qualitatively demonstrate "semantic" image segmentation. Normalized Cuts (a) and our method (b) are compared. In the left pair, as expected, the final segmentations are similar, but our algorithm successfully disregards any boundary on the front line of the yard wall because a wall exists on both sides. In the right pair Normalized Cuts give a segmentation that follows the shadows. Our algorithm, being illumination invariant, crosses the shade while following the building wall.

setting. Fig. 5.8 demonstrates Normalized cuts examples without (up) and with (down) nuisance-invariant patch similarity. A less sensitive threshold results in a finer segmentation. The images are taken from the Make3d Cornell dataset 1 [137].

5.4 Discussion

We have empirically tested the hypothesis that a fairly simple learning architecture can satisfactorily manage the nuisance variability in the imaging process. To this end, we have established two binary classification tasks; one with intrinsic variability (in segmentation, patches from the same object present intraclass variability) and one without intrinsic variability (in occlusion detection, the underlying scene is known to be the same). We have shown empirically that our network manages to reduce nuisance variability significantly, thus challenging recent work that suggests that nuisance variability accounts for most of the complexity in imaging data [155].

Using multi-scale joint superpixels, our framework provides competitive occlusion detection that in many cases outperforms recent algorithms based on optical flow and boundary features. However, hand-crafting features is a more complicated and time-consuming process. A Gated RBM is capable of learning effective features automatically, while we can specialize the setting and the nuisances that we need to deal with per application by providing appropriate training set.

It should be stressed that our method has no correspondence step in the preprocessing. Given two images, it targets to detect occluded regions and ignore any false alarms arising from local deformations. Inaccuracy in correspondence that is caused by motion is handled by our algorithm to some extent given its invariance in properties such as translations, rotations and scale.

5.5 Appendix - Mathematical Proofs

5.5.1 Proof of Eq. 5.2 (conditional distributions)

Let $\mathbf{x_{-l}}$ denote the state of all units in layer $\mathbf{x}$ except for the lth one and then define the quantities:

$$\alpha_l(\mathbf{y},\mathbf{h}) := -\sum_{f=1}^{F} u_{lf}(\sum_{j=1}^{J} v_{jf}y_j)(\sum_{k=1}^{K} w_{kf}h_k) - a_l,$$

$$\beta(\mathbf{x_{-l}},\mathbf{y},\mathbf{h}) := -\sum_{f=1}^{F}(\sum_{i=1,i\neq l}^{I} u_{if}x_i)(\sum_{j=1}^{J} v_{jf}y_j)(\sum_{k=1}^{K} w_{kf}h_k) - \sum_{i=1,i\neq l}^{I} a_i x_i - \sum_{j=1}^{J} b_j y_j - \sum_{k=1}^{K} c_k h_k.$$

Given the definition of the energy function in Eq. 5.1, we have $E(\mathbf{x},\mathbf{y},\mathbf{h}) = \beta(\mathbf{x_{-l}},\mathbf{y},\mathbf{h}) + x_l\alpha_l(\mathbf{y},\mathbf{h})$. Thus:

$$\begin{aligned}
p(X_l=1|\mathbf{y},\mathbf{h}) = p(X_l=1|\mathbf{x_{-l}},\mathbf{y},\mathbf{h}) &= \frac{p(X_l=1,\mathbf{x_{-l}},\mathbf{y},\mathbf{h})}{p(\mathbf{x_{-l}},\mathbf{y},\mathbf{h})} \\
&= \frac{\frac{1}{Z}e^{-E(X_l=1,\mathbf{x_{-l}},\mathbf{y},\mathbf{h})}}{\frac{1}{Z}e^{-E(X_l=1,\mathbf{x_{-l}},\mathbf{y},\mathbf{h})} + \frac{1}{Z}e^{-E(X_l=0,\mathbf{x_{-l}},\mathbf{y},\mathbf{h})}} \\
&= \frac{e^{-\beta(\mathbf{x_{-l}},\mathbf{y},\mathbf{h})-\alpha_l(\mathbf{y},\mathbf{h})}}{e^{-\beta(\mathbf{x_{-l}},\mathbf{y},\mathbf{h})-\alpha_l(\mathbf{y},\mathbf{h})} + e^{-\beta(\mathbf{x_{-l}},\mathbf{y},\mathbf{h})}} = \frac{e^{-\beta(\mathbf{x_{-l}},\mathbf{y},\mathbf{h})}\cdot e^{-\alpha_l(\mathbf{y},\mathbf{h})}}{e^{-\beta(\mathbf{x_{-l}},\mathbf{y},\mathbf{h})}\cdot(e^{-\alpha_l(\mathbf{y},\mathbf{h})}+1)} \\
&= \frac{1}{1+e^{\alpha_l(\mathbf{y},\mathbf{h})}} = \sigma(-\alpha_l(\mathbf{y},\mathbf{h})) = \sigma[\sum_{f=1}^{F} u_{lf}(\sum_{j=1}^{J} v_{jf}y_j)(\sum_{k=1}^{K} w_{kf}h_k) + a_l],
\end{aligned}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function.

Given the conditional independence of variables $\mathbf{X}$, the joint conditional distribution is written as:

$$p(\mathbf{x}|\mathbf{y},\mathbf{h}) = \prod_{i=1}^{I} \mathcal{B}(x_i;\sigma[\sum_{f=1}^{F} u_{if}(\sum_{j=1}^{J} v_{jf}y_j)(\sum_{k=1}^{K} w_{kf}h_k) + a_i]),$$

where $\mathcal{B}(x;p)$ is the pdf of a Bernoulli random variable x with parameter p, i.e.:

$$\mathcal{B}(x;p) = \begin{cases} p, & \text{if } x=1. \\ 1-p, & \text{if } x=0. \end{cases}$$

with $p = \sigma[\sum_{f=1}^{F} u_{if}(\sum_{j=1}^{J} v_{jf}y_j)(\sum_{k=1}^{K} w_{kf}h_k) + a_i]$. Similar proofs hold for the other two expressions of Eq. 5.2.

5.5.2 Proof of Eq. 5.7 (marginal distribution of the visible variables)

The conditional independence of each layer's units given the other two layers simplifies the calculations:

$$
\begin{aligned}
p(\mathbf{x},\mathbf{y}) &= \frac{1}{Z}\sum_{\mathbf{h}} e^{-E(\mathbf{x},\mathbf{y},\mathbf{h})} \\
&= \frac{1}{Z} e^{\sum_{i=1}^{I} a_i x_i + \sum_{j=1}^{J} b_j y_j} \sum_{h_1} \cdots \sum_{h_K} \prod_{k=1}^{K} e^{h_k\left(c_k + \sum_{f=1}^{F} w_{kf} (\sum_{i=1}^{I} u_{if} x_i)(\sum_{j=1}^{J} v_{jf} y_j)\right)} \\
&= \frac{1}{Z} e^{\sum_{i=1}^{I} a_i x_i} e^{\sum_{j=1}^{J} b_j y_j} \prod_{k=1}^{K} \sum_{h_k} e^{h_k\left(c_k + \sum_{f=1}^{F} w_{kf} (\sum_{i=1}^{I} u_{if} x_i)(\sum_{j=1}^{J} v_{jf} y_j)\right)} \\
&= \frac{1}{Z} \prod_{i=1}^{I} e^{a_i x_i} \prod_{j=1}^{J} e^{b_j y_j} \prod_{k=1}^{K} \left(1 + e^{c_k + \sum_{f=1}^{F} w_{kf} (\sum_{i=1}^{I} u_{if} x_i)(\sum_{j=1}^{J} v_{jf} y_j)}\right).
\end{aligned}
$$

Then the log-likelihood is calculated as:

$$
\begin{aligned}
log\ p(\mathbf{x},\mathbf{y}) &= C + log \prod_{i=1}^{I} e^{a_i x_i} + log \prod_{j=1}^{J} e^{b_j y_j} + log \prod_{k=1}^{K} \left(1 + e^{c_k + \sum_{f=1}^{F} w_{kf} (\sum_{i=1}^{I} u_{if} x_i)(\sum_{j=1}^{J} v_{jf} y_j)}\right) \\
&= C + \sum_{i=1}^{I} a_i x_i + \sum_{j=1}^{J} b_j y_j + \sum_{k=1}^{K} log \left(1 + e^{c_k + \sum_{f=1}^{F} w_{kf} (\sum_{i=1}^{I} u_{if} x_i)(\sum_{j=1}^{J} v_{jf} y_j)}\right).
\end{aligned}
$$

where $C = -logZ$.

CHAPTER 6

Summary of Findings

In this thesis we provide a thorough empirical analysis of the robustness of concurrent deep Convolutional Neural Networks (CNNs) with respect to nuisance transformations, such as domain-size scaling. Based on our discovery that a CNN is not as effective in dealing with simple group transformations, we propose techniques to alleviate this problem that span three directions: algorithmic deployment and model design for CNNs, and learning away nuisances with a Gated Restricted Boltzmann Machine (RBM). Next, we concisely summarize our findings on these directions.

Conditioning CNNs on estimated nuisances

• We conduct an empirical study to test the ability of CNNs to reduce the effects of nuisance transformations of the input data, such as location, scale and aspect ratio. We isolate factors by adopting a common convolutional architecture either deployed globally on the image to compute class posterior distributions, or restricted locally to compute class conditional distributions given location, scale and aspect ratios of bounding boxes determined by proposal heuristics. In theory, averaging the latter should yield inferior performance compared to learned marginalization inside the model. Yet empirical evidence suggests the converse, leading us to conclude that – at the current level of complexity of convolutional architectures and scale of the data sets used to train them – CNNs are not very effective at marginalizing nuisance variability (Chapter 2).

• We quantify the effects of context on the overall classification task and its impact on the performance of CNNs, which justifies the widespread use of object proposals in the detection pipelines. Next, we extend regular data augmentation with proposals in a adaptive, data-driven fashion, we propose a novel pruning technique based on Rényi entropy and improve the end-to-end classification performance to state-of-the-art levels. Additionally, we test our hypothesis on

a wide-baseline matching task using the Oxford and Fischer datasets, where we perform domain-size pooling around regions that are selected by a generic low-level detector. Finally, we explore the use of our sampling techniques in a dense CNN testing scenario, where multiple regions share weights at the convolutional part of the model and their testing is conducted in one single pass. In that case, we achieve comparable performance gains in Imagenet Classification, in a fraction of time compared to our method that is based on region proposals (Chapter 2).

• We present a method to generate object proposals, in the form of bounding boxes in a test image, to be fed to a classifier such as a CNN, in order to reduce test time complexity of object detection and classification. We leverage on filters learned in the lower layers of CNNs to design a binary boosting classifier and deploy higher convolutional CNN layers for a linear regressor in order to discard as many windows as possible that are unlikely to contain objects of interest. We test our method against competing proposal schemes, and end-to-end on the Imagenet detection challenge. We show state-of-the-art performance when at least 1000 proposals per frame are used, at a manageable computational complexity compared to alternate schemes that make heavier use of low-level image processing (Chapter 3).

• We target person re-identification (ReID) from depth sensors such as Kinect. Since depth is invariant to illumination and less sensitive than color to day-by-day appearance changes, a natural question is whether depth is an effective modality for Person ReID, especially in scenarios where individuals wear different colored clothes or over a period of several months. We explore the use of recurrent CNNs for learning high-level shape information from low-resolution depth images. In order to tackle the small sample size problem, we use regularization and propose a hard temporal attention unit. The whole model, along with newly introduced attention layer can be trained end to end with a hybrid supervised loss. We carry out a thorough experimental evaluation of the proposed method on three person re-identification datasets, which include side views, views from the top and sequences with varying degree of partial occlusion, pose and viewpoint variations. To that end, we introduce a new dataset with RGB-D and skeleton data. In a scenario where subjects are recorded after three months with new clothes, we demonstrate large performance gains attained using Depth ReID compared to a state-of-the-art Color ReID. Finally, we show further improvements using the temporal attention unit in multi-shot setting (Chapter 4).

Dealing with nuisances in the CNN architecture

• We study the structure of representations, defined as approximations of minimal sufficient statistics that are maximal invariants to nuisance factors, for visual data subject to scaling and occlusion of line-of-sight. We derive analytical expressions for such representations and show that, under certain restrictive assumptions, they are related to features commonly in use in the computer vision community. Deep convolutional architectures can be understood as implementing successive approximations of an optimal representation by stacking layers of (conditionally) independent local representations, which have been shown to increasingly achieve invariance to large deformations. Thus, as it has been demonstrated for SIFT, we conjectured that pooling over domain size would improve the performance of a convolutional network as well. Domain-size pooling transforms the convolutional neural networks to be scale invariant in the convolutional operator level. We present a reference implementation and experimental results for DSP-CNN in Pascal 2007 Classification, which show a modest improvement over the baseline (Appendix A).

Learning away nuisance variability

• We test the hypothesis that a representation-learning architecture can train away the nuisance variability present in images, owing to noise and changes of viewpoint and illumination. First, we tackle occlusion detection, which is a binary classification task with no intrinsic variability. It amounts to the determination of co-visibility from different images of the same underlying scene. Our network, a Gated Restricted Boltzmann Machine (RBM), learns away the nuisance variability appearing on the background scene and the occluder, which are irrelevant with occlusions, and in turn is capable of discriminating between co-visible and occluded areas by thresholding a one-dimensional semi-metric. Our method, combined with Superpixels, outperforms algorithms using features specifically engineered for occlusion detection, such as optical flow, appearance, texture and boundaries. We further challenge our framework with a more complex task, image segmentation from a single frame. It is cast as binary classification too, but here we also have to deal with the intrinsic variability of the scene objects. We build a similarity map over all patch pairs based on Gated RBM scores and provide a semantic segmentation using Normalized Cuts (Chapter 5).

APPENDIX A

Visual Scene Representations: Contrast, Scaling and Occlusion

A.1 Introduction

A visual representation is a function of past images that is useful to answer questions about the scene given future images from it, regardless of nuisance variability that will affect them. [150] define an optimal representation as a minimal sufficient statistic (of past data for the scene) and a maximal invariant (of future data to nuisance factors), and propose a measure of how "useful" (informative) a representation is, via the uncertainty of the prediction density. What is a nuisance depends on the task, that includes decision and control actions about the surrounding environment, or *scene*, and its geometry (shape, pose), photometry (reflectance), dynamics (motion) and semantics (identities, relations of "objects" within). Depending on the task, nuisance variables may include viewpoint, illumination, sensor calibration, and *occlusion* of line of sight. In this paper we focus on the latter and its impact in the design and learning of representations.

A.1.1 Related Work and Contributions

This paper builds on [150] by focusing on occlusion and scaling phenomena. There, a representation is seen as an approximation of the likelihood function, with nuisance factors either marginalized or profiled (*max-out*). Most work in *low-level vision* handles occlusions by restricting the attention to local regions of the *image*, resulting in representations known as *local descriptors* – too many to review here, with SIFT a prototypical representative [107]. Scale changes are handled by performing computation in *scale-space* [104]. Empirical comparisons abound (*e.g.,* [119]) and recently expanded to include convolutional networks [54]. Our work is aimed at understanding how to relate various descriptors to each other, so the assumptions on which they rely become

patent, and to an "ideal" representation, so one can see how to improve them, not just compare them on any given dataset.

We show that optimal management of nuisance variability due to occlusion is generally intractable, but can be approximated leading to a composite (correspondence) hypothesis test, which provides grounding for the use of "patches" or "receptive fields," ubiquitous in practice (Sect. A.3.2.1). The analysis reveals that the size of the domain of the filters should be *decoupled* from spectral characteristics of the image, unlike traditionally taught in scale-space theory, an unintuitive consequence of the analysis. This idea has been exploited by [43] to approximate the optimal descriptor *of a single image*, under an explicit model of image formation (the Lambert-Ambient, or LA, model) and nuisance variability, leading to DSP-SIFT. Extensions to multiple training images, leading to MV-HoG and R-HoG, has been championed by [41]. Here, we apply domain-size pooling to a convolutional neural network, leading to DSP-CNN, and to deformable part models [52], leading to DSP-DPM, in Sect. A.3.4 and A.3.5 respectively.

A.2 Background

We treat images as random vectors x, y and the scene θ as an (infinite-dimensional) parameter. An optimal representation is a function ϕ of past images $x^t \doteq \{x_1, \dots, x_t\}$ that maximally reduces uncertainty on questions about the scene [58] given images from it and regardless of nuisance variables $g \in G$. In [150] the sampled orbit anti-aliased (SOA) likelihood is introduced as:

$$\hat{L}_{G,\epsilon}(\theta; x) = \max_i \hat{L}(\theta, g_i; x),\ i = 1, \dots, N(\epsilon) \tag{A.1}$$

where

$$\hat{L}(\theta, g_i; x) \doteq \int_G L(\theta, g_i g; x) dP(g) \tag{A.2}$$

and $L(\theta, g; x) \doteq p_{\theta,g}(x)$ is the joint likelihood, understood as a function of the parameter θ and nuisance g for fixed data x, with $dP(g) = w(g^{-1})d\mu(g)$ an *anti-aliasing* measure with positive weights w. There, it is also shown that the SOA likelihood is an optimal representation in the sense that, for any ϵ, it is possible to choose N and a finite number of samples $\{g_i\}_{i=1}^N$ so that $\phi_\theta(x^t) \doteq \hat{L}_{G,\epsilon}(\theta; x^t)$ approximates to within ϵ a minimal sufficient statistic (of x^t for θ) that is

maximally invariant to group transformations in G, *i.e.,* an optimal representation. This result is valid under the assumptions of the Lambert-Ambient (LA) model [42], which is the simplest known to capture the phenomenology of image formation including scaling, occlusion, and rudimentary illumination. For us, what matters of the LA model are three facts: First, the scene *separates* the past from the future: $x^t \perp y \mid \theta$, meaning that $p_\theta(x^t, y) = p_\theta(x^t)p_\theta(y)$. Second, conditioning on viewpoint factorizes the likelihood: If $g \in G = SE(3)$ is the position and orientation of the camera in the reference frame of the scene θ and the image y is made of pixels y_i, then

$$p_\theta(y|g) = \prod_i p_\theta(y_i|g) \tag{A.3}$$

Third, the action of restricted groups $G \subset SE(3)$, for instance planar translations, rotations, scalings, affine and projective transformations, contrast transformations, etc. is *approximately* equivariant, in the sense that for a sufficiently small domain,

$$p_\theta(g_1 y|\bar{g}_2) = p_\theta(y|\bar{g}_1\bar{g}_2) \tag{A.4}$$

where the product $g_1 g_2$ denotes group composition and the bar (omitted henceforth) denotes the embedding of the group action on the (2-D) plane into (3-D) Euclidean space. In Sect. A.3.2 we will motivate these assumptions by restricting the representation to local spatial domains, and use it in Sect. A.3.3 to achieve invariance to arbitrary vantage points. When the task corresponds to a partition of the space of scenes θ, for instance those providing the same answer to a finite collection of questions based on (future) data y and represented by a (supervised, past) training set x^t, then $\phi_{\theta,G}(y, x^t) \doteq \phi_{\theta,G}(y)\phi_\theta(x^t) \simeq \hat{p}_{_{X^t,G}}(y)\hat{p}_{_{X^t}}(x^t) \propto \hat{p}_{_{x^t,G}}(y)$ can be considered a "learned approximation" of an optimal representation. Next, we illustrate how to compute such an approximation explicitly under the assumptions of the LA model.

Remark 1 (Active Sensing)**.** *A representation, informative as it may be, can be no more informative than the data itself, uninformative as it may be. This is irrelevant in our context, for we are seeking statistics that are* as informative as the (training) data *(sufficient), however good or bad that is. For the representation to (asymptotically) approach the informative content of the* scene, *it is necessary to* design the experiment *so that the data collected x^t, with $t \to \infty$, yields statistics that are asymptotically* complete *[50]. Such active learning or active sensing is beyond the scope of this paper.*

When a single training datum is given, $x^t = x$, no intrinsic (intra-class) variability can be learned, and the variability in the data is ascribed to the nuisances. The representation for $t = 1$ thus reduces to

$$\phi_{x,G}(y) = p_G(y|x) \tag{A.5}$$

which is approximated locally by DSP-SIFT [43].

A.3 Learning Visual Representations

In [150], it is shown that the orbit likelihood of the LA model is maximally invariant and minimally sufficient. Thus, visual representations can be trained or designed to compute the SOA likelihood with respect to nuisances that include illumination, viewpoint (with the associated scale changes), and partial occlusions.

A.3.1 Contrast invariance

Contrast transformations are monotonic continuous transformation of the (range space of the) data. If applied globally to an image, they are a crude approximation of changes in the image due to illumination. However, applied locally and independently in each receptive field, they can capture complex illumination effects. As we will see, occlusion will force our representation to be restricted to local statistics, so we adopt local contrast transformation as a model of illumination changes. It is well-known that the curvature of the level sets is a maximal invariant [4]. Since the gradient orientation is everywhere orthogonal to the level sets, it is also a maximal contrast invariant. The following expression for the invariant is obtained via marginalization of the norm of the gradient for a single training image, since the action of contrast is independent at each pixel.

Theorem 1 (Contrast invariant[1])**.** *Given a training image x and a test image y, assuming that the latter is affected by noise that is independent in the gradient direction and magnitude, then the maximal invariant of y to the group G of contrast transformations is given by*

$$p_{x,G}(y) = p(\angle\nabla y|x)\ \|\nabla x\|. \tag{A.6}$$

[1]Proof of Theorem 1 is derived in [149].

The independence assumption above is equivalent to assuming that the gradient magnitude and orientation of y are related to the gradient magnitude and orientation of x by a simple additive model: $\|\nabla y\| = \|\nabla x\| + n_\rho$ and $\angle\nabla y = \angle\nabla x \oplus n_\alpha$, where $\oplus$ denotes addition modulo 2π, and n_ρ and n_α are independent. These are all modeling assumptions, clearly not strictly satisfied in practice, but reasonable first-order approximations. Note that, other than for the gradient, the computations above can be performed point-wise, so we could write (A.6) at each pixel y_i: if $\alpha \doteq \angle\nabla y_i$,

$$\phi_x(\alpha) = \prod_i \mathcal{N}_{\mathbb{S}^1}(\alpha_i - \angle\nabla x_i; \epsilon_\alpha)\|\nabla x_i\| \tag{A.7}$$

In the rest of the paper, we use the symbol α to denote the orientation of the image gradient relative to one of the coordinate axes, and omit the subscript G when referring to contrast (since the use of the argument α makes it unambiguous). The width of the kernel ϵ_α is a design (regularization) parameter.

Remark 2 (No invariance for x)**.** *Note that* (A.7) *is invariant to contrast transformations of y, but* not *of x. For a single training image, the latter can be handled by* normalization *as we will see next. For multiple images, the factor can in principle be different for each training image.*

Remark 3 (Bayesian invariant)**.** *In the proof of Theorem 1, the gradient magnitude is marginalized with respect to the base measure. With a different prior, for instance arising from bounds on the gradient or from statistics of natural images, marginalization yields a factor other than $\|\nabla x\|$. Clamping, described next, can be understood as a particular choice of prior for marginalization of the gradient magnitude.*

Invariance to contrast transformations in the (single) *training* image can be performed by normalizing the likelihood, which in turn can be done in a number of ways. If contrast transformations are globally affine, then the joint likelihood can be normalized by simply dividing by the integral over α, which is the ℓ^1 norm of the histogram across the entire image/patch

$$\frac{\phi_x(\alpha)}{\|\phi_x(\alpha)\|_{\ell^1}} = \frac{p(\alpha|x)\|\nabla x\|}{\int p(\alpha|x)d\alpha\|\nabla x\|} = p(\alpha|x) \tag{A.8}$$

that should be used instead of the customary ℓ^2 [107]. If the contrast transformation is non-linear, it cannot be eliminated by global normalization.

Remark 4 (Clamping)**.** *When the joint distribution is approximated by the product of marginals, as in [107], joint normalization is still favored in practice as it introduces some correlations among marginal histograms [35]. However, cells with large gradients tend to dominate the histogram, pushing all other peaks lower. Alternatively, one could independently normalize each of the histograms, $\phi_{x_i}(\alpha)$ and then concatenate them. But this has the opposite effect: Cells with faint peaks, once re-normalized, are given undue importance and relative intensity difference between different cells are discarded. A common trick consisting of joint normalization (so faint cells do not prevail) followed by "clamping" (saturation of the maximum to a fraction of the value of the highest peak, so large gradients do not dominate), and then re-normalization, seems to achieve a tradeoff between the two [107]. This process can also be understood as a way of marginalizing ρ, with respect to a different measure $dP(\rho)$, as described in Rem. 3 while assuming that, within each region, contrast transformations are affine.*

Once invariance to contrast transformations is achieved, which can be done on a single image x, we are left with nuisances G that include general viewpoint changes, including the occlusions they induce. This can be handled by computing the SOA likelihood with respect to the product G of $SE(3)$ (the group of general rigid motions, Sect. A.3.3) from a training sample x^t, leading to

$$\hat{L}(\theta, g_i; x^t) = \Big\{ \int_G \phi_{x^t}(\alpha | g_i \circ g) dP(g) \Big\}_{i=1}^{N} \tag{A.9}$$

In the next section we show how to handle occlusions, and in the following one general viewpoint changes.

A.3.2 Occlusions

We do not know ahead of time what portion of an object or scene, seen in training images, will be visible in a test image. Occlusion, or visibility, is arguably the single most critical aspect of visual representations. It enforces *locality*, as dealing with occlusion nuisances entails searching through, or marginalizing, all possible (multiply-connected) subsets of the test image. This power set is clearly intractable even for very small images.

A.3.2.1 Bypassing shape and justifying "patches" or "receptive fields"

We illustrate a principle to bypass combinatorial explosion for a single training image, absent all other nuisances. A training x and a test image y *correspond* (hypothesis H_0) if there exist subsets of x, Ω_x, and of y, Ω_y, such that the restrictions come from the same scene, *i.e.,* in this setting they differ by a white (zero-mean, uninformative) residual.[2] Under this simplistic model, the subsets $\Omega_x = \Omega_y \doteq \Omega$ are the same, and $y = x + n$ where n is either a white (spatially i.i.d) zero-mean process with a small covariance, $n_{|\Omega} \sim \mathcal{N}(0, \epsilon^2)$ in the corresponding region, or something else, for instance uniform with a mean in the order of magnitude of the intensity range, assumed normalized to one, $n_{|\Omega^c} \sim \mathcal{U}$. Hypothesis H_1 is that there exists no such region, and $n \sim \mathcal{U}$ on the entire domain. Since we do not know the region Ω, this is a composite hypothesis testing problem, where the likelihood ratio is given by

$$\frac{p(y|x, H_0)}{p(y|x, H_1)} = \frac{\max_\Omega p(y_{|\Omega}|x_{|\Omega}, H_0)p(y_{|\Omega^c}|x_{|\Omega^c}, H_0)}{\max_\Omega p(y_{|\Omega}|x_{|\Omega}, H_1)p(y_{|\Omega^c}|x_{|\Omega^c}, H_1)} = \frac{\max_\Omega \mathcal{N}(y_{|\Omega} - x_{|\Omega}; \epsilon^2)}{\max_\Omega \mathcal{U}(y_{|\Omega} - x_{|\Omega})} \tag{A.10}$$

Missed detections (treating a co-visible pixel as occluded) and *false alarms* (treating an occluded pixel as visible) have different costs: Omitting a co-visible pixel from Ω decreases the likelihood by a factor corresponding to multiplication by a Gaussian for samples drawn from the same distribution; vice-versa, including a pixel from Ω^c (false alarm) decreases the log-likelihood by a factor equal to multiplying by a Gaussian evaluated at points drawn from another distribution, such as uniform. So, testing for correspondence on *subsets of the co-visible regions*, assuming the region is sufficiently large, reduces the power, but not the validity, of the test. This observation can be used to *fix the shape* of the regions, *leaving only their size to be marginalized, or searched over.*[3] This reasoning justifies the use of "patches" or "receptive fields" to seed image matching, but emphasizes that a search over different *sizes* [43] is needed.

Together with the SOA likelihood, this also justifies the local marginalization of *domain sizes*, along with translation, as recently championed in [43].

[2]Of course, absent all other nuisances, all pixels are independent so corresponding regions can be determined by "background subtraction" techniques. This requires absence of other nuisances, so the example serves just to illustrate the principle.

[3]Alternatively, the sampling can be framed as a sequential hypothesis test for joint matching and domain size estimation, as in region-growing or quickest setpoint change detection.

Corollary 1 (DSP-SIFT)**.** *The DSP-SIFT descriptor [43] approximates an optimal representation* (A.9) *for G the group of planar similarities and local contrast transformations, when the scene is a single training image, and the test image is restricted to an unknown subset of its domain.*

SIFT as designed violates the sampling principles described here, as sampling occurs with respect to the full similarity group (positions, scales and rotations are selected using a co-variant detector), but *anti-aliasing* is only performed in position (spatial pooling) and orientation (histogram smoothing), *not in scale*, which in SIFT is tied to domain size.

A.3.3 General viewpoint changes

If a co-variant translation-scale *and size* sampling/anti-aliasing mechanism is employed, then around each sample the only residual variability to viewpoint $SE(3) = \mathbb{R}^3 \times SO(3)$ is reduced[4] to $SO(3)$. That can be further factored into a rotation of the image plane ("in-plane" rotation), and its complement ("out-of-plane" rotation). We next show how in-plane rotations can be eliminated, leaving only out-of-plane rotations.

Canonization is the process by which a co-variant detector (a functional of the data and a chosen group whose zero-level set identifies isolated elements of the group that co-vary with it) is used to determine (multiple) local reference frames with respect to which the data is, by construction, invariant to the chosen group [147]. This procedure is particularly well suited to deal with planar rotation, since the statistics of natural images ensure that with high probability orientation-co-variant detectors have few isolated extrema. An example is the local maximum of the norm of the gradient along the direction $\alpha = \hat{\alpha}_l(x)$.[5] Invariance to $G = SO(2)$ can be achieved by retaining the samples

$$p_\theta(\alpha|G) = \{p_\theta(\alpha|\hat{\alpha}_l)\}_{l=1}^L \tag{A.11}$$

[4]In reality, translation in space is not equivalent to translation and scaling of the image plane, for the former induces deformations of the image domain due to parallax effects and occlusions, which are absent in the latter. However, locally and away from occlusions, one is a first-order approximation of the other, so the derivation is valid for each local region that does not straddle an occluding boundary, justified by our handling of occlusions via the restriction to receptive fields in Sect. A.3.2.

[5]Here g acts on x via $gx(u_i, v_i) = x(u_i'', v_i'')$ where $u'' = u\cos\alpha - v\sin\alpha$ and $v'' = u\sin\alpha + v\cos\alpha$, and a canonical element $\hat{g}_l(x) = \hat{\alpha}$ can be obtained as $\hat{\alpha} = \arg\max_\alpha \|\nabla x(u_i', v_i')\|$. The corresponding rotation invariant $\hat{g}^{-1}(x)x$ is $\angle\nabla x(u_i', v_i')$ where $u' = u\cos\alpha + v\sin\alpha$ and $v' = -u\sin\alpha + v\cos\alpha \doteq \alpha'$.

Rotation anti-aliasing is performed by regularizing the orientation histogram. Note that, as it was for contrast, planar rotations can affect both the training x and the test image y. In some cases, a consistent reference (canonical element) is available for both when scenes or objects are geo-referenced: The projection of the gravity vector onto the image plane [81, 40]. In this case, $L = 1$, and $\hat{\alpha}$ is the angle of the projection of gravity onto the image plane (well defined unless they are orthogonal):

$$p_\theta(\alpha|G) = p_\theta(\alpha|\hat{\alpha}). \tag{A.12}$$

In reality, rotation canonization should contend with spatial quantization, neglected here since rotation errors are absorbed by the binning of gradient orientation ϵ_α.

This leaves out-of-plane rotations to be managed. Unfortunately, the effects of such rotations on future images depend on the shape of the underlying scene, which is unknown, and that cannot be determined from a single image. Therefore, the only way in which true viewpoint changes can be factored out of the representation is if multiple training images *of the same scene* are available. [41] have proposed extensions of local descriptors based on a sampling approximation of the likelihood function, $\hat{p}_\theta$, or on a point estimate of the scene $p_{\hat{\theta}}$, multi-view HOG and reconstructive HOG respectively. The estimated scene has a geometric component (shape) $\hat{S}$ and a photometric component (radiance) $\hat{\rho}$, inferred from the LA model as described in [42]. These in turn enable the approximation of the predictive likelihood $p_{\hat{\theta},G}$, and hence the representation:

$$\phi_{\hat{\theta},G}(\alpha_i) = \int_{SO(3)} \mathcal{N}_{\mathbb{S}^1}(\alpha_i - \hat{\rho} \circ g \circ \pi_{\hat{S}}^{-1}(u_j, v_j); \epsilon_\alpha) \|\nabla \hat{\rho}\| \mathcal{N}_\sigma(i-j) d\mu(j) dP_{SO(3)}(g) \tag{A.13}$$

where $\hat{\theta} = (\hat{S}, \hat{\rho})$, $\angle \nabla y = \alpha$ and π^{-1} is the pre-image of a perspective projection (the point of first intersection of the ray through the pixel (u_j, v_j) with the surface $\hat{S}$). Alternatively, a sampling approximation of the likelihood function $\hat{p}_\theta(x^t)$ yields "multi-view HOG"

$$\phi_G(\alpha_i|x^t) \doteq \frac{1}{t} \sum_\tau \int_{\mathbb{R}^2} \mathcal{N}_{\mathbb{S}^1}(\alpha_i - \angle \nabla x_{\tau j}; \epsilon_\alpha) \mathcal{N}_\sigma(i-j) d\mu(j) dP(\sigma) \tag{A.14}$$

Note that the gradient weight $\|\nabla x_\tau\|$ is absent, since individual samples of past data do not enable separating nuisance from intrinsic variability, and each sample image x_τ has different contrast, so the factor cannot be simply eliminated by normalization as done in Rem. 3 for a single image. Therefore, in MV-HOG it is necessary to assume that training images are captured under the same

illumination conditions. In MV-HOG, regularization is implicit in the kernel, and the predictive likelihood is based on simple planar transformations. In R-HOG, the estimated scene (which requires regularization to be inferred) acts as the regularizer [41]. Once the effects of occlusions are considered (which force the representation to be local), and the effects of general viewpoint changes are accounted for (which creates the necessity for multiple training images of the same scene), a maximal contrast/viewpoint/occlusion invariant can be approximated via the SOA likelihood. Using (A.13), the SOA likelihood (A.9) becomes:

$$\hat{L}_{SE(3),\epsilon(N)}(\alpha_i) = \max_k \Big\{ \int_{SO(3)} \mathcal{N}_{\mathbb{S}^1}(\alpha_i - \hat{\rho} \circ g_k g \circ \pi_{\hat{S}}^{-1}(x_j); \epsilon_\alpha) \kappa_\sigma(i-j) d\mu(j) dP(\sigma) dP_{SO(3)}(g)) \Big\}_{k=1}^{N} \tag{A.15}$$

The assumption that all existing multiple-view extensions of SIFT do *not* overcome is the conditional independence of the intensity of different pixels (A.3). This is discussed in [150] for the case of deep convolutional architectures.

A.3.4 Domain-Size Pooling in Convolutional Neural Networks (DSP-CNN)

Deep convolutional architectures can be understood as implementing successive approximations of an optimal representation by stacking layers of (conditionally) independent local representations of the form (A.15), which have been shown by [150] to increasingly achieve invariance to large deformations, despite locally marginalizing only affine (or similarity) transformations. As [43] did for SIFT, we conjectured that pooling over domain size would improve the performance of a convolutional network.

After a brief review of a classic convolutional neural network [98], we next formally present the proposed method. We show experiments to test the conjecture using a pre-trained network which is fine-tuned with domain-size pooling on benchmark datasets. For computational reasons, we limited the domain-size pooling to 6 sizes (including the base size), and only to the first convolutional layer. Still, the experiments show marginal improvement. We conjecture that more thorough incorporation of domain-size pooling would yield further performance benefits.

A.3.4.1 Convolutional neural networks

Let $I : D \subset \mathbb{R}^2 \to \mathbb{R}^+; x \mapsto I(x)$ be the input grayscale image. Let l be the layer index and define

$$P_l^i : D_l \to \mathbb{R}; x \mapsto P_l^i(x) \tag{A.16}$$

$P_l^i(\cdot)$ is the i-th (max- or mean-) pooled feature map at layer l and $P_l^i(x)$ is its value at location x. Let $P_l : D_l \times Z^+ \to \mathbb{R}$ be all feature maps for layer l and $P_0(x) = I(x)$, *i.e.,* 0-layer is the input image itself. Also define

$$C_l^i : D_l' \to \mathbb{R}; x \mapsto C_l^i(x) \tag{A.17}$$

$C_l^i(\cdot)$ denotes the i-th (un-pooled) feature map at layer l and $D_l \subseteq D_l'$.

Given the above notation, the convolutional neural network (CNN) can be recursively defined as

$$C_{l+1}^i(m,n) = f[\sum_{\substack{\forall(\mu,v)\in F_{l+1}\\ \forall z\in Z_{l+1}}} K_{l+1}^i(\mu,v,z) P_l(m-\mu, n-v, J_{l+1}^i(z)) + B_{l+1}^i],$$

$$(m,n) \in D_{l+1}' \qquad \text{CONV} \tag{A.18}$$

$$P_{l+1}^i(\hat{x}) = \underset{\forall x\in\mathcal{B}_{2\times 2}(\hat{x})}{pool} C_{l+1}^i(x),\ x \in D_{l+1}' \text{ and } \hat{x} \in D_{l+1} \qquad \text{SPATIAL POOLING} \tag{A.19}$$

where $K_{l+1}^i : F_{l+1} \times Z_{l+1} \to \mathbb{R}$ is the kernel for feature map i at layer $l+1$ ($F_{l+1} \subset \mathbb{R}^2$ is the kernel support at layer $l+1$ and $Z_{l+1} \subset Z^+$ is the number of channels which that kernel spans). Therefore, the convolution is applied on 2 dimensions, but the convolution kernel is three-dimensional, as the feature map i at layer $l+1$ is computed from several pooled feature maps from the previous layer l (where $J_{l+1}^i(\cdot)$ is the set of their indices.) $B_{l+1}^i \in \mathbb{R}$ is the bias term for the i kernel (feature) of $l+1$ layer. f can be $\tanh(\cdot)$ function or the ReLU operator $\max\{\cdot, 0\}$. In Eqn. (A.18), f is applied component-wise.

Typically, the first several layers of a CNN are computed by a series of convolution and pooling operations. Because of the stride in sampling feature map and the pooling operation, C_l or P_l will be eventually reduced to a $1 \times 1 \times N$ matrix where N is the number of feature maps in this last layer. Upon now an image I is "encoded" into a vector which will be the input to a standard fully-connected multilayer neural network to produce the final vector representation for classification.

A.3.4.2 Domain-size pooling

Let's modify the standard model and introduce pooling over S domain sizes for each convolutional operation. At each convolutional layer l, we apply each kernel i over S different scales[6], and therefore we get S response maps per feature, before pooling over scale. We define

$$C^i_{l,s} : D'_{l,s} \rightarrow \mathbb{R};\ x \mapsto C^i_{l,s}(x),\ s \in \{1, \ldots, S\} \tag{A.20}$$

where $C^i_{l,s}(\cdot)$ is the i-th feature map at layer l and scale s. All convolutional maps over scale have the same size, given that convolutions are computed around the same locations over different scales, modulo size variation on the boundaries because of different kernel sizes. These artifacts can be resolved by cropping or padding all maps to D'_l with the minimum (average) activation, which does not affect the max- (average-) pooling operations that follow. Thus, all feature maps $\hat{C}^i_{l,s} : D'_l \rightarrow \mathbb{R}$ have the same support region D'_l and next they are pooled over S scales.

The network's main components can be reformulated as:

$$C^i_{l+1,s}(m,n) = f[\sum_{\substack{\forall(\mu,v)\in F_{l+1,s} \\ \forall z \in Z_{l+1}}} K^i_{l+1,s}(\mu, v, z) P_l(m-\mu, n-v, J^i_{l+1}(z)) + B^i_{l+1}],$$

$$(m,n) \in D'_{l+1}, s \in \{1, \ldots, S\} \qquad \text{CONV} \tag{A.21}$$

$$P^i_{l+1}(\hat{x}) = \underset{\forall x \in \mathcal{B}_{2\times 2}(\hat{x})}{pool}\ \underset{\forall s \in S}{pool}\ \hat{C}^i_{l+1,s}(x),\ x \in D'_{l+1}, \hat{x} \in D_{l+1} \quad \text{DS \& SPATIAL POOLING} \tag{A.22}$$

where $K^i_{l+1,s} : F_{l+1,s} \times Z_{l+1} \rightarrow \mathbb{R}$ and $B^i_{l+1} \in \mathbb{R}$ are kernel and bias terms correspondingly for the i feature at $l+1$ layer and s scale. $F_{l+1,s},\ s \in \{1, \ldots, S\}$ are S kernel sizes which can be obtained via bilinear interpolation with uniformly sampled ratio in the neighborhood of the normalized unit scale (*e.g.,* for 5 scales, ratios $\{0.6, 0.8, 1, 1.2, 1.4\}$ are chosen). The *pool* operator can be max or mean for both spatial and domain-size pooling.

[6]The computations take place at the same locations where single-scale convolutions are computed, given the layer's size, stride and padding parameters. This process can be implemented either with domain or kernel warping, and generates a structure that resembles with a *truncated cone*.

Another way to write these components is to extend the convolutional operator to directly pool over different domain sizes, as follows:

$$C^i_{l+1}(m,n) = f[\sum_{\substack{\forall(\mu,v)\in F_{l+1,s}\\ \forall z\in Z_{l+1}\\ \forall s\in S}} K^i_{l+1,s}(\mu,v,z) P_l(m-\mu, n-v, J^i_{l+1}(z)) + B^i_{l+1}],$$

$$(m,n) \in D'_{l+1} \qquad \text{CONV WITH DS POOLING} \quad \text{(A.23)}$$

$$P^i_{l+1}(\hat{x}) = \underset{\forall x\in\mathcal{B}_{2\times 2}(\hat{x})}{pool} C^i_{l+1}(x),\ x \in D'_{l+1}, \hat{x} \in D_{l+1} \qquad \text{SPATIAL POOLING} \quad \text{(A.24)}$$

where each symbol has been defined above.

A.3.4.3 Implementation and experiments

To the best of our knowledge, HMAX [139] and Locally Scale-invariant CNNs [83] are the only network that pools across different scales, but yet no approach that we know of pools across different domain sizes. The representation is built in successive stages, where at each stage n the representation θ_n is represented by a distribution of images x, from which one can sample.

To compare a domain-size pooled CNN (DSP-CNN) to an ordinary CNN, we use a discriminatively pretrained model on Imagenet (ILSVRC-2012 [136]), which we fine-tune on Pascal 2007 train-validation data [48] and test on the VOC test set. Considering that Pascal VOC is a multi-label dataset, the softmax regression loss can be replaced with either one-vs-rest classification hinge loss or a ranking hinge loss. To keep things simple (with an expected small performance loss) we keep softmax and augment the training set so that images with multiple labels are entered in the pipeline at each epoch as many times as labels they have.

The pretrained on ILSVRC-2012 model which is described as *CNN-S* in [26] is used in our experiments. Then similar fine-tuning protocol with the one suggested by the authors is followed for both the ordinary and the DSP model. In order to control overfitting, we use the following learning rate schedule: 10^{-3} / 10^{-4} (epochs $1-15$), 10^{-4} / 10^{-4} (epochs $16-35$), 10^{-5} / 10^{-5} (epochs $36-50$) (first/second number pertain to last/hidden layers correspondingly). Average-pooling over domain-size is deployed for *conv1* layer (6 sizes; ratios 0.55, 0.7, 0.85, 1, 1.25, 1.5). No data augmentation is applied for either training or testing, which explains the lower reported

mAP statistics compared to the numbers that are reported in [26]. Our interest is only to evaluate the relative merit of DSP, so the absolute numbers are not as relevant.

All trainings were performed using the MatConvNet toolbox [165]. Representative results are shown in Fig. A.1 and Table A.1. The improvement is marginal but nevertheless present. We conjecture that more thorough experimentation with domain-size pooling in convolutional architectures may reveal more improvements in performance.

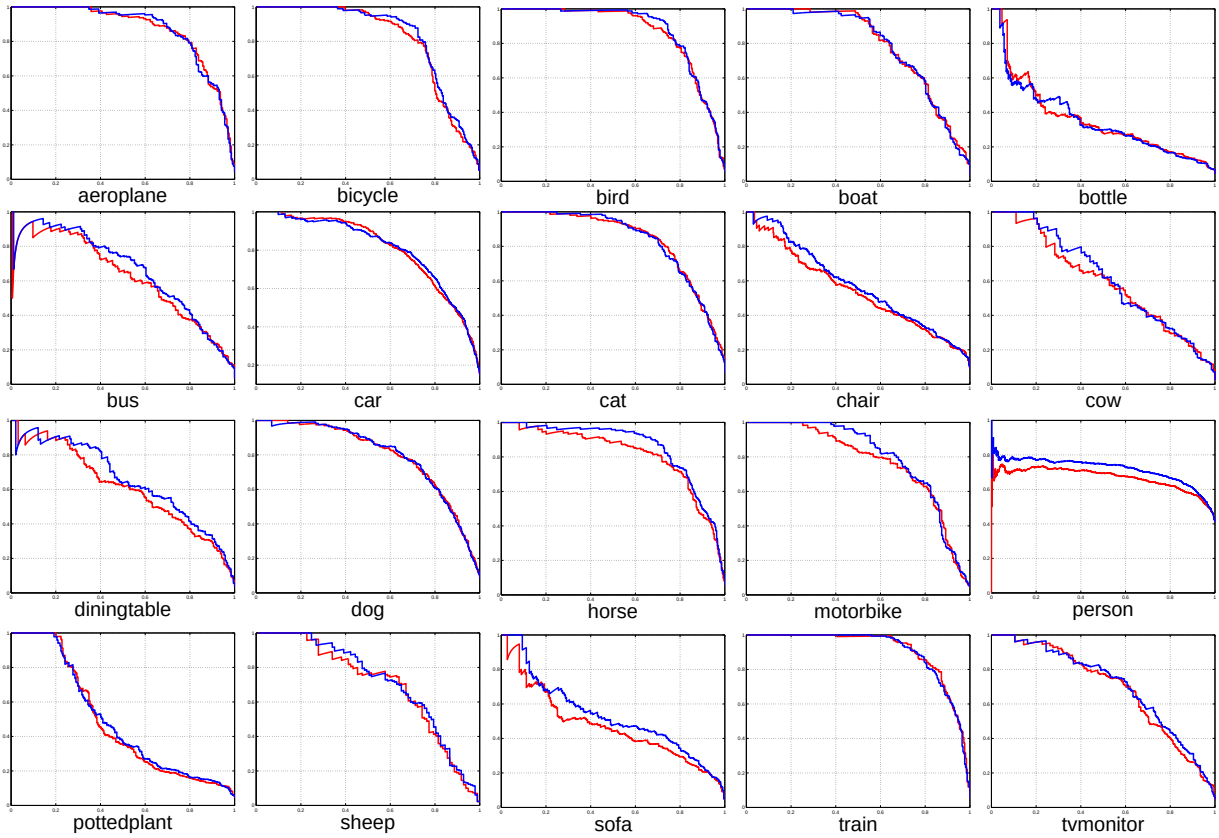


Figure A.1: Precision-recall curves over 20 classes in the Pascal 2007 Classification Challenge. DSP-CNN is plotted in blue, while the original CNN in red.

A.3.5 Domain-Size Pooling in Deformable Part Models (DSP-DPM)

We have also developed domain-size pooling extensions of deformable part models (DPMs) [52], small trees of local HOG descriptors ("parts"), whereby local photometry is encoded in the latter (nodes), and geometry is encoded in their position on the image relative to the root node (edges).

	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow
CNN	.8813	**.8331**	.8566	.8205	**.4223**	.6965	.8456	.8320	.5779	.6826
DSP-CNN	**.8840**	.8309	**.8570**	**.8228**	.4186	**.7169**	**.8504**	**.8513**	**.6014**	**.6857**

	Table	Dog	Horse	Motorbike	Person	Plant	Sheep	Sofa	Train	TV Monitor	mAP
CNN	.6693	.8100	.8259	.8053	.8779	.5351	.7549	.5765	.8722	.6786	.7427
DSP-CNN	**.7014**	**.8253**	**.8261**	**.8109**	**.8903**	**.5619**	**.7703**	**.6388**	**.8887**	**.7171**	**.7575**

Table A.1: PASCAL VOC 2007 Classification Challenge.

Intra-class shape variability is captured by the posterior density of edge values, learned from samples. Photometry is captured by a "HOG pyramid" where the *size* of each part is pre-determined and fixed relative to the root. Interpreting the photometric descriptor as a likelihood function, rather than a "feature vector," helps interpreting DPM as a (factorized) posterior density, where photometry is encoded by the SOA likelihood. One could therefore conjecture that performing anti-aliasing with respect to the size of the parts would improve performance.

A.3.5.1 Implementation and experiments

Deformable part model (DPM) consists of a fixed set of HOG templates (one root and several parts) and the set of learnable deformation costs for all the parts. Domain-size pooling can be applied to the low level HOG descriptors, yielding DSP-DPM. We sampled 10 domain sizes ranging from 0.5σ to 1.5σ where σ is the original size used for HOG computation. By average pooling of the HOGs computed from each domain sizes, we obtain a dense DSP-HOG response for the whole image. These DSP-HOGs are used to train the deformable model for each object in the PASCAL VOC 2007 detection challenge[7] [48]. The results are reported in Table A.2. DSP-DPM outperforms the vanilla DPM in most object categories in terms of mean average precision. Among these categories, we found they are mostly classes of animals whose configurations are more versatile and thus more likely to hit occlusions. In other cases when objects are less "deformable", the performances between two DPMs are close. Moreover, at test time, the original DPM samples scale very densely (10 intermediate levels between two octaves), which is observed to be critical

[7]Note that we evaluate the performance of DSP-CNN on PASCAL VOC 2007 Classification task, but DSP-DPM on the Detection task.

	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	
DPM	.3221	.5814	.0835	.1212	.2931	.5241	**.5733**	.2221	.2101	.2447	
DSP-DPM	**.3526**	**.5951**	**.1070**	**.1377**	**.3022**	**.5246**	.5679	**.2668**	**.2281**	**.3042**	
	Table	Dog	Horse	Motorbike	Person	Plant	Sheep	Sofa	Train	TV Monitor	mAP
DPM	**.2803**	.1215	.6078	.4604	.4020	.1246	.1745	**.3248**	.4243	**.4470**	.3271
DSP-DPM	.2760	**.1329**	**.6149**	**.4634**	**.4127**	**.1345**	**.2005**	.3193	**.4538**	.4318	**.3413**

Table A.2: PASCAL VOC 2007 Detection Challenge.

to achieve a good performance [52]. In that case, the effect of DS-pooling becomes less obvious.

A.4 Conclusions

We have derived an expression (A.15) for minimal sufficient statistics of past data when the test image is restricted to a neighborhood of y where α_i is computed, corresponding to sampled locations around (u_k, v_k), with scales σ pooled according to the prior $dP(\sigma)$ around the samples σ_k. If a *sufficiently exciting* training set is available, spanning variability due to out-of-plane rotations, marginalization of $SO(3)$ can be replaced by temporal averaging of the training images (A.14). The joint distribution of local descriptors can be captured by a stacked architectures, as shown in [150] and illustrated here for deformable parts models and deep convolutional architectures.

[1] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[2] A. Albiol, J. Oliver, and J. M. Mossi. Who is who at different cameras: people re-identification using depth cameras. In *IET Computer Vision*, 2012.

[3] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[4] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel. Axioms and fundamental equations of image processing. In *Archive for Rational Mechanics and Analysis*, 1993.

[5] V. Andersson, R. Dutra, and R. Araújo. Anthropometric and human gait identification using skeleton data from kinect sensor. In *ACM Symposium on Applied Computing*, 2014.

[6] F. Anselmi, L. Rosasco, and T. Poggio. On invariance and selectivity in representation learning. In *Journal on Information and Inference*, 2016.

[7] O. M. Aodha, A. Humayun, M. Pollefeys, and G. Brostow. Learning a confidence measure for optical flow. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[8] N. Apostoloff and A. W. Fitzgibbon. Learning spatiotemporal T-Junctions for occlusion detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[9] R. Appel, T. Fuchs, P. Dollár, and P. Perona. Quickly boosting decision trees-pruning underachieving features early. In *Journal of Machine Learning Research*, 2013.

[10] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[11] A. Ayvaci, M. Raptis, and S. Soatto. Sparse Occlusion Detection with Optical Flow. In *International Journal of Computer Vision*, 2012.

[12] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *International Journal of Computer Vision*, 2011.

[13] I. Barbosa, M. Cristani, A. Del Bue, L. Bazzani, and V. Murino. Re-identification with rgb-d sensors. In *European Conference on Computer Vision - workshops*, 2012.

[14] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[15] M. V. D. Bergh, G. Roig, X. Boix, S. Manen, and L. V. Gool. Online video seeds for temporal window objectness. In *IEEE International Conference on Computer Vision*, 2013.

[16] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: A CPU and GPU math compiler in Python. In *Proceedings of 9th Python in Science Conference*, 2010.

[17] H. Bilen, M. Pedersoli and T. Tuytelaars. Weakly supervised object detection with posterior regularization. In *British Machine Vision Conference*, 2014.

[18] L. Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, 2012.

[19] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *European Conference on Computer Vision*, 2010.

[20] J. Bruna and S. Mallat. Invariant scattering convolution networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[21] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[22] F. M. Castro, M. J. Marín-Jiménez, N. Guil, and N. P. de la Blanca. Automatic learning of gait signatures for people identification. In *arXiv:1603.01006*, 2016.

[23] F. M. Castro, M. J. Marín-Jimenez, and R. Medina-Carnicer. Pyramidal fisher motion for multiview gait recognition. In *IEEE International Conference on Pattern Recognition*, 2014.

[24] W. Chan, N. Jaitly, Q. Le, and O. Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[25] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai. Fusing generic objectness and visual saliency for salient object detection. In *International Conference on Computer Vision*, 2011.

[26] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[27] D. Chen, Z. Yuan, B. Chen, and N. Zheng. Similarity learning with spatial constraints for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[28] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu. A new LDA-based face recognition system which can solve the small sample size problem. In *Pattern recognition*, 2000.

[29] M. Cheng, Z. Zhang, W. Lin, and P. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[30] Y.-J. Cho and K.-J. Yoon. Improving person re-identification via pose-aware multi-shot matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[31] R. G. Cinbis, J. Verbeek, and C. Schmid. Segmentation driven object detection with fisher vectors. In *International Conference on Computer Vision*, 2013.

[32] T. Cohen and M. Welling. Group Equivariant Convolutional Networks. In *International Conference on Machine Learning*, 2016.

[33] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A Matlab-like environment for machine learning. In *BigLearn, workshop in Neural Information Processing Systems*, 2011.

[34] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[35] N. Dalal, and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[36] S. Ding, L. Lin, G. Wang, and H. Chao. Deep feature learning with relative distance comparison for person re-identification. In *Pattern Recognition*, 2015.

[37] P. Dollár. Image and video Matlab toolbox. 2013.

[38] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *British Machine Vision Conference*, 2009.

[39] J. Donahue, L.A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[40] J. Dong, X. Fei, and S. Soatto. Visual inertial semantic scene representation for 3D object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[41] J. Dong, N. Karianakis, D. Davis, J. Hernandez, J. Balzer, and S. Soatto. Multiview feature engineering and learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[42] J. Dong and S. Soatto. The Lambert-Ambient shape space and the systematic design of feature descriptors. In *Machine Learning for Computer Vision, chapter Visual Correspondence*, R. Cipolla, S. Battiato, G.-M. Farinella (Eds), Springer Verlag, 2014.

[43] J. Dong and S. Soatto. Domain-size pooling in local descriptors: DSP-SIFT. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[44] A. Dosovitskiy, P. Fischer, J. Springenberg, M. Riedmiller, and T. Brox. Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[45] A. Dubois, and F. Charpillet. A gait analysis method based on a depth camera for fall prevention. In *IEEE International Conference on Engineering in Medicine and Biology Society*, 2014.

[46] I. Endres and D. Hoiem. Category independent object proposals. In *European Conference on Computer Vision*. 2010.

[47] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[48] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. In *International Journal of Computer Vision*, 2010.

[49] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[50] V. V. Fedorov. Theory of optimal experiments. In *Elsevier*, 1972.

[51] P. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 2004.

[52] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[53] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun. Salient object detection by composition. In *International Conference on Computer Vision*, 2011.

[54] P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor matching with convolutional neural networks: a comparison to sift. In *arXiv:1405.5769*, 2014.

[55] A. Fischer and C. Igel. An Introduction to Restricted Boltzmann Machines. In *Journal in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, 2012.

[56] B. J. Frey, N. Jojic, and A. Kannan. Learning appearance and transparency manifolds of occluded objects in layers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[57] A. Bedagkar-Gala and S. K. Shah. A survey of approaches and trends in person re-identification. In *Image and Vision Computing*, 2014.

[58] D. Geman, S. Geman., N. Hallonquist, and L. Younes. Visual Turing test for computer vision systems. In *Proceedings of the National Academy of Sciences*, 2015.

[59] R. Gens, and P. Domingos. Deep symmetry networks. In *Advances in Neural Information Processing Systems*, 2014.

[60] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[61] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[62] S. Gong, M. Cristani, S. Yan, and C. C. Loy. Person re-identification. In *Springer*, 2014.

[63] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng. Measuring invariances in deep networks. In *Advances in Neural Information Processing Systems*, 2009.

[64] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *IEEE European Conference on Computer Vision*, 2008.

[65] D. B. Grimes and R. P. N. Rao. Bilinear sparse coding for invariant vision. In *Journal of Neural Computation*, 2005.

[66] A. Haque, A. Alahi, and L. Fei-Fei. Recurrent attention models for depth-based person identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[67] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE International Conference on Computer Vision*, 2015.

[68] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[69] X. He and A. L. Yuille. Occlusion boundary detection using pseudo-depth. In *European Conference on Computer Vision*, 2010.

[70] G. E. Hinton. Training products of experts by minimizing contrastive divergence. In *Neural Computation*, 2002.

[71] M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, and G. Rigoll. The TUM Gait from Audio, Image and Depth (GAID) database: Multimodal recognition of subjects and traits. In *Journal of Visual Communication and Image Representation*, 2014.

[72] D. Hoiem, A. A. Efros, and M. Hebert. Recovering Occlusion Boundaries from an Image. In *International Journal of Computer Vision*, 2010.

[73] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[74] A. Humayun, O. M. Aodha, and G. J. Brostow. Learning to find occlusion regions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[75] A. Humayun, F. Li, and J. M. Rehg. RIGOR: Reusing inference in graph cuts for generating object regions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[76] D. Ioannidis, D. Tzovaras, I. G. Damousis, S. Argyropoulos, and K. Moustakas. Gait recognition using compact feature extraction transforms and depth information. In *IEEE Transactions on Information Forensics and security*, 2007.

[77] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

[78] N. Jacobson, Y. Freund, and T. Q. Nguyen. An online learning approach to occlusion boundary detection. In *IEEE Transactions on Image Processing*, 2012.

[79] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, 2014.

[80] P. M. Jodoin, C. Rosenberger, and M. Mignotte. Detecting half-occlusion with a fast region-based fusion procedure. In *British Machine Vision Conference*, 2006.

[81] E. Jones, and S. Soatto. Visual-inertial navigation, localization and mapping: A scalable real-time large-scale approach. In *Internation Journal of Robotics Research*, 2011.

[82] A. Kale, N. Cuntoor, B. Yegnanarayana, A. N. Rajagopalan, and R. Chellappa. Gait analysis for human identification. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, 2003.

[83] A. Kanazawa, A. Sharma, and D. Jacobs. Locally scale-invariant convolutional neural networks. In *Deep Learning and Representation Learning Workshop: NIPS*, 2014.

[84] H. Kang, M. Hebert, A. Efros, and T. Kanade. Data-driven objectness. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[85] N. Karianakis, J. Dong, and S. Soatto. An empirical evaluation of current convolutional architectures' ability to manage nuisance location and scale variability. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[86] N. Karianakis, T. Fuchs, and S. Soatto. Boosting convolutional features for robust object proposals. In *arXiv:1503.06350*, 2015.

[87] N. Karianakis, Z. Liu, Y. Chen and S. Soatto. Person Depth ReID: Robust person re-identification with commodity depth sensors. In *arXiv:1705.09882*, 2017.

[88] N. Karianakis, Y. Wang and S. Soatto. Learning to discriminate in the wild: Representation-learning network for nuisance-invariant image comparison. *Technical Report, UCLA Computer Science Department*, 2013.

[89] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[90] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *European Conference on Computer Vision*, 2014.

[91] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[92] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Person re-identification by unsupervised ℓ_1 graph learning. In *European Conference on Computer Vision*, 2016.

[93] M. Koestinger, M. Hirzer, P. Wohlhart, P.M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[94] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. In *IEEE International Conference on Computer Vision*, 2001.

[95] I. Kviatkovsky, A. Adam, and E. Rivlin. Color invariants for person re-identification. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[96] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[97] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. In *Nature*, 2015.

[98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998.

[99] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *International Conference on Artificial Intelligence and Statistics*, 2015.

[100] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith. Learning locally-adaptive decision functions for person verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[101] W. Li, R. Zhao, T. Xiao, and X. Wang. DeepReID: Deep filter pairing neural network for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[102] S. Liao, Y. Hu, X. Zhu, and S. Z. Li. Person re-identification by local maximal occurrence representation and metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[103] K. P. Lim, A. Das, and M. N. Chong. Estimation of occlusion and dense motion fields in a bidirectional bayesian framework. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[104] T. Lindeberg. Feature detection with automatic scale selection. In *International Journal of Computer Vision*, 1998.

[105] G. Lisanti, I. Masi, A. D. Bagdanov, and A. Del Bimbo. Person re-identification by iterative re-weighted sparse ranking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[106] E. Lobaton, R. Vasudevan, R. Bajcsy, and R. Alterovitz. Local occlusion detection under deformations using topological invariants. In *European Conference on Computer Vision*, 2010.

[107] D. G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, 2004.

[108] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*, 2015.

[109] B. Ma, Y. Su, and F. Jurie. Local descriptors encoded by fisher vectors for person re-identification. In *IEEE European Conference on Computer Vision - Workshops*, 2012.

[110] L. Ma, X. Yang, and D. Tao. Person re-identification over camera networks using multi-task distance metric learning. In *IEEE Transactions on Image Processing*, 2014.

[111] S. Manen, M. Guillaumin, and L. V. Gool. Prime object proposals with randomized Prim's algorithm. In *IEEE International Conference on Computer Vision*, 2013.

[112] A. I. Mansur, Y. Makihara, R. Aqmar, and Y. Yagi. Gait recognition under speed transition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[113] D. R. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[114] N. Martinel, A. Das, C. Micheloni, and A. K. Roy-Chowdhury. Temporal model adaptation for person re-identification. In *IEEE European Conference on Computer Vision*, 2016.

[115] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. In *Image and Vision Computing*, 2004.

[116] N. McLaughlin, J. Martinez del Rincon, and P. Miller. Recurrent convolutional network for video-based person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[117] R. Memisevic and G. E. Hinton. Learning to represent spatial transformations with factored higher-order Boltzmann machines. In *Neural Computation*, 2010.

[118] A. Mignon and F. Jurie. PCCA: A new approach for distance learning from sparse pairwise constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[119] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. In *International Journal of Computer Vision*, 2005.

[120] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, 2014.

[121] A. Mogelmose, T. B. Moeslund, and K. Nasrollahi. Multimodal person re-identification using RGB-D sensors and a transient identification database. In *IEEE International Workshop on Biometrics and Forensics*, 2013.

[122] G. Mori. Guiding model search using segmentation. In *IEEE International Conference on Computer Vision*, 2005.

[123] M. Munaro, A. Basso, A. Fossati, L. Van Gool, and E. Menegatti. 3D reconstruction of freely moving persons for re-identification with a depth sensor. In *IEEE International Conference on Robotics and Automation*, 2014.

[124] M. Munaro, S. Ghidoni, D. T. Dizmen, and E. Menegatti. A feature-based approach to people re-identification using skeleton keypoints. In *IEEE International Conference on Robotics and Automation*, 2014.

[125] B. Munsell, A. Temlyakov, C. Qu, and S. Wang. Person identification using full-body motion and anthropometric biometrics from Kinect videos. In *European Conference on Computer Vision - workshops*, 2012.

[126] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[127] S. Paisitkriangkrai, C. Shen, A. van den Hengel. Learning to rank in person re-identification with metric ensembles. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[128] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian. Local fisher discriminant analysis for pedestrian re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[129] M. Piccardi. Background subtraction techniques: a review. In *International Conference on Systems, Man and Cybernetics*, 2004.

[130] T. Poggio. The computational magic of the ventral stream. In *Nature Precedings*, 2011.

[131] B. Prosser, W.-S. Zheng, S. Gong, T. Xiang, and Q. Mary. Person re-Identification by support vector ranking. In *British Machine Vision Conference*, 2010.

[132] E. Rahtu, J. Kannala, and M. Blaschko. Learning a category independent object detection cascade. In *International Conference on Computer Vision*, 2011.

[133] P. Rantalankila, J. Kannala, and E. Rahtu. Generating object segmentation proposals using global and local search. In *Conference on Computer Vision and Pattern Recognition*, 2014.

[134] M. A. Ranzato, A. Krizhevsky, and G. E. Hinton. Factored 3-Way Restricted Boltzmann Machines for modeling natural images. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[135] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[136] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. In *International Journal of Computer Vision*, 2015.

[137] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[138] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations*, 2014.

[139] T. Serre, A. Oliva, and T. Poggio. A feedforward architecture accounts for rapid categorization. In *Proceedings of the National Academy of Sciences*, 2007.

[140] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *International Journal of Computer Vision*, 2002.

[141] C. E. Shannon. A mathematical theory of communication. In *ACM SIGMOBILE Mobile Computing and Communications Review*, 2001.

[142] H. Shi, Y. Yang, X. Zhu, S. Liao, Z. Lei, W. Zheng, and S. Z. Li. Embedding deep metric for person re-identification: A study against large variations. In *European Conference on Computer Vision*, 2016.

[143] J. Shi and J. Malik. Normalized Cuts and image segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[144] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. In *Communications of the ACM*, 2013.

[145] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations*, 2014.

[146] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[147] S. Soatto. Steps towards a theory of visual information: Active perception, signal-to-symbol conversion and the interplay between sensing and control. In *arXiv:1110.2053*, 2012.

[148] S. Soatto, J. Dong, and N. Karianakis. Visual scene representations: Contrast, scaling and occlusion. In *International Conference on Learning Representations - workshop*, 2015.

[149] S. Soatto, J. Dong, and N. Karianakis. Visual scene representations: Contrast, scaling and occlusion. *Technical report, UCLA CSD:140024 - extended version*, 2014.

[150] S. Soatto and A. Chiuso. Visual representations: Defining properties and deep approximation. In *International Conference on Learning Representations*, 2016.

[151] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. In *Journal of Machine Learning Research*, 2014.

[152] A. N. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. In *International Journal of Computer Vision*, 2009.

[153] C. Su, S. Zhang, J. Xing, W. Gao, and Q. Tian. Deep attributes driven multi-camera person re-identification. In *IEEE European Conference on Computer Vision*, 2016.

[154] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[155] G. Sundaramoorthi, P. Petersen, V. S. Varadarajan, and S. Soatto. On the set of images modulo viewpoint and contrast changes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[156] P. Sundberg, T. Brox, M. Maire, P. Arbelez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[157] J. Susskind, G. E. Hinton, R. Memisevic, and M. Pollefeys. Modeling the joint density of two images under a variety of transformations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[158] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[159] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov. Scalable, high-quality object detection. In *arXiv:1412.1441*, 2014.

[160] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[161] D. Tao, L. Jin, Y. Wang, Y. Yuan, and X. Li. Person re-identification by regularized smoothing kiss metric learning. In *IEEE Transactions on Circuits and Systems for Video Technology*, 2013.

[162] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *European Conference on Computer Vision*, 2010.

[163] Y. W. Teh and G. E. Hinton. Rate-coded Restricted Boltzmann Machines for face recognition. In *Advances in Neural Information Processing Systems*, 2000.

[164] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

[165] A. Vedaldi and K. Lenc. MatConvNet: Convolutional neural networks for Matlab. In *ACM International Conference on Multimedia*, 2015.

[166] R. Vezzani, D. Baltieri, and R. Cucchiara. People re-identification in surveillance and forensics: A survey. In *ACM Computing Surveys*, 2013.

[167] P. Viola and M. J. Jones. Robust real-time face detection. In *International Journal of Computer Vision*, 2004.

[168] V. Vonikakis, D. Chrysostomou, R. Kouskouridas, and A. Gasteratos. A biologically inspired scale-space for illumination invariant feature selection. In *Measurement Science and Technology*, 2013.

[169] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[170] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *International Conference on Computer Vision*, 2013.

[171] T. Whytock, A. Belyaev, and N. M. Robertson. Dynamic distance-based shape features for gait recognition. In *Journal of Mathematical Imaging and Vision*, 2014.

[172] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*, 1992.

[173] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[174] F. Xiong, M. Gou, O. Camps, and M. Sznaier. Person re-identification using kernel-based metric learning methods. In *European Conference on Computer Vision*, 2014.

[175] Y. Yan, B. Ni, Z. Song, C. Ma, Y. Yan, and X. Yang. Person re-identification via recurrent feature aggregation. In *European Conference on Computer Vision*, 2016.

[176] Y. Yang, J. Yang, J. Yan, S. Liao, D. Yi, and S. Z. Li. Salient color names for person re-identification. In *European Conference on Computer Vision*, 2014.

[177] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *IEEE International Conference on Computer Vision*, 2015.

[178] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Deep metric learning for person re-identification. In *IEEE International Conference on Pattern Recognition*, 2012.

[179] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, 2014.

[180] W. Zaremba and I. Sutskever. Learning to execute. In *arXiv:1410.4615*, 2014.

[181] P. Zehnder, E. Koller-Meier, and L. V. Gool. An efficient shared multi-class detection cascade. In *British Machine Vision Conference*, 2008.

[182] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, 2014.

[183] W. Zeng, C. Wang, and F. Yang. Silhouette-based gait recognition via deterministic learning. In *Pattern Recognition*, 2014.

[184] Z. Zhang, J. Warrell, and P. Torr. Proposal generation for object detection using cascaded ranking SVMs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[185] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[186] G. Zhao, G. Liu, H. Li, and M. Pietikainen. 3D gait recognition using multiple cameras. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2006.

[187] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[188] R. Zhao, W. Ouyang, and X. Wang. Learning mid-level filters for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[189] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian. Mars: A video benchmark for large-scale person re-identification. In *European Conference on Computer Vision*, 2016.

[190] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *IEEE International Conference on Computer Vision*, 2015.

[191] W.-S. Zheng, S. Gong, and T. Xiang. Re-identification by relative distance comparison. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[192] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, 2014.