Analysis and Improvement of the Visual Object Detection Pipeline

Jan Hosang, Dipl.-Inform.

A dissertation submitted towards the degree
Doctor of Engineering (Dr.-Ing.)
of the Faculty of Mathematics and Computer Science
of Saarland University

Saarbrücken, März 2017

Day of Colloquium	2nd of May, 2017
Dean of the Faculty	Univ.-Prof. Dr. Frank-Olaf Schreyer Saarland University, Germany

Examination Committee

Chair	Prof. Dr. Matthias Hein
Reviewer, Advisor	Prof. Dr. Bernt Schiele
Reviewer	Prof. Dr. Vittorio Ferrari
Academic Assistant	Dr. Mykhaylo Andriluka

Abstract

Visual object detection has seen substantial improvements during the last years due to the possibilities enabled by deep learning. While research on image classification provides continuous progress on how to learn image representations and classifiers jointly, object detection research focuses on identifying how to properly use deep learning technology to effectively localise objects. In this thesis, we analyse and improve different aspects of the commonly used detection pipeline.

We analyse ten years of research on pedestrian detection and find that improvement of feature representations was the driving factor. Motivated by this finding, we adapt an end-to-end learned detector architecture from general object detection to pedestrian detection. Our deep network outperforms all previous neural networks for pedestrian detection by a large margin, even without using additional training data.

After substantial improvements on pedestrian detection in recent years, we investigate the gap between human performance and state-of-the-art pedestrian detectors. We find that pedestrian detectors still have a long way to go before they reach human performance, and we diagnose failure modes of several top performing detectors, giving direction to future research. As a side-effect we publish new, better localised annotations for the Caltech pedestrian benchmark.

We analyse detection proposals as a preprocessing step for object detectors. We establish different metrics and compare a wide range of methods according to these metrics. By examining the relationship between localisation of proposals and final object detection performance, we define and experimentally verify a metric that can be used as a proxy for detector performance.

Furthermore, we address a structural weakness of virtually all object detection pipelines: non-maximum suppression. We analyse why it is necessary and what the shortcomings of the most common approach are. To address these problems, we present work to overcome these shortcomings and to replace typical non-maximum suppression with a learnable alternative. The introduced paradigm paves the way to true end-to-end learning of object detectors without any post-processing.

In summary, this thesis provides analyses of recent pedestrian detectors and detection proposals, improves pedestrian detection by employing deep neural networks, and presents a viable alternative to traditional non-maximum suppression.

Zusammenfassung

Die visuelle Objektdetektion erfuhr in den letzten Jahren durch die Möglichkeiten von Deep Learning erhebliche qualitative Verbesserungen. Während durch die Forschung zur Bildklassifizierung kontinuierliche Fortschritte darin erzielt werden, wie Merkmalsrepräsentation und Klassifikatoren gemeinsam gelernt werden, konzentriert sich die Forschung zur Objektdetektion darauf, wie Deep Learning verwendet werden kann, um Objekte schnell und genau zu lokalisieren. In dieser Arbeit analysieren und verbessern wir verschiedene Aspekte des häufig verwendeten Objektdetektions-Prozesses.

Wir analysieren den Fortschritt von zehn Jahren Forschung an Fußgängererkennung und finden heraus, dass die Verbesserung von Merkmalsrepräsentationen den Schlüsselfaktor darstellt. Durch diese Erkenntnis motiviert, adaptieren wir ein tiefes neuronales Netzwerk zur allgemeinen Objekterkennung, das Merkmalsrepräsentation und Klassifikatoren gemeinsam lernt, für die Fußgängererkennung. Unser Netzwerk übertrifft alle bisherigen neuronalen Netze für die Fußgängererkennung bei Weitem, sogar wenn keine zusätzlichen Trainingsdaten verwendet werden.

Nach signifikanten Verbesserungen der Fußgängererkennung in den letzten Jahren untersuchen wir den qualitativen Unterschied zwischen menschlicher Leistung und Ergebnissen von Fußgängerdetektoren auf dem neuesten Stand der Technik. Unsere Experimente zeigen, dass Fußgängerdetektoren noch einen langen Weg vor sich haben um menschliche Qualität zu erreichen. Wir untersuchen Fehler von mehreren starken Fußgängerdetektoren und charakterisieren häufige Fehlerquellen. Ein Nebenprodukt dieser Arbeit sind neue und besser lokalisierte Annotationen für den Caltech Fußgängerdetektions-Benchmark.

Wir analysieren Erkennungsvorschläge (detection proposals) als Vorverarbeitungsschritt für Objektdetektion. Wir definieren verschiedene Metriken und vergleichen eine breite Palette von Methoden nach diesen Metriken. Durch die Untersuchung der Beziehung zwischen der Lokalisierung von Erkennungsvorschlägen und der endgültigen Objektdetektionsleistung definieren und verifizieren wir experimentell eine Metrik, die als Stellvertreter für die Detektorleistung verwendet werden kann.

Darüber hinaus behandeln wir eine strukturelle Schwäche von praktisch allen Objekterkennungs-Prozessen: Unterdrückung nicht-maximaler Detektionen. Wir analysieren, warum dieser Schritt notwendig ist und was die Unzulänglichkeiten des gebräuchlichen Ansatzes sind. Um diese Probleme zu lösen, stellen wir Forschung vor, die diese Mängel überwindet und die die typische Unterdrückung durch eine erlernbare Alternative ersetzt. Das vorgestellte Paradigma ebnet den Weg zu echtem "End-to-End-Lernen" von Objektdetektoren, die keine weitere Nachbearbeitung benötigen.

Zusammenfassend stellt diese Dissertation Analysen der jüngsten Fußgänger-Detektoren und Erkennungsvorschläge vor, verbessert die Fußgängererkennung durch den Einsatz tiefer neuronaler Netze und präsentiert eine tragfähige Alternative zur herkömmlichen Unterdrückung nicht-maximaler Detektionen.

Contents

1 Introduction

Computers are ubiquitous nowadays. Many people have a desktop computer at work or a mobile phone in their pocket, even wrist watches are starting to be computers connected to the internet. Phones automatically organise our holiday photos and count our steps to assist us in building healthy habits. Home automation is conveniently controlling heating, ventilation, and security of houses, car engine control units regulate air-fuel mixture and idle speed. Computers have found their way into many objects, machines, and parts of our lives to improve them by saving money, assisting us and saving time, or assisting us to improve our safety.

As the tasks that are autonomously solved by computers are becoming more complex, they are moving away from predefined, mundane chores and towards tasks that are impossible to approach by explicitly designing an algorithm to directly solve the problem. The most successful approaches to many of these tasks are driven by machine learning, such as natural language processing and speech recognition, image understanding, handwriting recognition, medical drug discovery, credit card fraud detection, human motion capturing, and many more.

Computer vision defines a field of problems that are primarily concerned with visual modalities, such as images or videos, but also laser depth images, x-ray energy images, or 3 dimension computer tomography images. Intuitively this field can be described as “teaching computers to see”. There is a large variety of applications that is enabled by computer vision, ranging from convenience applications such as image search, over financially motivated applications such as analysing customer behaviour in a shop, shops without checkout, or autonomous vehicles (“side-walk robots”, cars, trucks, flying drones, container ships), all the way to security applications that have the potential to save lives, such as automatic breaking in vehicles (mandatory for new cars in the EU[1]) and surveillance (anomaly detection, tracking of people).

Recent advances in machine learning, enabled by faster computing hardware and more available data, have transformed the field of computer vision. Highly non-linear models with a large number of parameters (deep neural networks), often referred to as deep learning, have shown incredible performance improvements. While representations and algorithms were traditionally engineered by humans, deep networks enable us to learn most components directly from data with few assumptions. The resulting improvement in quality of applications was so great, that many applications that previously have been researched in academia have successfully transitioned into the industry and our everyday life. This transition happened relatively quickly, within one or two years, which underlines the usefulness of computer vision.

[1] https://ec.europa.eu/growth/sectors/automotive/safety_en

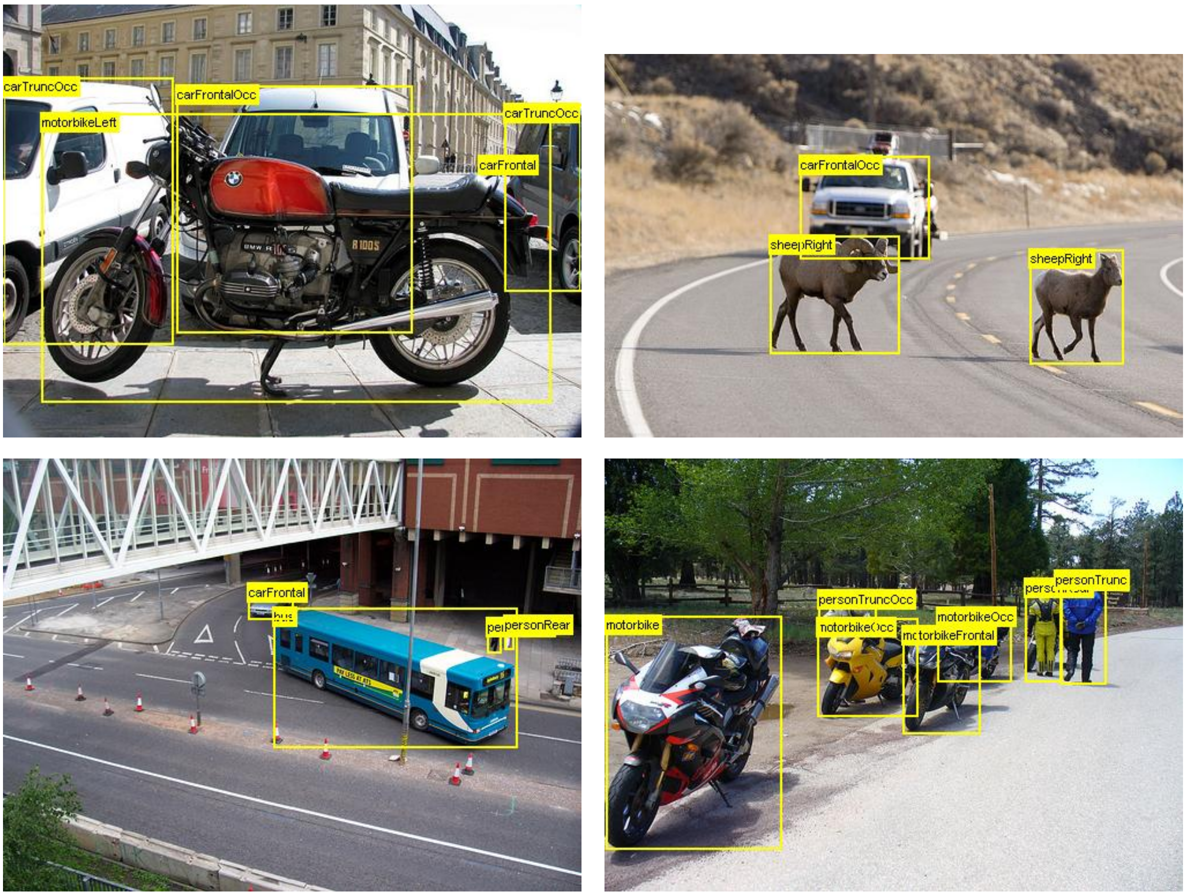


Figure 1.1: Examples from the Pascal VOC benchmark (Everingham *et al.*, 2007b). Yellow boxes are manual annotations by humans.

This dissertation focuses on object detection, the task of both recognising and localising object classes in images. Object detection is a core problem of computer vision, as many applications directly use an object detector as a building block or indirectly profit from the insights gained in object detection research. Typically the problem is posed as finding all instances of a certain object class (e.g. people, cars, cows, bottles, chairs) in an image and accurately localising it by marking it with a tight bounding box as illustrated in figure 1.1.

In essence object detection research seeks to find the tools for learning representations that enable fast and high-quality recognition and localisation. This involves not only proposing promising representations, but also identifying suitable ways of reducing the problem to existing machine learning methodologies, with time constrains and limited training data in mind.

1.1 Challenges of object detection

This section gives a short overview of the challenges of object detection in the context of recent advancements in deep learning. More details about typical approaches of solving the problem can be found with related work in chapter 2.

Object recognition. Some of the core challenges in object detection are inherent to computer vision and are also being researched in image classification. Objects in images can have large variance due to view point, illumination, articulation, and appearance changes that we might wish to ignore. Since machine learning models are being trained on specific instances, learning needs to abstract away the specifics of the training instances in order to generalise to unseen instances at test time.

Traditionally these problems were approached by using a powerful classifier, such as support vector machines (SVM), Adaboost, decision trees, or (for today's standards small) neural networks, and engineering representations that explicitly accomplish the desired abstraction, making the image content palatable to the classifier. A typical and successful approach involves image gradients (Lowe, 2004; Dalal and Triggs, 2005): Specific pixel colors and lighting are converted into differences between neighbouring pixels, which is more stable across different instances of an object class or different lighting conditions. Building histogram statistics over small image regions discards precise locations of gradients, while keeping approximate locations. This accomplishes to abstract away small deformations.

In modern deep learning the distinction between features and classifiers disappears. Modern neural networks directly map images to classification predictions, effectively functioning as a feature extractor and classifier at the same time. This avoids hand designing representations and instead learns the representation jointly with the classifier. Compared to previous classification pipelines, current neural networks can have an enormous number of model parameters that are learned from data, providing large capacity for feature extraction and classification. As a result many challenges of recognition are solved by these models without explicit modelling, e.g. previously mentioned appearance and lighting changes, or deformations. Sometimes the importance of particular challenges is historically so established that we can still find explicit handling in neural networks, even though they are later discarded, as the community discovers more important modifications of neural networks. One such example is contrast normalisation used by the landmark work of Krizhevsky *et al.* (2012). Overall, we see a clear trend towards simpler (fewer ingredients), larger, and deeper networks (He *et al.*, 2016a).

Object localisation. Typical object detection approaches approach the search problem of finding a small object in a larger image by sliding a window through the image and classifying the image content of each window location. This can be seen as reducing the localisation problem to the classification problem, potentially benefiting from advances in image classification. On the other hand it is important to note that image classification seeks to be mostly invariant to object sizes and locations ("Is there a car *anywhere* in the image?”), while for the sliding window approach location and scale is crucial (“Is there a car and does it fill the entire sliding window?”).

As mentioned before, some challenges of object detection are also addressed by research on classification. Some challenges are specific to the fact that object detection needs to localise potentially small objects. Objects may appear at almost any location and at almost any size, which poses the problem of how to efficiently reduce the search space. The sliding window approach applies recognition and localisation many times over the same image, which requires to discover representations that allow accurate and fast recognition and localisation. Further problems are caused by objects that are hard to recognise due to occlusion, small scale, or poor lighting and contrast, all of which pose significant problems as they reduce the image evidence of an object. Some object classes have “distractor objects” for which they are frequently mistaken (e.g. pedestrians and lamp posts). Also background clutter such as leaves or random objects can trigger undesired behaviour in the detector.

Deep learning has improved detector performance significantly. Previously dominant problems may lose significance if they are automatically solved by deep neural networks. Other, previously unimportant issues, can turn out to be new blocking points for better performance. It is important to regularly re-assess the failure modes of the state of the art to make well founded decisions about future research, but even more so after great improvements. The work reported in this thesis was done before and during the start of the success of deep learning and includes a lot of analyses of state-of-the-art detectors or aspects of the detector pipeline. Some of the work also aims to improve the current paradigm of detector training and enables end-to-end training.

1.2 Contributions

This thesis contributes to the entire detection pipeline either by analysis of detection systems or by improving upon established methods and providing novel insights. The first part of this thesis is specific to pedestrian detection, a subset of the object detection problem. The second part of the thesis addresses general object detection. Even though some of the work is evaluated on people detection, the insights are more general.

The first part of this thesis is the result of various collaborations. The details of the collaborations (lead authorship, credit of different contributions) are mentioned in beginning of the respective chapters. In the second part, Jan Hosang is the lead author and contributed all work.

1.2.1 Pedestrian detection

Historical analysis. We analyse the progress in pedestrian detection in 40 published methods over the course of the decade from 2004 to 2014 in chapter 3. We compare and analyse trends with respect to training data, test data, method families, classifiers, additional data and both spatial and temporal context. By combining different sources of information, we analyse complementarity. We present experiments that show that models with larger model capacity might be beneficial. Model transfer experiments between benchmarks show that some datasets are more suitable for training than others in order to generalise to a new dataset. We show that progress has been driven by designing and learning better features and experimentally emulate this progress by only changing features in a pedestrian detector. Deep learning as a means of joint learning of features and classifiers has since become a core ingredient of the strongest pedestrian detectors.

Failure analysis. Based on more recent detectors from 2015, we do a detailed failure analysis in chapter 5. We group most significant failure cases into categories, such as double detections, vertical structures, or other background. Using oracle experiments we predict how much detector performance can improve by rectifying all detection mistakes on background or pedestrians respectively. Insights from this work suggest future directions for pedestrian detection research.

Annotations. In chapter 5 we estimate human performance the Caltech pedestrian detection benchmark by providing additional human annotations. The annotators were provided with the same information as (monocular) pedestrian detectors, so this result serves as a lower bound of achievable performance and shows that there is significant room for improvement, even for state-of-the-art pedestrian detectors.

We sanitise the annotations using additional information and are able to show that these annotations are better localised than the standard annotations. The new annotations allow us to train better localised detectors and measure differences in localisation performance better than with previous annotations.

Deep learning. In chapter 4 we show that deep learning is a suitable means of learning not only for general object detectors, but also pedestrian detectors. Our approach uses standard convolutional neural networks, showing that it is not necessary to model aspects of the problem explicitly as believed previously, such as parts or occlusion. As a result our approach is simpler while outperforming previous neural networks by a large margin. This result holds true, even without pre-training the neural network on additional data.

1.2.2 Detection proposals

Chapter 6 contains an evaluation of detection proposals, an important component of today's object detection pipelines.

Systematic overview. We give an overview over 12 different detection proposal methods, which we categorise into bottom-up grouping and top-down window scoring. Proposal methods typically either approach the problem by grouping pixels or superpixels into objects or by ranking a large set of boxes. We discuss families of approaches, common ideas, and differences.

Repeatability. Using proposals during training of a detector changes the distribution of both positive and negative training examples compared to exhaustive search. In order for the detector to generalise well, training and test distributions should be similar. In particular proposal methods should show similar behaviour on similar images. We call this property repeatability and propose an evaluation method that measures how repeatable proposals are. We gradually perturb the image content with effects such as blur, rotation, or illumination and measure how much the set of proposals changes.

Recall. We perform a unified evaluation for all considered methods, measuring recall as a function of overlap criterion (between proposals and annotations) and number of proposals that each proposal method is allowed to return. This allows to compare methods over a wide range of operating points, depending on the desired application.

Class generalisation. Most considered methods have been tuned on the PASCAL VOC dataset, which consists of 20 object classes. To analyse bias of the proposal methods towards these specific 20 classes and to explore class generalisation, we show the recall evaluation also on the COCO and ImageNet detection benchmark, containing 80 and 200 classes respectively. We find that the considered proposals methods do generalise to unseen categories.

Detector performance. We show detector performance for several detectors on all considered proposal methods. This enables us to analyse which proposal evaluation measure is a good proxy for predicting detector performance. We propose average recall as a justified, new standard metric for comparing proposals.

1.2.3 Learning non-maximum suppression

Chapters 7 and 8 introduce novel models for learning non-maximum suppression, a typically hand-crafted, standard post-processing step of general object detectors. Standard non-maximum suppression is a structural weakness in the object detection pipeline, because it directly trades off precision and recall while being an integral part of current object detection paradigms.

Chapter 7 show that it is possible to improve over standard non-maximum suppression using the same information as standard non-maximum suppression. However, this approach is still takes standard non-maximum suppression decisions as an input. Chapter 8 overcomes this limitation and as such is the first neural network that is fully capable of performing non-maximum suppression. It also allows detection features as an input, which can come directly from the image or from a detector. The chapter also goes into greater detail on why standard non-maximum suppression was successful so far and which ingredients are crucial to learn non-maximum suppression, particularly in a neural network: (1) A loss that mirrors the detection evaluation and penalises double detections and (2) allowing neighbouring detections to "communicate".

While previous work on non-maximum suppression also proposed to learn this step, we are the first to enable end-to-end learning in a neural network. We cast the problem as a re-scoring task and show that the problem is solvable by a neural network that requires no post-processing. This work allows to merge detectors and non-maximum suppression, enabling joint training and true end-to-end learning.

1.3 Outline

This first part of this thesis (chapters 3–5) focuses on pedestrian detection, while the second part (chapters 5–8) contributes to general object detection.

Chapter 2: Related work. This chapter discusses related work on pedestrian detection, detection proposals, and non-maximum suppression. We contrast our research with previous as well as subsequent work in the field and establish historical context.

Large parts of the related work for detection proposals have been folded into the discussion of methods in chapter 6.

Chapter 3: Ten years of pedestrian detection. In this chapter we survey the progress of pedestrian detection between 2004 and 2014. We group methods into families of approaches and analyse ingredients that have improved performance. We combine successful ingredients and improve over previous pedestrian detectors. The chapter also highlights the importance of good features and feature learning.

Chapter 4: Deep learning for pedestrian detection. Motivated by the importance of feature learning for pedestrian detection, we explore deep learning for training a pedestrian detector. This chapter shows how vanilla large convolutional neural networks can be used for pedestrian detection. We use a strong pedestrian detector with high recall as a high quality detection proposal method. Oversampling frames in the training videos provides free additional training data. We show significant improvement over previous neural networks, small improvements over previous single frame detectors, and competitive performance to detectors that use additional information.

Chapter 5: Towards human performance pedestrian detection. This chapter establishes a human baseline as a landmark comparison for pedestrian detection performance. We analyse failure modes pedestrian detectors (during 2015) that suggest directions for future research and publish new, more accurate annotations that are crucial for driving pedestrian detection towards human level performance.

Chapter 6: What makes for effective detection proposals? In this chapter we survey object proposal methods for the purpose of general object detection. We compare different approaches, introduce evaluation metrics, and analyse their impact on final detection performance.

Chapter 7: A convnet for improving non-maximum suppression. This chapter is a first stab at learning non-maximum suppression for object detection in a convolutional neural network. We introduce a representation that encodes detection scores and overlap between neighbouring detections in a way that allows convolutions to pick up sufficient contextual information to outperform standard non-maximum suppression on very crowded scenes. The network rescores all detections, so that post-processing is no longer necessary.

Chapter 8: Learning non-maximum suppression. This chapter proposes an improvement over the method from chapter 7 that allows to operate on arbitrary feature vectors, instead of only detection scores and overlap specifically. This allows to stack the architecture to resolve complicated situations and gives the opportunity to incorporate image features. It no longer needs access to decisions from standard non-maximum suppression and performs well, even on current detection benchmarks with relatively few occlusion cases.

Chapter 9: Conclusions, insights, and future perspectives. This chapter contains concluding remarks of the thesis. We summarise insights and point out interesting future research directions.

The following work, originating from collaboration and unpublished work, is included into the thesis for completeness. Since it is not an integral part of the thesis, however, it is included as an appendix.

Appendix A: Weakly Supervised Segmentation. In this chapter we explore learning both instance and semantic segmentation from weak annotations. Starting from bounding box annotations we show how simple standard techniques can be used to not only outperform other weakly supervised semantic segmentation methods, but to even reach performance of methods with strong supervision. We show similar results for instance segmentation.

Appendix B: Distributed Shape. This chapter analyses detection of object classes that have been argued to be best represented by shape. We show that a standard detector (deformable parts model) is more competitive than previously reported and propose a new distributed shape detector, both on the ETHZ shape dataset. We analyse the complementarity between a “typical appearance detector” (deformable parts model) and our shape representation.

It's still magic, even if you know how it's done.

— Terry Pratchett, *A Hat Full of Sky*

2 Related work

Both pedestrian detection and general object detection have a long history of research. This chapter provides a historical context of recent work in these areas and discusses differences and similarities to the work presented in this thesis.

We first give background information on developments in general object detection in section 2.1, which influenced and inspired the work presented in this thesis. Section 2.2 goes into detail about works on pedestrian detection specifically. Section 2.3 considers work published after our analysis on proposals in chapter 6. In section 2.4, we give an overview of work published on non-maximum suppression and methods that are related to our approaches for tackling non-maximum suppression in chapters 7 and 8.

2.1 Recent developments in object detection

First we will give an overview over recent work on general object detection that influenced the entire field of object detection. These are not necessarily the best performing detectors, but the most related work to this thesis, including state-of-the-art detection systems.

Benchmarks. Typically these works were benchmarked on yearly versions of the PASCAL VOC dataset (Everingham *et al.*, 2014) between 2005–2012. The detection benchmark contains 20 object classes and about 12 000 annotated images. More recently the community started using PASCAL VOC for tuning methods and final results are presented on ImageNet or COCO. All these benchmarks contain challenging real-world images typically taken from flickr, resulting in a bias towards how people compose photos. The ImageNet detection benchmark (Russakovsky *et al.*, 2015) is significantly larger than PASCAL VOC: it contains 200 object classes in about 520 000 images. The COCO benchmark (Lin *et al.*, 2014) contains 80 classes and about 200 000 annotated images and focuses more on annotating rough segmentation masks instead of bounding boxes. COCO contains more small scale objects than PASCAL VOC and ImageNet and contains more objects per image on average.

Several other datasets are designed to benchmark the autonomous driving scenario (Dollár *et al.*, 2009b; Geiger *et al.*, 2012; Cordts *et al.*, 2016), i.e. they are acquired by filming from a driving car. This heavily biases the viewpoint from which objects are seen and which objects appear in the videos. Persons are usually either pedestrians (upright, walking) or cyclists. Other objects of interest are typically cars, traffic signs, and traffic lights. In contrast to this, the general object detection benchmarks mentioned before (PASCAL VOC, ImageNet, COCO) depict people in less constrained situations, poses, and often with heavy truncation. The set of classes and the appearance is more diverse.

Deformable parts model. The deformable parts model (DPM; Felzenszwalb *et al.*, 2010) was the de facto standard detector for a long time. It was not necessarily the winner of the yearly PASCAL VOC detection competition, but winning entries of both the classification and detection track of the competition often built on the DPM. Outside of the competition a lot of research has been exploring modifications and additions to the DPM. Our work on shape recognition in appendix B shows that it also works well on benchmarks that are not intended to be solvable by "appearance detectors".

The DPM is an extension of the work of Dalal and Triggs (2005), which consists of features that have been engineered to work well with a support vector machine (SVM) in a sliding window setting. The features are histograms of oriented gradients (HOG), extracted for a grid of small patches covering the entire image. The DPM uses HOG features with a latent SVM that models two important ingredients: components and parts. Instead of having one rigid template as in Dalal and Triggs (2005), the DPM has several deforming templates. The different components can capture different viewpoints or in general different appearance modes, as well as the very common left-right mirroring of objects. Objects in images are typically invariant to left-right mirroring (because gravity points down), which is modelled explicitly by latently estimating the orientation and normalizing it during training. Each component consists of one root template and several parts, that are attached to the root at anchor points but are allowed to move around for incurring some deformation cost.

Selective Search detector. One notable exception for winning the PASCAL VOC 2012 detection competition and not using the DPM is the Selective Search detector (van de Sande *et al.*, 2011). In the literature and also in the rest of this thesis, the name "Selective Search" is typically used to reference the proposal stage of the pipeline, but the paper describes an entire detector pipeline.

The Selective Search detector is neither a sliding window approach, nor does it use parts. Instead it groups superpixels into detection proposals, an over-complete set of image regions that likely contain all objects in the image. This is a alternative way of constructing the detector search space and we will have a closer look at proposals in chapter 6. The generated detection proposals are encoded using HOG and a spatial pyramid (of bags of words) of several different versions of the SIFT descriptor (Lowe, 2004). On this representation, they employ an SVM with a histogram intersection kernel, which is slower but feasible because of the reduced search space.

Deep learning: R-CNN. When Alexnet (Krizhevsky *et al.*, 2012) won the ImageNet classification challenge by a large margin it was unclear how to combine the insights of previous research on object detection with deep learning: How to add parts and components into a deep neural network such as the Alexnet? R-CNN (Girshick *et al.*, 2014) takes the simplest approach of treating the Alexnet just like any other feature+classifier combination. To score a potential object location, R-CNN crops the image content, resizes it to the input size of the Alexnet and feeds it into the convolutional neural network (convnet). Since this process is relatively slow, much too slow to be applied in a sliding window fashion, R-CNN adopts the idea of using detection proposals to reduce the search space. This approach boosted performance from 33.4% mAP on PASCAL VOC 2010 for DPM and 40.4% mAP for SegDPM (Fidler *et al.*, 2013) to 53.7% mAP with R-CNN.

ROI pooling: Fast R-CNN/Spatial Pyramid Pooling. Despite the search space reduction, both training and testing is still very slow, due to the required number of proposals. The idea of Spatial Pyramid Pooling (SPP, He *et al.*, 2014) is to share computation between neighbouring detection proposals by cropping and resizing a feature map instead of the image. They interpret the output of an intermediate convolutional layer in a neural network as a feature map, so the layers before this output can be seen as any other feature extractor for an entire image. The remaining problem is to resize the feature description of all detection proposals, which vary in size, to a fixed-size representation, so it can be used in a classifier. SPP uses a spatial pyramid pooling operation that was also frequently used in image classification (Grauman and Darrell, 2005), but instead of counting visual words (sum pooling over discrete words that all contribute one) SPP uses max pooling. This “ROI pooling” trick—in this context detection proposals are typically called “regions of interest” (ROI)—allows to resize arbitrarily sized feature representation into a fixed size representation to learn a multilayer neural network on top.

One limitation of SPP is that it does not jointly learn the feature extractor and the classifier part of the network. Fast R-CNN (Girshick, 2015) removes this limitation by running backpropagation (the standard learning procedure for neural networks using SGD and the chainrule) through the ROI pooling operation, allowing a faster version of R-CNN that also jointly learns features and the classifier.

Joint proposals and detection: Faster R-CNN. Ren *et al.* (2015) integrate detection proposal generation into the network. The convolutional feature map is an input to both the proposal generation stage and the ROI pooling. The proposal generation is implemented with one convolution to generate proposals of all sizes and aspect ratios, which can be viewed as a sliding window detector.

The original paper alternated between training the proposal stage and the detection stage, because the authors were concerned that detector learning is too hard while the proposals are still changing. Girshick successfully experimented with approximately joint training of proposals and detection in `py-faster-rcnn`[2], i.e. training both proposal and detection stage at the same time. It is called *approximately* joint, as the ROI pooling is not differentiable wrt. the proposal bounding box coordinates and thus the error of the detector stage is not backpropagated into the proposal stage.

Single stage: SSD, YOLO. Some recent research is dedicated to unifying the two-staged approach from Faster R-CNN into one stage, avoiding to resample features. SSD (Liu *et al.*, 2016) borrows the technique of “anchor detections” from Faster R-CNN’s proposal network and applies them to several feature maps of different resolution in the

[2] `https://github.com/rbgirshick/py-faster-rcnn`

convnet. This allows the detector to consider image regions of different sizes and at different resolutions for detection at varying scales, which can be seen as more traditional sliding window detection on feature maps of different scales as for example in the DPM.

All previous detectors are designed to be location invariant: they operate on some sort of local feature descriptor and predict offsets of anchor detections, so they are unaware of the detection's location in the image. YOLO (Redmon *et al.*, 2016) is not location invariant and takes a more global approach in the sense that the network is implicitly aware of the location of each detection in the image. The last layers of the network are fully connected layers that predict *all* detections on the entire image in the sense of mapping features of the entire image to all detections, as opposed to mapping features of a local region to detection on that local region.

Deep learning progress: AlexNet, VGG, Resnet, Inception. The task of image classification is typically the test bed for developing new convnets architectures. New architectures are typically quickly adopted in object detectors, leading to stronger detections or faster run-time. AlexNet (Krizhevsky *et al.*, 2012) evolved into the faster ZF network (Zeiler and Fergus, 2014) and the deeper, simpler VGG networks (Simonyan and Zisserman, 2015). The Inception models (Szegedy *et al.*, 2015a) present the idea of combining convolutions of different sizes. The Resnet architecture (He *et al.*, 2016a) introduces a technique to scale convnets ever deeper, all improving the speed-quality trade-off. So far, the research on better convnets was somewhat orthogonal to the research on how to employ them for object detection.

2.2 Pedestrian detection

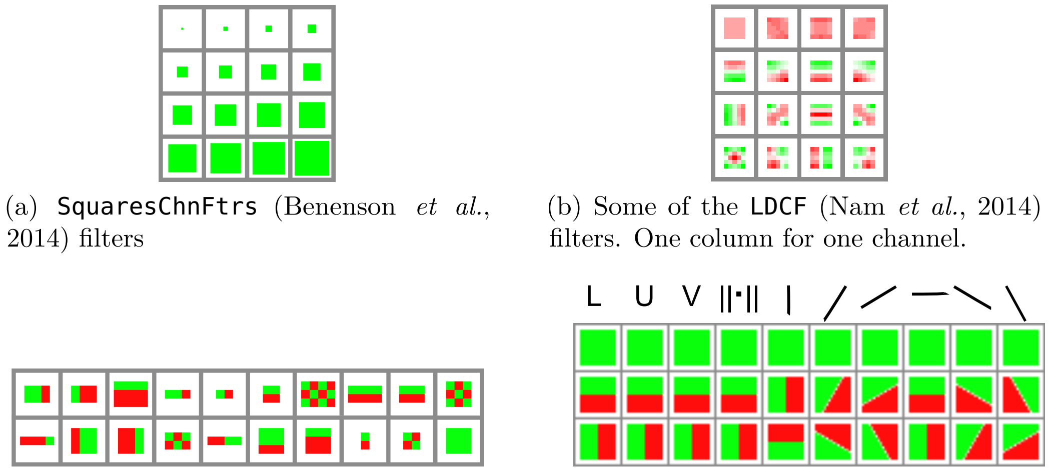
Pedestrians are of particular interest, for example for car safety, surveillance, or robotics. Although it is a sub problem of general object detection, pedestrians exhibit different statistics, which merits studying pedestrian detection separately.

2.2.1 Early deep learning for pedestrian detection (before 2014)

Despite the popularity of the task of pedestrian detection, only few works have applied deep neural networks to this task before 2015. Before the publication of the work in chapter 4, we are only aware of six.

Sermanet *et al.* (2013) focus on how to handle the limited training data (they use the INRIA dataset, which provides 614 positives and 1218 negative images for training). First, each layer is initialized using a form of convolutional sparse coding, and the entire network is subsequently fine-tuned for the detection task. They propose an architecture that uses features from the last and second last layer for detection.

A different line of work extends a deformable parts model (`DPM`, Felzenszwalb *et al.*, 2010) with a stack of Restricted Boltzmann Machines (RBMs) trained to reason about parts and occlusion (`DBN-Isol`, Ouyang and Wang, 2012). This model was extended to account for person-to-person relations (`DBN-Mut`, Ouyang and Wang, 2013b) and finally



(a) `SquaresChnFtrs` (Benenson *et al.*, 2014) filters

(b) Some of the `LDCF` (Nam *et al.*, 2014) filters. One column for one channel.

(c) Some examples of the 61 Checkerboards filters (Zhang *et al.*, 2015b)

(d) Illustration of Rotated filters (chapter 5) applied on each feature channel

Figure 2.1: Comparison of filters between some filtered channels detector variants. Green denotes +1 and red −1 filter values.

to jointly optimize all these aspects: `JointDeep` (Ouyang and Wang, 2013a) jointly optimizes features, parts deformations, occlusions, and person-to-person relations.

The `MultiSDP` network (Zeng *et al.*, 2013) feeds each layer with contextual features computed at different scales around the candidate pedestrian detection. Finally `SDN` (Luo *et al.*, 2014), the previously best performing convnet for pedestrian detection, uses additional "switchable layers" (RBM variants) to automatically learn both low-level features and high-level parts (e.g. "head", "legs").

Note that none of the papers rely on a "straightforward" convolutional network similar to the original LeNet (LeCun *et al.*, 1998) (layers of convolutions, non-linearities, pooling, inner products, and a softmax on top). Inspired by the success of R-CNN, we explore standard networks without problem specific design choices for pedestrian detection in chapter 4, where we find that detection with the AlexNet outperforms previous neural nets by a large margin.

2.2.2 ICF detectors for pedestrian detection

The integral channel feature detector (ICF) was proposed in Dollár *et al.* (2009a, 2014), which shows that HOG+LUV features (histograms of oriented gradients and the CIELUV color space) selected by boosted forests are very effective for pedestrian detection, surpassing previous detectors by a large margin. Following its success, many variants were proposed and showed significant improvement (Zhang *et al.*, 2014, 2015b; Nam *et al.*, 2014; Paisitkriangkrai *et al.*, 2014; Zhang *et al.*, 2015a). For instance, SquaresChannelFeatures (SCF, Benenson *et al.*, 2014) uses square averaging filters with different sizes; InformedHaar filters (Zhang *et al.*, 2014) are tailored to the pedestrian's up-right body shape; more generally, LDCF filters (Nam *et al.*, 2014) are top eigenvectors from linear discriminant analysis (LDA) on natural images; Checkerboard features (Zhang *et al.*, 2015b) is a naïve set of filters that consists of a uniform square, all horizontal and vertical gradient detectors (±1 values), and all possible checkerboard patterns for each size; RotatedFilters (chapter 5) is a simplified version of LDCF, inspired by its characteristical filter shapes. See the comparison of different filter types in figure 2.1.

2.2.3 Recent deep learning for pedestrian detection

In chapter 4, we show how to apply deep classification networks to pedestrian detection, analogous to R-CNN (Girshick *et al.*, 2014) for general object detection. Since that work, deep learning has become more successful and popular in pedestrian detection. Learning features directly from pixels instead of hand designing certain constrains on features as for ICF detectors seems to be the path to better pedestrian detection.

TA-CNN (Tian *et al.*, 2015b) additionally learns about both pedestrian and scene attributes to improve performance for pedestrian. Tian *et al.* (2015a) follow our approach from chapter 4 and extend it with weakly supervised part detectors to achieve robustness to occlusions and jittering of proposals to improve localisation. The method shows large improvements ($>$ 10% log-average MR on Caltech) both on the entire datasets and the occluded cases compared to previously best methods. For the purpose of both high quality and efficiency, Cai *et al.* (2015) combine hand-crafted features and CNN features into a large pool and design complexity-aware cascaded boosted trees to select inexpensive features at early stages, while pushing the more expensive ones to the later stages. The early stages of the cascade act effectively as a detection proposal method, generalizing the explicit detection proposal step of previous work. This yields large performance gains, comparable to Tian *et al.* (2015a) on Caltech, and strong performance on KITTI.

Explicit scale handling. All previous convnet detectors are scale agnostic in the sense that they either rescale image patches to the input size of the convnet or rescale image features to match the input size of a subsequent network. The following works handle scale explicitly, typically with similar techniques as in SSD (Liu *et al.*, 2016), which was published in parallel: they use lower convolutional feature maps with higher resolution for detecting smaller pedestrians.

Zhang *et al.* (2016a) analyse Faster R-CNN (Ren *et al.*, 2015) for pedestrian detection and find that the proposal network works well but the second stage of the detector actually degrades performance. They replace the second stage of Faster R-CNN with a boosted decision forest that operates on the concatenation of ROI pooled features from several convonlutional feature maps of the convnet. This means the convolutional features are only learned jointly for the first stage, but not for the second stage. Cai *et al.* (2016) also explore a two-stage architecture similar to Faster R-CNN with a multi-scale extension. They apply the proposal and detection network of Faster R-CNN on several convolutional feature maps of different resolution from the underlying convnet separately. Both approaches re-introduce bootstrapping into the training procedure, which was not used in convnet pedestrian detectors before. They both reach comparable performance on the Caltech pedestrian detection benchmark (10% log-average MR), but Cai *et al.* (2016) performs significantly better on KITTI pedestrians (74% AP vs. 61%).

Du *et al.* (2016) combine SSD (Liu *et al.*, 2016) as the proposal network with an ensemble of deep convnets to rescore detection proposals and a semantic segmentation network (Yu and Koltun, 2015). The predictions of all components—proposal score, segmentation mask, individual predictions from the ensemble—are merged by a hand-crafted soft-rejection scheme. This approach is the state of the art on the Caltech pedestrian detection benchmark.

This trend of better performance through gradual structural improvements of models to obtain better features for small pedestrians underlines our insight that features are a key factor for progress.

2.2.4 Analysis of pedestrian detection

In the last years, diverse efforts have been made to improve the performance of pedestrian detection. Most recent papers focus on novelty and better results, but neglect the analysis of the resulting system. Some analysis work can be found for general object detection in Agrawal *et al.* (2014) and Hoiem *et al.* (2012a): They analyse the effect of different pre-training strategies and characterise types of detector mistakes. In contrast, in the field of pedestrian detection, this kind of analysis is rarely done.

In 2008, Wojek and Schiele (2008) provided a failure analysis on the INRIA dataset, which is relatively small. In 2009, Enzweiler and Gavrila (2009) surveyed technologies used in different components of a pedestrian detection system and studied some popular systems at that time, including wavelet-based AdaBoost cascade (Viola *et al.*, 2005), HOG/linSVM (Dalal and Triggs, 2005), combined shape-texture detection (Gavrila and Munder, 2007), and others. In 2012, Dollár *et al.* (2012b) proposed a more challenging dataset and new evaluation metrics, which are more meaningful to real-world applications. The evaluation for sixteen detectors under various scenarios and for multiple datasets provides useful insights on challenges and research directions.

In chapter 3, we reviewe more than 40 methods covering a decade of research, quantifying the impact of different components on final detection quality. We conclude that improved features have been driving performance and are likely to continue doing so. We also show that optical flow (Park *et al.*, 2013) and context information (Ouyang and Wang, 2013b) are complementary to image features and can further boost detection accuracy.

In chapter 5, we analyse performance of more recent detectors more closely. The best method considered in the 2012 Caltech dataset survey (Dollár *et al.*, 2012b) had 10× more false positives at 20% recall than the methods considered here, and no method had reached the 95% mark. Since pedestrian detection has improved significantly in recent years, a deeper and more comprehensive analysis based on state-of-the-art detectors is valuable to provide better understanding as to where future efforts would best be invested.

2.3 Use cases for detection proposals

In chapter 6, we review class agnostic detection proposals in the context of object detection. We analyse bounding box proposals, since that is the information used in detectors such as Fast R-CNN and in the traditional single-image, fully supervised detection setting. We also focus on proposal methods that generalise beyond the training classes.

In chapter 6, we ask the question whether proposals bring something new to the table or whether they are merely a transition technology. Given some time since the research conducted in 2015, we answer this question by considering applications of proposals. In the following, we give an overview over use cases for proposals that have crystallised out since our analysis. We find this perspective important because proposals are no ends in themselves and new proposal research should rigorously optimise for a specific application. Without an application, detection proposals are just bad detectors.

Detection with full supervision. If we have a lot of training data with exactly the annotations that we want to predict at test time, there is no need for class generalisation. Proposals can be learned as part of a cascade (Faster R-CNN, Ren *et al.*, 2015) or skipped altogether without losing significant speed or quality (SSD, Liu *et al.*, 2016). If, instead of a bounding boxes, an instance mask is desired, it is possible to train a instance segmentation network (e.g. Pinheiro *et al.*, 2015, 2016; Dai *et al.*, 2016).

In this setting detection proposals seem to have been a transition technology to speed up detection. The lessons we take away from research in this direction of proposal generation focus on how to learn representations that can fast and accurately predict smaller search spaces for a cascade or segmentation masks from full supervision.

Tracking and segmenting moving object in videos. The task is to track moving objects in completely unlabelled videos. The class agnostic aspect of the localisation task effectively makes this task an “object tube” proposal method for videos. Since the problem is extended to the temporal domain, the search space expands by orders of magnitudes compared to the single frame case, which poses computational problems.

Recently methods for segmenting moving objects started to add a more high-level, object-centric information into the approach (as opposed to local, low level queues, such as optical flow). Perazzi *et al.* (2015) formulate an energy in a fully connected conditional random field (CRF) of object proposals. Fragkiadaki *et al.* (2015) use temporal information to generate single frame proposals and then grow them into “tubes” along the temporal dimension. Xiao and Jae Lee (2016) proceed similarly, but instead of segmentation proposals, they use bounding box proposals, which are refined to segmentations after they have been extended into “tubes”. All of these approaches are enabled by the search space reduction via proposals.

A different line of research does not require the speed-up through proposals but rather utilises the insights of how to detect image patches that are likely to contain objects. Luo and Kim (2013) track arbitrary objects in videos given annotations on the first frame. Tokmakov *et al.* (2016) train an image labeller on synthetic data and optical flow, that predicts for each pixel whether it belongs to a moving object. These predictions are fused with segmentation proposals and refined with a CRF. The explicit usage of proposals is starting to fade away for the top performaing methods on the DAVIS challenge (Perazzi *et al.*, 2016). Caelles *et al.* (2017) and Khoreva *et al.* (2017) train a network for segmenting objects in general and then finetune it to specific instances at test time. It seems that proposals are going to be integrated into the approaches more tightly and the distinction will disappear as for object detection.

Weakly supervised detection and instance segmentation. If proposals generalise beyond the object classes they have been trained on, this enables interesting variants of object detection with weaker supervision. Although generated object proposals are overcomplete and noisy in the sense that the first few proposals are typically either not well localised on objects or even cover background, there is useful signal in the statistics of proposals that can be used for weakly supervised training. Alexe *et al.* (2010) designed proposals and their scores specifically with weakly supervised object detection in mind: The same authors explored learning bounding box detectors from image level annotations in Deselaers *et al.* (2010) and later followed up by a wide range of works (Siva *et al.*, 2012; Prest *et al.*, 2012; Shapovalova *et al.*, 2012; Siva *et al.*, 2013; Pinheiro and Collobert, 2015; Bilen and Vedaldi, 2016; Qi *et al.*, 2016; Cinbis *et al.*, 2017). It is also possible to transfer bounding box or instance segmentation annotations between images (Guillaumin *et al.*, 2014; Tang *et al.*, 2014), use proposals for learning instance segmentation from bounding box annotations (appendix A; Dai *et al.*, 2015a), and even unsupervised object discovery (Cho *et al.*, 2015).

Contrary to the intuition that deep neural networks tend to be specific to the object classes they have been trained on, they can be designed to generalise well. For example Ghodrati *et al.* (2015) show good class generalisation even when training on only five classes.

In this setting proposals accomplish something remarkable: the means to bootstrap detection systems from weak or missing annotations. The objectness bias expressed in detection proposals that truly generalise beyond their training classes contributes information for these system to predict something they have no explicit supervision of. For this application, proposals seem to be more than a transition technology.

Others. With imagination it is possible to come up with countless other applications that revolve around objects, in particular when we are interested in objects of any class and supervision is scarce. For example Sun and Ling (2011) use proposals for retargeting images, i.e. they generate thumbnails of images that are more likely to contain objects. Another example is a metric for clutter (Russakovsky *et al.*, 2013) that was shown to correlate negatively with classification and localisation performance of several algorithms, so it may be used as a indicator for difficulty of images.

2.4 Non-maximum suppression

All previously mentioned detection works have in common that they use a hand crafted post-processing step. Almost all previous detectors since at least 1994 (Burel and Carel, 1994) uses some form of non-maximum suppression (NMS) that is supposed to merge all detections that cover the same object into one detection. Exceptions to this are Hough detectors and Stewart and Andriluka (2016). NMS is so common and so deeply embedded into how we think about detectors that typically papers do not even mention using NMS. The most common algorithm in recent detectors for performing NMS greedily accepts high scoring detections and rejects lower scoring detections that overlap more than a fixed threshold; we call it GreedyNMS in this thesis.

We seek neural networks that overcome the limitations of GreedyNMS. A neural network that is able to perform NMS without any post-processing is desirable, so we can combine it with a convnet detector and achieve proper end-to-end learning of detectors.

Clustering detections. The de facto standard algorithm, GreedyNMS, has survived several generations of detectors, from Viola and Jones (2004), over the deformable parts model (DPM, Felzenszwalb *et al.*, 2010), to the current state-of-the-art R-CNN family (Girshick *et al.*, 2014; Girshick, 2015; Ren *et al.*, 2015). Several other clustering algorithms have been explored for the task of NMS without showing consistent gains: mean-shift clustering (Dalal and Triggs, 2005; Wojek *et al.*, 2008), agglomerative clustering (Bourdev *et al.*, 2010), affinity propagation clustering (Mrowca *et al.*, 2015), determinantal point processes Lee *et al.* (2016), and heuristic variants (Sermanet *et al.*, 2014). Principled clustering formulations with globally optimal solutions have been proposed in Tang *et al.* (2015b) and Rothe *et al.* (2014), although they have yet to surpass performance of GreedyNMS. None of these methods enable end-to-end learning with the detector.

Linking detections to pixels. Hough voting establishes correspondences between detections and the image evidence supporting them, which can avoid overusing image content for several detections (Leibe *et al.*, 2008; Barinova *et al.*, 2012; Kontschieder *et al.*, 2012; Wohlhart *et al.*, 2012). Overall performance of hough voting detectors remains comparatively low. Yao *et al.* (2012) and Dai *et al.* (2015b) combine detections with semantic labelling, while Yan *et al.* (2015) rephrase detection as a labelling problem. Explaining detections in terms of image content is a sound formulation but these works rely on image segmentation and labelling, while our system operates purely on detections without additional sources of information or supervision.

Co-occurrence. One line of work proposes to detect pairs of objects instead of each individual object in order to handle strong occlusion (Sadeghi and Farhadi, 2011; Tang *et al.*, 2012; Ouyang and Wang, 2013b). It faces an even more complex NMS problem, since single and double detections need to be handled. Rodriguez *et al.* (2011) base suppression decisions on estimated crowd density. Our method does neither use image information nor is it hand-crafted to specifically detect pairs of objects.

Auto-context. Some methods improve object detection by jointly rescoring detections locally (Tu and Bai, 2010; Chen *et al.*, 2013) or globally (Vezhnevets and Ferrari, 2015) using image information. These approaches tend to produce fewer spread-out double detections and improve overall detection quality, but still require NMS. We also approach the problem of NMS as a rescoring task, but we will completely eliminate any post-processing.

Adaptive filtering. Zheng *et al.* (2015) formulate the mean-field approximation for solving fully connected CRFs (Krähenbühl and Koltun, 2011) as a recurrent neural network (RNN). This method operates on a dense grid like our approach in chapter 7, but which we avoid in chapter 8. Further the state between repeated mean-field iterations is a probability distributed over classes, while we build latent feature vectors for detections. Kiefel *et al.* (2014) and Jampani *et al.* (2016) use a sparse permutohedral lattice (Adams *et al.*, 2010) for efficiency, however the structure necessitates discretisation, which we would like to avoid. Overall these techniques are typically applied to locally propagate information to make neighbouring elements more similar, for example the task of denoising, which seeks to do edge preserving smoothing. The task of NMS is very different in the sense that the closer two detections are the more certain we are that they cover the same object and that they need to receive *different* scores.

Neural networks on graphs. A set of detections can be seen as a graph where overlapping windows are represented as edges in a graph of detections. Niepert *et al.* (2016) tackle the problem of building graph representations for either obtaining graph wide representations or per-node representations, that can be useful for several tasks. They use a vertex ordering to build a fixed-length sequence of vertices, collect fixed-size neighbourhoods for each vertex in the sequence, normalise the collected neighbourhood, and learn a neighbourhood representation based on the normalised neighbourhood. Most operations require node orderings, which are ill-defined in our case. Furthermore, we would like our approach to adapt to the density of the detection distribution in each image.

Set prediction with deep learning. On principle NMS is a function that maps a set of detections to another set of detections. Neural networks are successfully applied to sequences, but having a well defined order matters (Vinyals *et al.*, 2016) and it is unclear how to correctly order a sets of detections. Stewart and Andriluka (2016) successfully experiment with predicting detection sets from high to low detection confidence for small patches, but not for the entire image. Rezatofighi *et al.* (2016) formulate the problem using finite set statistics for the task of multi-label image classification. For NMS they use a subset of the network which predicts the number of detections, which they use to choose a GreedyNMS threshold, so the set prediction is not done by the network. They approach multi-label classification by factoring it into predicting a cardinality distribution and each scoring element in the output set individually. In chapter 8 we will argue that independent scoring cannot work for NMS. Vinyals *et al.* (2016) approach the problem, arguing that shuffling input elements should not change the resulting representation, by using an attention mechanism with an LSTM to build a fixed size representation in a permutation invariant manner. This is an interesting approach, but we suspect that a fixed-size representation for all detections in an image is not practicable, so this approach would have to be applied for *each* detection on an image.

End-to-end learning for detectors. Few works have explored true end-to-end learning while including NMS. One idea is to include GreedyNMS at training time (Wan *et al.*, 2015; Henderson and Ferrari, 2016), making the classifier aware of the NMS procedure at test time. This is conceptually more satisfying, but does not make NMS learnable. Another idea is to directly generate a sparse set of detections, so that NMS is unnecessary, which is done in Stewart and Andriluka (2016) by training an LSTM that generates detections on overlapping patches of the image. At the boundaries of neighbouring patches, objects might be predicted from both patches, so post-processing is still required.

NMS by rescoring. In chapter 7 and 8, we formulate the task of NMS as a rescoring problem and use a "matching loss" inspired by Stewart and Andriluka (2016) to train neural networks to perform NMS. These networks do not require any further post-processing. In chapter 7 we design a convnet that combines decisions of GreedyNMS with different overlap thresholds, allowing the network to choose the GreedyNMS operating point locally. All detections are assigned into a grid, so we can use convolutions for combining context information. Although the network has only access to the same information as GreedyNMS, we are able to achieve better performance.

None of the works mentioned in this section—including chapter 7—actually completely *remove* GreedyNMS from the final decision process that outputs a sparse set of detections. In chapter 8, we propose a network that is capable of performing NMS without being given a set of suppression alternatives to chose from and without assigning detections into a dense grid. The network operates purely on geometric information about detections, but builds up a rich feature representation, which allows to perform NMS. Instead of assigning detection to discrete image locations and operating over all image locations, we operate on a sparse set of detections directly. The network alternates between operating on pairs of detections and updating the representation of each detection individually and can be seen as a version of message passing.

Part I

PEDESTRIAN DETECTION

Pedestrian detection is a canonical instance of object detection. Because of its direct applications in car safety, surveillance, and robotics, it has attracted much attention. Importantly, it is a well defined problem with established benchmarks and evaluation metrics. As such, it has served as a playground to explore different ideas for object detection. The main paradigms for object detection — "Viola&Jones variants" (integral channels), HOG+SVM rigid templates, deformable part detectors (DPM), and convolutional neural networks (convnets) — have all been explored for this task.

This part starts with a retrospective in section 3, which analyses a decade of pedestrian detection before the deep learning breakthrough. Motivated by the importance of features and the success of convnets for general object detection we explore convnets for pedestrian detection in section 4. We experiment with different architectures and parameters, resulting in large performance gains over previous convnets. In section 5, we perform a detailed study of a state-of-the-art pedestrian detector. We show failure modes, provide a human baseline, introduce improved annotations, and suggest future directions of research.

3 Ten years of pedestrian detection

PAPER-BY-PAPER results make it easy to miss the forest for the trees. We analyse the remarkable progress of the decade from 2004–2014 by discussing the main ideas explored in the 40+ detectors present in the Caltech pedestrian detection benchmark. We observe that there exist three families of approaches, all reaching similar detection quality. Based on our analysis, we study the complementarity of the most promising ideas by combining multiple published strategies. This new decision forest detector achieves the best performance on the challenging Caltech dataset in July 2014.

This work has been published at the "Computer Vision for Road Scene Understanding and Autonomous Driving" workshop (Benenson *et al.*, 2014). Rodrigo Benenson was the lead author, Mohamed Omran conducted most of the experiments, and Jan Hosang contributed experiments incorporating context using `2Ped` in section 3.4.2, plots, writing, and analyses.

3.1 Introduction

The aim of this chapter is to review progress over a decade of pedestrian detection between 2004 and 2014 (40+ methods), identify the main ideas explored, and try to quantify which ideas had the most impact on final detection quality. In the next sections we review existing datasets (section 3.2), provide a discussion of the different approaches (section 3.3), and experiments reproducing/quantifying the recent years' progress (section 3.4, presenting experiments over ~ 20 newly trained detector models). Although we do not aim to introduce a novel technique, by putting together existing methods we report best detection results at the time of publication on the challenging Caltech dataset.

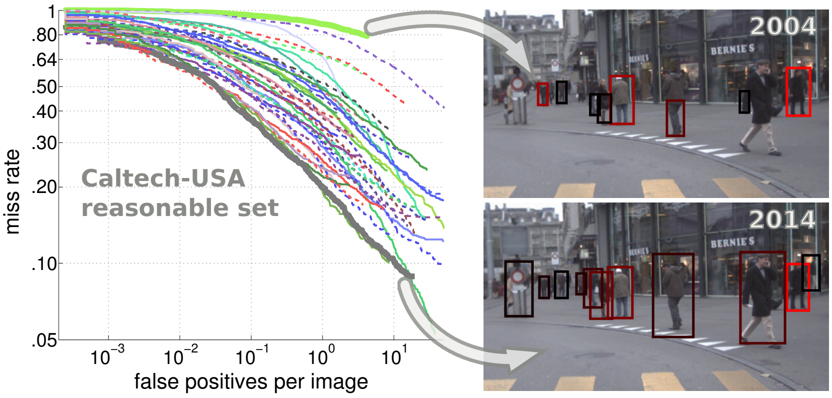
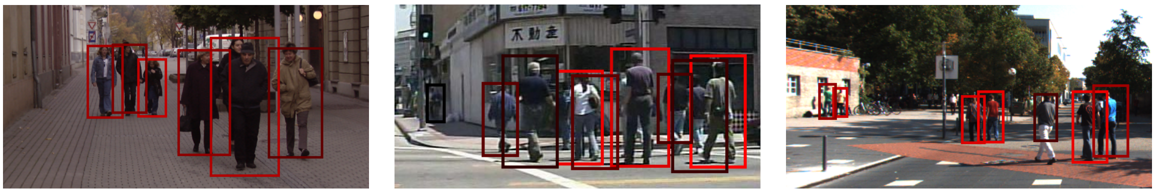


Figure 3.1: The last decade has shown tremendous progress on pedestrian detection. What have we learned out of the 40+ proposed methods?



(a) INRIA test set (b) Caltech test set (c) KITTI test set

Figure 3.2: Example detections of a top performing method (`SquaresChnFtrs`).

3.2 Datasets

Multiple public pedestrian datasets have been collected over the years; INRIA (Dalal and Triggs, 2005), ETH (Ess *et al.*, 2008), TUD-Brussels (Wojek *et al.*, 2009), Daimler (Enzweiler and Gavrila, 2009) (Daimler stereo (Keller *et al.*, 2009)), Caltech (Dollár *et al.*, 2009b), and KITTI (Geiger *et al.*, 2012) are the most commonly used ones. They all have different characteristics, weaknesses, and strengths.

INRIA is amongst the oldest and as such has comparatively few images. It benefits however from high quality annotations of pedestrians in diverse settings (city, beach, mountains, etc.), which is why it is commonly selected for training (see also section 3.4.4). ETH and TUD-Brussels are mid-sized video datasets. Daimler is not considered by all methods because it lacks colour channels. Daimler stereo, ETH, and KITTI provide stereo information. All datasets but INRIA are obtained from video, and thus enable the use of optical flow as an additional cue.

Today, Caltech and KITTI are the predominant benchmarks for pedestrian detection. Both are comparatively large and challenging. Caltech stands out for the large number of methods that have been evaluated side-by-side. KITTI stands out because its test set is slightly more diverse, but is not yet used as frequently. For a more detailed discussion of the datasets, see section 3.2.1, Dollár *et al.* (2012b), and Geiger *et al.* (2012). INRIA, ETH (monocular), TUD-Brussels, Daimler (monocular), and Caltech are available under a unified evaluation toolbox; KITTI uses its own separate one with unpublished test data. Both toolboxes maintain an online ranking where published methods can be compared side by side.

In this chapter we use primarily Caltech for comparing methods, INRIA and KITTI secondarily. See figure 3.2 for example images. Caltech and INRIA results are measured in log-average miss-rate (MR, lower is better), while KITTI uses area under the precision-recall curve (AUC, higher is better).

3.2.1 The Caltech pedestrian detection benchmark

The Caltech benchmark (Dollár *et al.*, 2012b) consists of 2.5 hours of 30Hz video recorded from a vehicle traversing the streets of Los Angeles, USA. The video annotations amount to a total of 350 000 bounding boxes covering ~2 300 unique pedestrians. Detection methods are evaluated on a test set consisting of 4 024 frames. The provided evaluation toolbox generates plots for different subsets of the test set based on annotation size, occlusion level and aspect ratio. The established procedure for training is to use every 30th video frame which results in a total of 4 250 frames with ∼1 600 pedestrian annotations.

More recently, methods which benefit from more training data have resorted to a finer sampling of the videos (chapter 4, 5; Nam *et al.*, 2014; Zhang *et al.*, 2015b), yielding 10× as much training data as the standard "1×" setting: ∼1 600 annotations on 42 782 frames. Here 10× and 1× refer to sampling every 3rd and 30th frame, respectively.

Typically, methods are evaluated in the so-called "reasonable" setting, which excludes particularly hard to detect pedestrians from the evaluation. This subset consists of pedestrians that are taller than 50 pixels and are occluded less than 35%.

3.2.2 Value of benchmarks

Individual papers usually only show a narrow view over the state of the art on a dataset. Having an official benchmark that collects detections from all methods greatly eases the author's effort to put their curve into context, and provides reviewers easy access to the state of the art results. The collection of results enable retrospective analyses such as the one presented in the next section.

3.3 Main approaches to improve pedestrian detection

Table 3.1 and figure 3.3 together provide a quantitative and qualitative overview over 40+ methods whose results are published on the Caltech pedestrian detection benchmark up until July 2014. Methods marked in italic are our newly trained models (described in section 3.4). We refer to all methods using their Caltech benchmark shorthand. Instead of discussing the methods' individual particularities, we identify the key aspects that distinguish each method (ticks of table 3.1) and group them accordingly. We discuss these aspects in the next subsections.

Brief chronology. In 2003, Viola *et al.* (2003) applied their `VJ` detector to the task of pedestrian detection. In 2005, Dalal and Triggs (2005) introduced the landmark `HOG` detector, which in 2008 served as a building block for the now classic deformable part model DPM (named `LatSvm` here) by Felzenszwalb *et al.* (2008). In 2009, the Caltech pedestrian detection benchmark was introduced by Dollár *et al.* (2009b), comparing seven pedestrian detectors. At this point in time, the evaluation metrics changed from per-window (FPPW) to per-image (FPPI), once the flaws of the per-window evaluation were identified (Dollár *et al.*, 2012b). Under this new evaluation metric some of the early detectors turned out to under-perform.

About one third of the methods considered here were published during 2013, reflecting a renewed interest on the problem. Similarly, half of the KITTI results for pedestrian detection were submitted in 2014.

Method	MR	Family	Features	Classifier	Context	Deep	Parts	M-Scales	More data	*Feat. type*	*Training*
VJ (Viola and Jones, 2004)	94.73%	DF	✓	✓						Haar	I
Shapelet (Sabzmeydani and Mori, 2007)	91.37%	-	✓							Gradients	I
PoseInv (Lin and Davis, 2008)	86.32%	-					✓			HOG	I+
LatSvm-V1 (Felzenszwalb *et al.*, 2008)	79.78%	DPM					✓			HOG	P
ConvNet (Sermanet *et al.*, 2013)	77.20%	DN				✓				Pixels	I
FtrMine (P. Dollár and Belongie, 2007)	74.42%	DF	✓							HOG+Color	I
HikSvm (Maji *et al.*, 2008)	73.39%	-		✓						HOG	I
HOG (Dalal and Triggs, 2005)	68.46%	-	✓	✓						HOG	I
MultiFtr (Wojek and Schiele, 2008)	68.26%	DF	✓	✓						HOG+Haar	I
HogLbp (Wang *et al.*, 2009)	67.77%	-	✓							HOG+LBP	I
AFS+Geo (Levi *et al.*, 2013)	66.76%	-			✓					Custom	I
AFS (Levi *et al.*, 2013)	65.38%	-								Custom	I
LatSvm-V2 (Felzenszwalb *et al.*, 2010)	63.26%	DPM		✓			✓			HOG	I
Pls (Schwartz *et al.*, 2009)	62.10%	-	✓	✓						Custom	I
MLS (Nam *et al.*, 2011)	61.03%	DF	✓							HOG	I
MultiFtr+CSS (Walk *et al.*, 2010)	60.89%	DF	✓							Many	T
FeatSynth (Bar-Hillel *et al.*, 2010)	60.16%	-	✓	✓						Custom	I
pAUCBoost (Paisitkriangkrai *et al.*, 2013)	59.66%	DF	✓	✓						HOG+COV	I
FPDW (Dollár *et al.*, 2010)	57.40%	DF								HOG+LUV	I
ChnFtrs (Dollár *et al.*, 2009a)	56.34%	DF	✓	✓						HOG+LUV	I
CrossTalk (Dollár *et al.*, 2012a)	53.88%	DF			✓					HOG+LUV	I
DBN−Isol (Ouyang and Wang, 2012)	53.14%	DN					✓			HOG	I
ACF (Dollár *et al.*, 2014)	51.36%	DF	✓							HOG+LUV	I
RandForest (Marin *et al.*, 2013)	51.17%	DF		✓						HOG+LBP	I&C
MultiFtr+Motion (Walk *et al.*, 2010)	50.88%	DF	✓						✓	Many+Flow	T
SquaresChnFtrs (Benenson *et al.*, 2013)	50.17%	DF	✓							HOG+LUV	I
Franken (Mathias *et al.*, 2013)	48.68%	DF		✓						HOG+LUV	I
MultiResC (Park *et al.*, 2010)	48.45%	DPM			✓		✓	✓		HOG	C
Roerei (Benenson *et al.*, 2013)	48.35%	DF	✓					✓		HOG+LUV	I
DBN−Mut (Ouyang *et al.*, 2013)	48.22%	DN			✓		✓			HOG	C
MF+Motion+2Ped (Ouyang and Wang, 2013b)	46.44%	DF			✓				✓	Many+Flow	I+
MOCO (Chen *et al.*, 2013)	45.53%	-	✓		✓					HOG+LBP	C
MultiSDP (Zeng *et al.*, 2013)	45.39%	DN	✓		✓	✓				HOG+CSS	C
ACF-Caltech (Dollár *et al.*, 2014)	44.22%	DF	✓							HOG+LUV	C
MultiResC+2Ped (Ouyang and Wang, 2013b)	43.42%	DPM			✓		✓	✓		HOG	C+
WordChannels (Costea and Nedevschi, 2014)	42.30%	DF	✓							Many	C
MT-DPM (Yan *et al.*, 2013)	40.54%	DPM					✓	✓		HOG	C
JointDeep (Ouyang and Wang, 2013a)	39.32%	DN			✓					Color+Gradient	C
SDN (Luo *et al.*, 2014)	37.87%	DN				✓	✓			Pixels	C
MT-DPM+Context (Yan *et al.*, 2013)	37.64%	DPM			✓		✓	✓		HOG	C+
ACF+SDt (Park *et al.*, 2013)	37.34%	DF	✓						✓	ACF+Flow	C+
SquaresChnFtrs (Benenson *et al.*, 2013)	34.81%	DF	✓							HOG+LUV	C
InformedHaar (Zhang *et al.*, 2014)	34.60%	DF	✓							HOG+LUV	C
Katamari-v1	22.49%	DF	✓		✓				✓	HOG+Flow	C+

Table 3.1: Listing of methods considered on Caltech, sorted by log-average miss-rate (lower is better). Consult sections 3.3.1 to 3.3.9 for details of each column. See also matching figure 3.3. “HOG” indicates HOG-like (Dalal and Triggs, 2005).

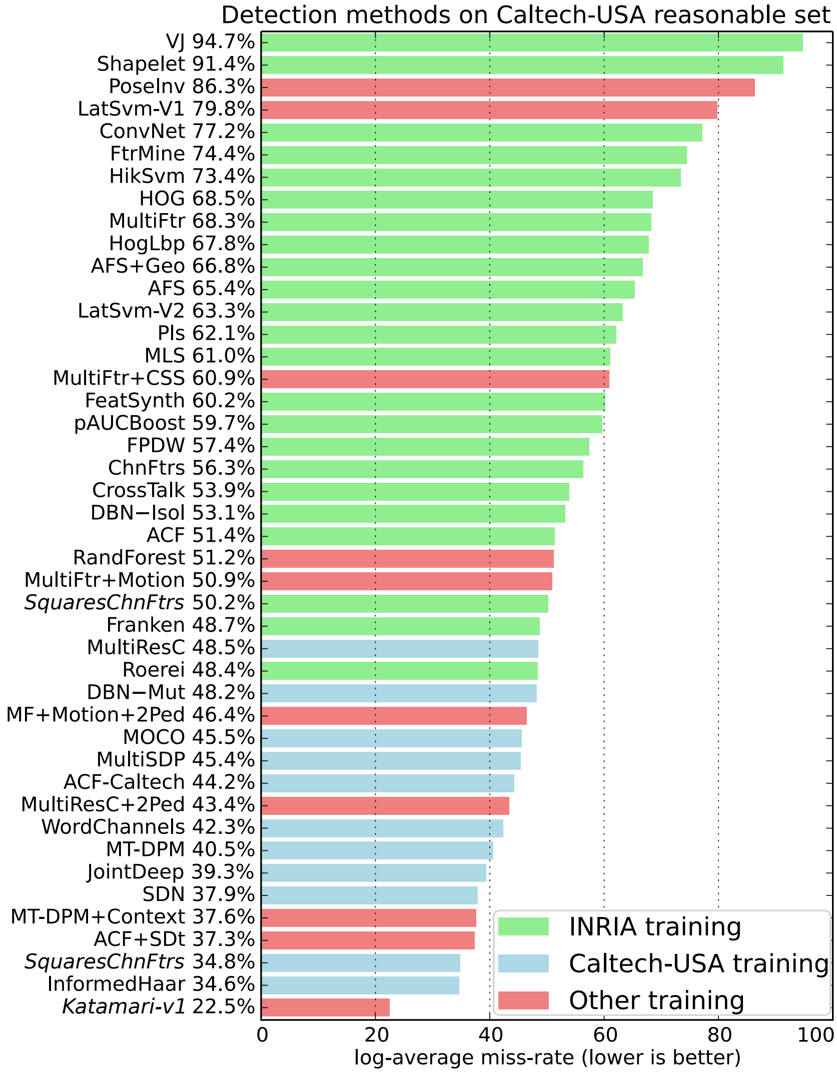


Figure 3.3: Caltech detection results.

3.3.1 Training data

Figure 3.3 shows that differences in detection performance are, not surprisingly, dominated by the choice of training data. Methods trained on Caltech systematically perform better than methods that generalise from INRIA. Table 3.1 gives additional details on the training data used[3]. High performing methods with "other training" use extended versions of Caltech. For instance `MultiResC+2Ped` uses Caltech plus an extended set of annotations over INRIA, `MT-DPM+Context` uses an external training set for cars, and `ACF+SDt` employs additional frames from the original Caltech videos.

[3] "Training" data column: I→INRIA, C→Caltech, I+/C+ →INRIA/Caltech and additional data, P→Pascal, T→TUD-Motion, I&C→both INRIA and Caltech.

3.3.2 Solution families

Overall we notice that out of the 40+ methods we can discern three families: 1) DPM variants (`MultiResC` (Park *et al.*, 2010), `MT-DPM` (Yan *et al.*, 2013), etc.), 2) Deep networks (`JointDeep` (Ouyang and Wang, 2013a), `ConvNet` (Sermanet *et al.*, 2013), etc.), and 3) Decision forests (`ChnFtrs`, `Roerei` (Benenson *et al.*, 2013), etc.). In table 3.1 we identify these families as `DPM`, `DN`, and `DF` respectively.

Based on raw numbers alone boosted decision trees (`DF`) seem particularly suited for pedestrian detection, reaching top performance on both the "train on INRIA, test on Caltech", and "train on Caltech, test on Caltech" tasks. It is unclear, however, what gives them an edge. The deep networks explored also show interesting properties and fast progress in detection quality.

Conclusion. Overall, at present, DPM variants, deep networks, and (boosted) decision forests all reach top performance in pedestrian detection (around 37 % MR on Caltech, see figure 3.3).

3.3.3 Better classifiers

Since the original proposal of `HOG+SVM` (Dalal and Triggs, 2005), linear and non-linear kernels have been considered. `HikSvm` (Maji *et al.*, 2008) considered fast approximations of non-linear kernels. This method obtains improvements when using the flawed FPPW evaluation metric (see section 3.3), but fails to perform well under the proper evaluation (FPPI). In the work on `MultiFtrs` (Wojek and Schiele, 2008), it was argued that, given enough features, Adaboost and linear SVM perform roughly the same for pedestrian detection.

Recently, more and more components of the detector are optimized jointly with the "decision component" (e.g. pooling regions in `ChnFtrs` (Dollár *et al.*, 2009a), filters in `JointDeep` (Ouyang and Wang, 2013a)). As a result the distinction between features and classifiers is not clear-cut anymore (see also sections 3.3.8 and 3.3.9).

Conclusion. There is no conclusive empirical evidence indicating whether non-linear kernels provide meaningful gains over linear kernels (for pedestrian detection, when using non-trivial features). Similarly, it is unclear whether one particular type of classifier (e.g. SVM or decision forests) is better suited for pedestrian detection than another.

3.3.4 Additional data

The core problem of pedestrian detection focuses on individual monocular colour image frames. Some methods explore leveraging additional information at training and test time to improve detections. They consider stereo images (Keller *et al.*, 2011), optical flow (using previous frames, e.g. `MultiFtr+Motion` (Walk *et al.*, 2010) and `ACF+SDt` (Park *et al.*, 2013)), tracking (Ess *et al.*, 2009), or data from other sensors (such as lidar (Premebida *et al.*, 2014) or radar).

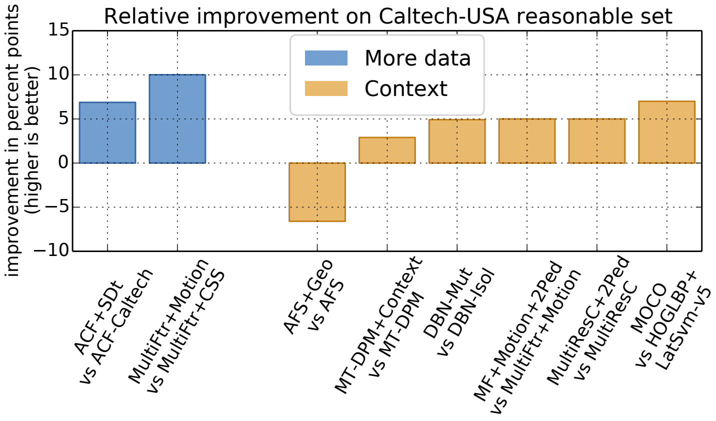


Figure 3.4: Caltech detection improvements for different method types. Improvement relative to each method's relevant baseline ("method vs baseline").

For monocular methods it is still unclear how much tracking can improve per-frame detection itself. As seen in figure 3.4 exploiting optical flow provides a non-trivial improvement over the baselines. Curiously, the top results at the time of publication of this work (`ACF-SDt`, Park *et al.*, 2013) are obtained using coarse rather than high quality flow. In section 3.4.2 we inspect the complementarity of flow with other ingredients. Good success exploiting flow and stereo on the Daimler dataset has been reported (Enzweiler and Gavrila, 2011), but similar results have yet to be seen on newer datasets such as KITTI.

Conclusion. At the time of publication of this study (Benenson *et al.*, 2014), using additional data provides meaningful improvements, albeit on modern dataset stereo and flow cues have yet to be fully exploited. Methods based merely on single monocular image frames have been able to keep up with the performance improvement introduced by additional information.

3.3.5 Exploiting context

Sliding window detectors score potential detection windows using the content inside that window. Drawing on the context of the detection window, i.e. image content surrounding the window, can improve detection performance. Strategies for exploiting context include: ground plane constraints (`MultiResC` (Park *et al.*, 2010), `RandForest` (Marin *et al.*, 2013)), variants of auto-context (Tu and Bai, 2010) (`MOCO` (Chen *et al.*, 2013)), other category detectors (`MT-DPM+Context` (Yan *et al.*, 2013)), and person-to-person patterns (`DBN−Mut` (Ouyang *et al.*, 2013), `+2Ped` (Ouyang and Wang, 2013b), `JointDeep` (Ouyang and Wang, 2013a)).

Figure 3.4 shows the performance improvement for methods incorporating context. Overall, we see improvements of $3 \sim 7$ MR percent points. (The negative impact of `AFS+Geo` is due to a change in evaluation, see section 3.3.) Interestingly, `+2Ped` (Ouyang and Wang, 2013b) obtains a consistent 2 $\sim$ 5 MR percent point improvement over existing methods, even top performing ones (see section 3.4.2).

Conclusion. Context provides consistent improvements for pedestrian detection, although the amount of improvement is lower compared to additional test data (section 3.3.4) and deep architectures (section 3.3.8). The bulk of detection quality must come from other sources.

3.3.6 Deformable parts

The DPM detector (Felzenszwalb *et al.*, 2010) was originally motivated for pedestrian detection. It is an idea that has become very popular and dozens of variants have been explored.

For pedestrian detection the results are competitive, but not salient (`LatSvm` (Yan *et al.*, 2014; Felzenszwalb *et al.*, 2008), `MultiResC` (Park *et al.*, 2010), `MT-DPM` (Yan *et al.*, 2013)). More interesting results have been obtained when modelling parts and their deformations inside a deep architecture (e.g. `DBN−Mut` (Ouyang *et al.*, 2013), `JointDeep` (Ouyang and Wang, 2013a)).

`DPM` and its variants are systematically outmatched by methods using a single component and no parts (`Roerei` (Benenson *et al.*, 2013), `SquaresChnFtrs` see section 3.4.1, casting doubt on the need for parts. Recent work has explored ways to capture deformations entirely without parts (Hariharan *et al.*, 2014b; Pedersoli *et al.*, 2014).

Conclusion. For pedestrian detection there is still no clear evidence for the necessity of components and parts, beyond the case of occlusion handling.

3.3.7 Multi-scale models

Typically for detection, both high and low resolution candidate windows are resampled to a common size before extracting features. It has recently been noticed that training different models for different resolutions systematically improve performance by 1 $\sim$ 2 MR percent points (Park *et al.*, 2010; Benenson *et al.*, 2013; Yan *et al.*, 2013), since the detector has access to the full information available at each window size. This technique does not impact computational cost at detection time (Benenson *et al.*, 2012), although training time increases.

Conclusion. Multi-scale models provide a simple and generic extension to existing detectors. Despite consistent improvements, their contribution to the final quality is rather minor.

3.3.8 Deep architectures

Large amounts of training data and increased computing power have lead to recent successes of deep architectures (typically convolutional neural networks) on diverse computer vision tasks (large scale classification and detection (Girshick *et al.*, 2014; Sermanet *et al.*, 2014), semantic labelling (Pinheiro and Collobert, 2014)). These results have inspired the application of deep architectures to the pedestrian task.

`ConvNet` (Sermanet *et al.*, 2013) uses a mix of unsupervised and supervised training to create a convolutional neural network trained on INRIA. This method obtains fair results on INRIA, ETH, and TUD-Brussels, however fails to generalise to the Caltech setup. This method learns to extract features directly from raw pixel values.

Another line of work focuses on using deep architectures to jointly model parts and occlusions (`DBN−Isol` (Ouyang and Wang, 2012), `DBN−Mut` (Ouyang *et al.*, 2013), `JointDeep` (Ouyang and Wang, 2013a), and `SDN` (Luo *et al.*, 2014)). The performance improvement such integration varies between 1.5 to 14 MR percent points. Note that these works use edge and colour features (Ouyang and Wang, 2013a; Ouyang *et al.*, 2013; Ouyang and Wang, 2012), or initialise network weights to edge-sensitive filters, rather than discovering features from raw pixel values as usually done in deep architectures. No results have yet been reported using features pre-trained on ImageNet, as in Girshick *et al.* (2014) and Azizpour *et al.* (2015).

Conclusion. At the time of this study, there was no clear evidence that deep networks are good at learning features for pedestrian detection. Most successful methods at this time use neural networks to model higher level aspects of parts, occlusions, and context. The obtained results are on par with DPM and decision forest approaches.

As discussed in section 2.2.3, this is not the case any more. Neural networks closed the gap to other approaches and now show top performance.

3.3.9 Better features

The most popular approach (about 30 % of the considered methods) for improving detection quality is to increase/diversify the features computed over the input image. By having richer and higher dimensional representations, the classification task becomes somewhat easier, enabling improved results. A large set of feature types have been explored: edge information (Dalal and Triggs, 2005; Dollár *et al.*, 2009a; Lim *et al.*, 2013; Luo *et al.*, 2014), colour information (Dollár *et al.*, 2009a; Walk *et al.*, 2010), texture information (Wang *et al.*, 2009), local shape information (Costea and Nedevschi, 2014), covariance features (Paisitkriangkrai *et al.*, 2013), amongst others. More and more diverse features have been shown to systematically improve performance.

While various decision forest methods use 10 feature channels (`ChnFtrs`, `ACF`, `Roerei`, `SquaresChnFtrs`, etc.), some papers have considered up to an order of magnitude more channels (Wojek and Schiele, 2008; Lim *et al.*, 2013; Paisitkriangkrai *et al.*, 2013; Marin *et al.*, 2013; Costea and Nedevschi, 2014). Despite the improvements by adding many channels, top performance is still reached with only 10 channels (6 gradient orientations, 1 gradient magnitude, and 3 colour channels, we name these HOG+LUV); see table 3.1 and figure 3.3. In section 3.4.1 we study different feature combinations in more detail.

From VJ (95% MR) to ChnFtrs (56.34% MR, by adding HOG and LUV channels), to SquaresChnFtrs-Inria (50.17% MR, by exhaustive search over pooling sizes, see section 3.4), improved features drive progress. Switching training sets (section 3.3.1) enables SquaresChnFtrs-Caltech to reach state of the art performance on the Caltech dataset; improving over significantly more sophisticated methods. InformedHaar (Zhang *et al.*, 2014) obtains top results by using a set of Haar-like features manually designed for the pedestrian detection task. In contrast SquaresChnFtrs-Caltech obtains similar results without using such hand-crafted features and being data driven instead.

More recent studies show that using more and better features yields further improvements (Paisitkriangkrai *et al.*, 2014; Nam *et al.*, 2014). It should be noted that better features for pedestrian detection have not yet been obtained via deep learning approaches (see caveat on ImageNet features in section 3.3.8).

Conclusion. In the last decade improved features have been a constant driver for detection quality improvement, and it seems that it will remain so in the years to come. Most of this improvement has been obtained by extensive trial and error. The next scientific step will be to develop a more profound understanding of the what makes good features good, and how to design even better ones[4].

3.4 Experiments

Based on our analysis in the previous section, three aspects seem to be the most promising in terms of impact on detection quality: better features (section 3.3.9), additional data (section 3.3.4), and context information (section 3.3.5). We thus conduct experiments on the complementarity of these aspects.

Among the three solution families discussed (section 3.3.2), we choose the Integral Channels Features framework (Dollár *et al.*, 2009a) (a decision forest) for conducting our experiments. Methods from this family have shown good performance, train in minutes∼hours, and lend themselves to the analyses we aim.

In particular, we use the (open source) SquaresChnFtrs baseline described in (Benenson *et al.*, 2013): 2048 level-2 decision trees (3 threshold comparisons per tree) over HOG+LUV channels (10 channels), composing one 64×128 pixels template learned via vanilla AdaBoost and few bootstrapping rounds of hard negative mining.

3.4.1 Reviewing the effect of features

In this section, we evaluate the impact of increasing feature complexity. We tune all methods on the INRIA test set, and demonstrate results on the Caltech test set (see figure 3.5).

[4] This insight echoes with the current success of deep learning, too.

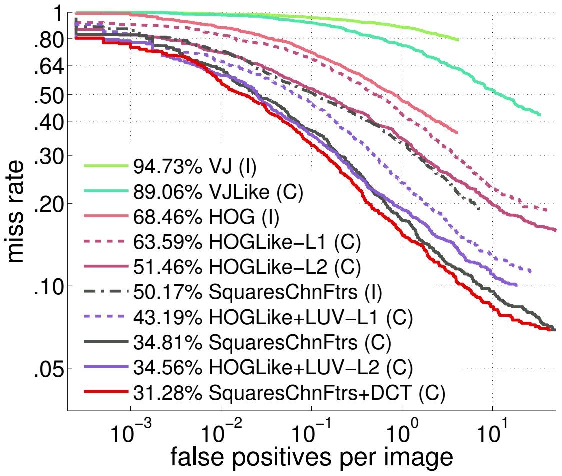


Figure 3.5: Effect of features on detection performance. Caltech reasonable test set.

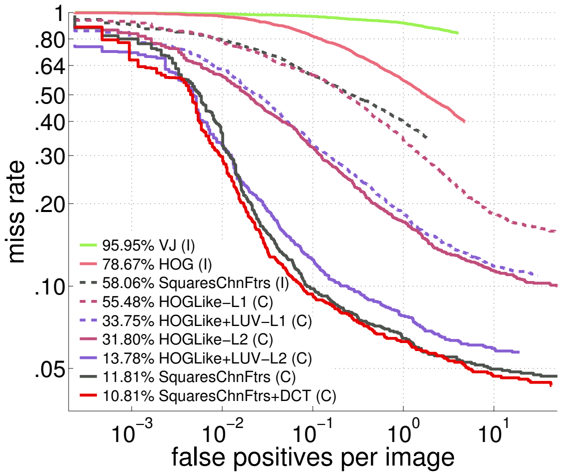


Figure 3.6: Caltech training set performance. (I)/(C) indicates using INRIA/Caltech training set.

The first series of experiments aims at mimicking landmark detection techniques, such as VJ (Viola *et al.*, 2003), HOG+linear SVM (Dalal and Triggs, 2005), and ChnFtrs (Dollár *et al.*, 2009a). VJLike uses only the luminance colour channel, emulating the Haar wavelet like features from the original (Viola *et al.*, 2003) using level 2 decision trees. HOGLike-L1/L2 use 8 × 8 pixel pooling regions, 1 gradient magnitude and 6 oriented gradient channels, as well as level 1/2 decision trees. We also report results when adding the LUV colour channels HOGLike+LUV (10 feature channels total). SquaresChnFtrs is the baseline described in the beginning of section 3.4, which is similar to HOGLike+LUV to but with square pooling regions of any size.

Inspired by Nam *et al.* (2014), we also expand the 10 HOG+LUV channels into 40 channels by convolving each channel with three DCT (discrete cosine transform) basis functions (of 7 × 7 pixels), and storing the absolute value of the filter responses as additional feature channels. We name this variant SquaresChnFtrs+DCT.

Conclusion. Much of the progress since VJ can by explained by the use of better features, based on oriented gradients and colour information. Simple tweaks to these well known features (e.g. projection onto the DCT basis) can still yield noticeable improvements.

3.4.2 Complementarity of approaches

After revisiting the effect of single frame features in section 3.4.1 we now consider the complementary of better features (HOG+LUV+DCT), additional data (via optical flow), and context (via person-to-person interactions).

We encode the optical flow using the same SDt features from ACF+SDt (Park *et al.*, 2010) (image difference between current frame T and coarsely aligned T-4 and T-8). The context information is injected using the +2Ped re-weighting strategy (Ouyang and

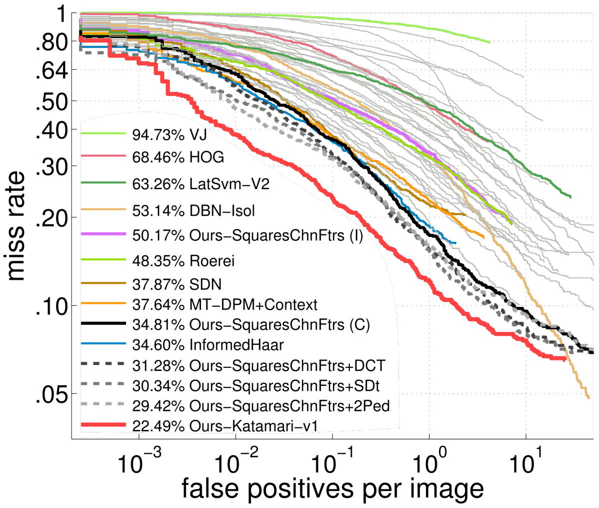


Figure 3.7: Some of the top quality detection methods for Caltech. See section 3.4.2.

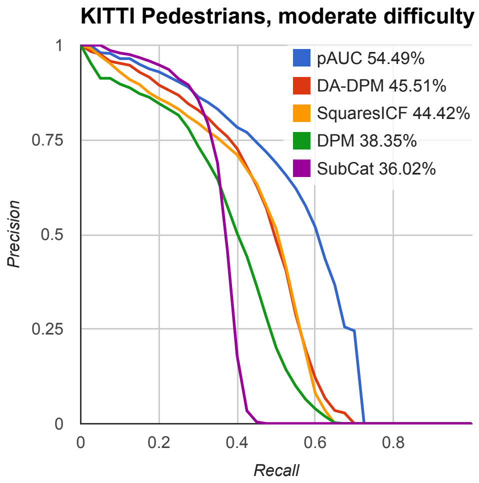


Figure 3.8: Pedestrian detection results on the KITTI dataset.

Wang, 2013b) (the detection scores are combined with the scores of a "2 person" DPM detector). In all experiments both DCT and SDt features are pooled over 8 × 8 regions (as in Park *et al.* (2010)), instead of "all square sizes" for the HOG+LUV features.

The combination `SquaresChnFtrs+DCT+SDt+2Ped` is called `Katamari-v1`. It reaches the best performance on the Caltech dataset in 2014. In figure 3.7 we show it together with the best performing method for each training set and solution family at the time (see table 3.1).

Conclusion. Our experiments show that adding extra features, flow, and context information are largely complementary (12 % gain, instead of $3 + 7 + 5$ %), even when starting from a strong detector.

It remains to be seen if future progress in detection quality will be obtained by further insights of the "core" algorithm (thus further diminishing the relative improvement of add-ons), or by extending the diversity of techniques employed inside a system.

3.4.3 How much model capacity is needed?

The main task of detection is to generalise from training to test set. Before we analyse the generalisation capability (section 3.4.4), we consider a necessary condition for high quality detection: is the learned model performing well on the training set?

In figure 3.6 we see the detection quality of the models considered in section 3.4.1, when evaluated over their training set. None of these methods performs perfectly on the training set. In fact, the trend is very similar to performance on the test set (see figure 3.5) and we do not observe yet symptoms of over-fitting.

Training set Test set	INRIA	Caltech	KITTI
INRIA	*17.42* %	60.50 %	**55.83** %
Caltech	**50.17** %	*34.81* %	61.19 %
KITTI	**38.61** %	28.65 %	*44.42* %
ETH	**56.27** %	76.11%	61.19 %

Table 3.2: Effect of training set on the detection quality on different test sets. Bold indicates second best training set for each test set, except for ETH where bold indicates the best training set.

Conclusion. Our results indicate that research on increasing the discriminative power of detectors is likely to further improve detection quality. More discriminative power can originate from more and better features or more complex classifiers.

3.4.4 Generalisation across datasets

For real world application beyond a specific benchmark, the generalisation capability of a model is key. In that sense results of models trained on INRIA and tested on Caltech are more relevant than the ones trained (and tested) on Caltech.

Table 3.2 shows the performance of `SquaresChnFtrs` over Caltech when using different training sets (MR for INRIA/Caltech/ETH, AUC for KITTI). These experiments indicate that training on Caltech or KITTI provides little generalisation capability towards INRIA, while the converse is not true. Surprisingly, despite the visual similarity between KITTI and Caltech, INRIA is the second best training set choice for KITTI and Caltech. This shows that Caltech pedestrians are of “their own kind”, and that the INRIA dataset is effective due to its diversity. In other words, a training set containing few diverse pedestrians (INRIA) is better than many similar ones (Caltech/KITTI).

The good news is that the best methods considered here seem to perform well both across datasets and when trained on the respective training data. Figure 3.8 shows methods trained and tested on KITTI, we see that `SquaresChnFtrs` (named `SquaresICF` in KITTI) is better than vanilla DPM and on par with the best DPM variant. The best method on KITTI as in July 2014, `pAUC` (Paisitkriangkrai *et al.*, 2014), is a variant of `ChnFtrs` using 250 feature channels (see the KITTI website for details on the methods). These two observations are consistent with our discussions in sections 3.3.9 and 3.4.1.

Conclusion. While detectors learned on one dataset may not necessarily transfer well to others, their ranking is stable across datasets, suggesting that insights can be learned from well-performing methods regardless of the benchmark.

3.5 Conclusion

Our experiments show that most of the progress in the last decade of pedestrian detection can be attributed to the improvement in features alone. Evidence suggests that this trend will continue. Although some of these features might be driven by learning, they are mainly hand-crafted via trial and error.

Our experiment combining the detector ingredients that our retrospective analysis found to work well (better features, optical flow, and context) shows that these ingredients are mostly complementary. Their combination produces best published detection performance on Caltech in July 2014.

While the three big families of pedestrian detectors (deformable part models, decision forests, deep networks) are based on different learning techniques, their state-of-the-art results are surprisingly close.

The main challenge ahead seems to develop a deeper understanding of what makes good features good, so as to enable the design of even better ones.

4 Deep learning for pedestrian detection

In chapter 3 we saw that the progress of a decade of research on pedestrian detection has been driven by feature engineering. Feature learning has been explored at the time, but only to some extend: the strongest detectors learn features, but only on top of hand crafted "feature channels".

In this chapter we study the use of convolutional neural networks (convnets) as an extreme point of feature learning that starts directly from the pixel grid. Despite their recent diverse successes, convnets historically underperform compared to other pedestrian detectors. We deliberately omit explicitly modelling the problem into the network (e.g. parts or occlusion modelling) and show that we can reach competitive performance without bells and whistles. In a wide range of experiments we analyse differently sized convnets, their architectural choices, parameters, and the influence of different training data, including pre-training on surrogate tasks.

This work was published at CVPR (Hosang *et al.*, 2015). Jan Hosang was the lead author and Mohamed Omran contributed experiments on smaller networks. We present the best convnet detector on the Caltech and KITTI dataset at the time, improving over all previous convnets both for the Caltech1x and Caltech10x training setup. Using additional data at training time, our strongest convnet model is competitive even to previous detectors that use additional data (optical flow) at test time.

4.1 Introduction

In recent years the field of computer vision has seen an explosion of success stories involving convolutional neural networks (convnets). Such architectures currently provide top results for general object classification (Krizhevsky *et al.*, 2012; Russakovsky *et al.*, 2015), general object detection (Szegedy *et al.*, 2015a), feature matching (Fischer *et al.*, 2014), stereo matching (Zbontar and LeCun, 2015), scene recognition (Zhou *et al.*, 2014; Chen *et al.*, 2014), pose estimation (Tompson *et al.*, 2014; Chen and Yuille, 2014), action recognition (Karpathy *et al.*, 2014; Simonyan and Zisserman, 2014) and many other tasks (Razavian *et al.*, 2014; Azizpour *et al.*, 2015). We would like to know if the success of convnets is transferable to the pedestrian detection task.

Previous work on neural networks for pedestrian detection has relied on special-purpose designs, e.g. hand-crafted features, part and occlusion modelling. Although these proposed methods perform reasonably, previous top methods are all based on decision trees learned via Adaboost (e.g. Benenson *et al.*, 2014; Zhang *et al.*, 2014; Paisitkriangkrai *et al.*, 2014; Nam *et al.*, 2014; Wang *et al.*, 2013). In this work we revisit the question, and show that both small and large vanilla convnets can reach top performance on the challenging Caltech pedestrians dataset. We provide extensive

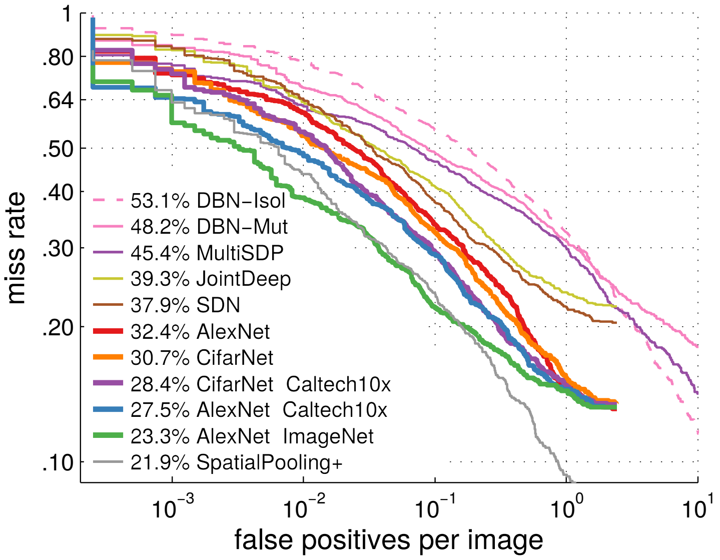


Figure 4.1: Comparison of convnet methods on the Caltech test set (see section 4.7). At the time of publication of this chapter, our CifarNet and AlexNet results significantly improve over previous convnets, and matches the best reported results at that time (`SpatialPooling+`, which additionally uses optical flow).

experiments regarding the details of training, network parameters, and different proposal methods.

Object detection. Other than pedestrian detection, related convnets have been used for detection of ImageNet (Russakovsky *et al.*, 2015; Krizhevsky *et al.*, 2012; He *et al.*, 2014; Szegedy *et al.*, 2015a; Ouyang *et al.*, 2014; Simonyan and Zisserman, 2015) and Pascal VOC categories (Girshick *et al.*, 2014; Agrawal *et al.*, 2014). The most successful general object detectors are based on variants of the `R-CNN` framework (Girshick *et al.*, 2014). Given an input image, a reduced set of detection proposals is created, and these are then evaluated via a convnet. This is essentially a two-stage cascade sliding window method.

The most popular proposal method for generic objects is `SelectiveSearch` (Uijlings *et al.*, 2013). Chapter 6 also points out `EdgeBoxes` (Zitnick and Dollár, 2014) as a fast and effective method. For pedestrian detection `DBN-Isol` and `DBN-Mut` use `DPM` (Felzenszwalb *et al.*, 2010) for proposals. `JointDeep`, `MultiSDP`, and `SDN` use a HOG+CSS+linear SVM detector for proposals, similar to Walk *et al.* (2010). Only `ConvNet` (Sermanet *et al.*, 2013) applies a convnet in a sliding window fashion. In chapter 6, we will have a closer look at detection proposals and implications for the detection pipeline.

Decision forests. Until 2015, most proposed methods for pedestrian detection do not use convnets. Leaving methods aside that use optical flow, the top performing methods (on Caltech and KITTI datasets) are `SquaresChnFtrs` (chapter 3), `InformedHaar` (Zhang *et al.*, 2014), `SpatialPooling` (Paisitkriangkrai *et al.*, 2014), `LDCF` (Nam *et al.*, 2014), and `Regionlets` (Wang *et al.*, 2013). All of them are boosted decision forests and can be considered variants of the integral channels features architecture (Dollár *et al.*, 2009a). `Regionlets` and `SpatialPooling` use a large set of features, including HOG, LBP and CSS, while `SquaresChnFtrs`, `InformedHaar`, and `LDCF` build over HOG+LUV. On the Caltech benchmark, the previously best convnet, `SDN`, is outperformed by all aforementioned methods.[5]

Input to convnets. It is important to highlight that `ConvNet` (Sermanet *et al.*, 2013) learns to predict from YUV input pixels, whereas all other methods use additional hand-crafted features. `DBN-Isol` and `DBN-Mut` use HOG features as input. `MultiSDP` uses HOG+CSS features as input. `JointDeep` and `SDN` uses YUV+Gradients as input (and HOG+CSS for the detection proposals). We will show in our experiments that good performance can be reached using RGB alone, but we also show that more sophisticated inputs systematically improve detection quality. Our data indicates that the antagonism "hand-crafted features versus convnets" is an illusion.

4.1.1 Contributions

In this chapter we propose to revisit pedestrian detection with convolutional neural networks by carefully exploring the design space (number of layers, filter sizes, etc.), and the critical implementation choices (training data preprocessing, effect of detection proposals, etc.). We show that both small (10^5 parameters) and large ($6 \cdot 10^7$ parameters) networks can reach good performance when trained from scratch (even when using the same data as previous methods). We also show the benefits of using extended and external data, which leads to the strongest single-frame detector on Caltech at the time of this study. At the time of publication, we report the best known performance for a convnet on the challenging Caltech dataset (improving by more than 10 percent points) and the first convnet results on the KITTI dataset.

4.2 Training data

It is well known that for convnets the volume of training data is quite important to reach good performance. Below are the datasets we consider in this chapter.

Caltech. See section 3.2.1 for details about the Caltech dataset.

Caltech validation set. In our experiments we also use Caltech training data for validation. For those experiments we use one of the suggested validation splits (Dollár *et al.*, 2012b): the first five training videos are used for validation training and the sixth training video for validation testing.

[5] `Regionlets` matches `SpatialPooling` on the KITTI benchmark, and, assuming transitivity, would improve over `SDN` on Caltech.

KITTI. The KITTI dataset (Geiger *et al.*, 2012) consists of videos captured from a car traversing German streets, also under good weather conditions. Although similar in appearance to Caltech, it has been shown to have different statistics (see Benenson *et al.* (2014, supplementary material)). Its training set contains 4 445 pedestrians (4 024 taller than 40 pixels) over 7 481 frames, and its test set 7 518 frames.

ImageNet, Places. In section 4.5 we will consider using large convnets that can exploit pre-training for surrogate tasks. We consider two such tasks (and their associated datasets), the ImageNet 2012 classification of a thousand object categories (Krizhevsky *et al.*, 2012; Russakovsky *et al.*, 2015; Girshick *et al.*, 2014) and the classification of 205 scene categories (Zhou *et al.*, 2014). The datasets provide $1.2 \cdot 10^6$ and $2.5 \cdot 10^6$ annotated images for training, respectively.

4.3 From decision forests to neural networks

Before diving into the experiments, it is worth noting that the proposal method we are using, `SquaresChnFtrs` (see section 4.4.1), can be converted into a convnet. The overall system then becomes a cascade of two neural networks.

`SquaresChnFtrs` (chapter 3) is a decision forest composed of 2 048 level-2 decision trees, applied over ten hand-crafted feature channels (HOG+LUV). These channels are sum-pooled over rectangular regions and fed into the split nodes of the trees. This architecture can be mapped into a convnet (Sethi, 1990; Cios and Liu, 1992; Ivanova and Kubat, 1995; Banerjee, 1997; Setiono and Leow, 1999).

As mentioned in section 2.2, using non-RGB input is a standard practice for pedestrian detection with convnets (more on this in section 4.4.4), we thus focus on converting the pooling and decision forest. The sum-pooling stage maps directly to an inner product layer. Each decision tree maps to a small column of two hidden layers, with sign-function non-linearities (hard non-linearities). Finally the output of all trees is combined via linear weighting.

The mapping from `SquaresChnFtrs` to a deep neural network is exact: evaluating the same inputs it will return the exact same outputs. What is special about the resulting network is that it has not been trained by back-propagation, but via Adaboost (Bengio *et al.*, 2005). This network already performs better than the previously best convnet on Caltech, `SDN` (Luo *et al.*, 2014).

Unfortunately, experiments to soften the non-linearities and use back-propagation to fine-tune the model parameters did not show significant improvements. We suspect that the parameters found via Adaboost are a local minimum that is hard to escape via stochastic gradient descent.

4.4 Vanilla convolutional networks

In our experience many convnet architectures and training hyper-parameters do not enable effective learning for diverse and challenging tasks. It is thus considered best

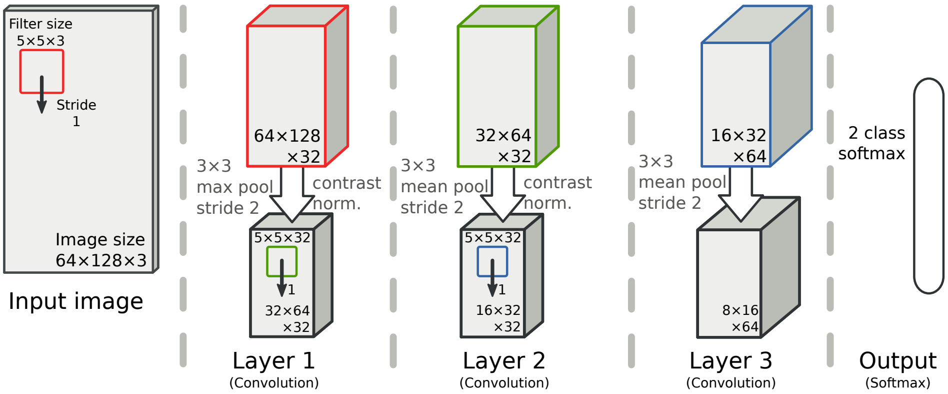


Figure 4.2: Illustration of the CifarNet, $\sim 10^5$ parameters.

practice to start exploration from architectures and parameters that are known to work well and progressively adapt it to the task at hand. This is the strategy of the following sections.

In this section we first consider CifarNet, a small network designed to solve the CIFAR-10 classification problem (10 objects categories, $(5+1)\cdot 10^5$ colour images of 32×32 pixels) (Krizhevsky, 2009). In section 4.5 we consider AlexNet, a network that has 600 times more parameters than CifarNet and designed to solve the ILSVRC2012 classification problem (1 000 objects categories, $(1.2+0.15)\cdot 10^6$ colour images of ∼VGA resolution). Both of these networks were introduced in Krizhevsky *et al.* (2012) and are re-implemented in the open source Caffe project (Jia *et al.*, 2014)[6].

Although pedestrian detection is quite a different task than CIFAR-10, we decide to start our exploration from the CifarNet, which provides fair performance on CIFAR-10. Its architecture is depicted in figure 4.2, unless otherwise specified we use raw RGB input.

We first discuss how to use the CifarNet network (section 4.4.1). This naïve approach already improves over the previously best convnets for pedestrian detection (section 4.4.2). Sections 4.4.3 and 4.4.4 explore the design space around CifarNet and further push the detection quality. All models in this section are trained using Caltech data only (see section 4.2).

4.4.1 How to use CifarNet?

Given an initial network specification, there are still several design choices that affect the final detection quality. We discuss some of them in the following paragraphs.

Detection proposals. Unless otherwise specified we use the `SquaresChnFtrs` (chapter 3) detector to generate proposals because, at the time of publication of this chapter in Hosang *et al.* (2015), it is the best performing pedestrian detector (on Caltech) with source code available. In figure 4.3 we compare `SquaresChnFtrs` against `EdgeBoxes` (Zit-

[6] http://caffe.berkeleyvision.org

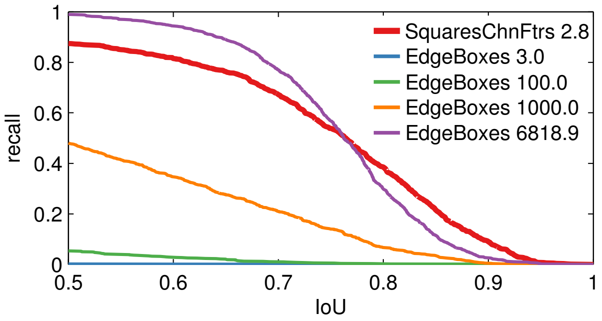


Figure 4.3: Recall of ground truth annotations versus Intersection-over-Union threshold on the Caltech test set. The legend indicates the average number of detection proposals per image for each curve. A pedestrian detector generates much better proposals than a state of the art generic method (`EdgeBoxes` (Zitnick and Dollár, 2014)).

Positives	Negatives	MR
GT	Random	83.1%
GT	`IoU` < 0.5	37.1%
GT	`IoU` < 0.3	37.2%
GT, `IoU` > 0.5	`IoU` < 0.5	42.1%
GT, `IoU` > 0.5	`IoU` < 0.3	41.3%
GT, `IoU` > 0.75	`IoU` < 0.5	39.9%

Table 4.1: Effect of positive and negative training sets on the detection quality. MR: log-average miss-rate on Caltech validation set. GT: ground truth bounding boxes.

nick and Dollár, 2014), a state of the art class-agnostic proposal method. Using class-specific proposals allows to reduce the number of proposals by three orders of magnitude.

Other than `ConvNet` (Sermanet *et al.*, 2013) (which does not use proposals), all other competing convnets also use a pedestrian detector for proposals (see also section 4.4.2).

Thresholds for positive and negative samples. Given both training proposals and ground truth (GT) annotations, we now consider which training label to assign to each proposal. A proposal is considered to be a positive example if it exceeds a certain Intersection-over-Union (IoU) threshold for at least one GT annotation. It is considered negative if it does not exceed a second IoU threshold for any GT annotation, and is ignored otherwise. We find that using GT annotations as positives is beneficial (i.e. not applying significant jitter, see table 4.1).

Model window size. A typical choice for pedestrian detectors is a model window of 128 × 64 pixels in which the pedestrian occupies an area of 96 × 48 (Dalal and Triggs, 2005; Dollár *et al.*, 2009a; Benenson *et al.*, 2013, 2014). It is unclear that this is the ideal input size for convnets. Despite CifarNet being designed to operate over 32 × 32 pixels, table 4.2 shows that a model size of 128 × 64 pixels indeed works best. We experimented

Window size	MR
32×32	50.6%
64×32	48.2%
128×64	39.9%
128×128	49.4%
227×227	54.9%

Table 4.2: Effect of the window size on the detection quality. MR: see table 4.1.

Ratio	MR
None	41.4%
1 : 10	40.6%
1 : 5	39.9%
1 : 1	39.8%

Table 4.3: Detection quality as a function of the strictly enforced ratio of positives:negatives in each training batch. *None*: no ratio enforced. MR: see table 4.1.

Method	Proposal	Test MR
Proposals of JointDeep	-	45.5%
JointDeep	Proposals of JointDeep	39.3% (Ouyang and Wang, 2013a)
SDN		37.9% (Luo *et al.*, 2014)
CifarNet		36.5%
SquaresChnFtrs	-	34.8% (chapter 3)
CifarNet	SquaresChnFtrs	*30.7%*

Table 4.4: Detection quality as a function of the method and the proposals used for training and testing (MR: log-average miss-rate on Caltech test set). When using the exact same training data as JointDeep (Ouyang and Wang, 2013a), our vanilla CifarNet already improves over the previous best known convnet on Caltech (SDN, Luo *et al.* 2014).

with other variants (stretching versus cropping, larger context border) with no clear improvement.

Training batch. In a detection setup, training samples are typically highly imbalanced towards the background class. Although in our validation setup the imbalance is limited, we found it beneficial throughout our experiments to enforce a strict ratio of positive to negative examples per batch of the stochastic gradient descend optimisation (see table 4.3). The final performance is not sensitive to this parameter as long as some ratio (vs. *None*) is maintained. We use a ratio of 1 : 5.

4.4.2 How far can we get with the CifarNet?

Given the parameter selection on the validation set from previous sections, how does CifarNet compare to previous convnet results on the Caltech test set? Table 4.4 and figure 4.1 show that our naive network right away improves over the previously best convnet (30.7% MR versus SDN 37.9% MR).

To decouple the contribution of our strong SquaresChnFtrs proposals to the CifarNet performance, we also train a CifarNet using the proposal from JointDeep (Ouyang

# layers	Architecture	MR
3	CONV1 CONV2 CONV3 (CifarNet, fig. 4.2)	*37.1%*
	CONV1 CONV2 LC	43.2%
	CONV1 CONV2 FC	47.6%
4	CONV1 CONV2 CONV3 FC	39.6%
	CONV1 CONV2 CONV3 LC	40.5%
	CONV1 CONV2 FC1 FC2	43.2%
	CONV1 CONV2 CONV3 CONV4	43.3%
DAG	CONV1 CONV2 CONV3 CONCAT23 FC	38.4%

Table 4.5: Detection quality of different network architectures (MR: log-average miss-rate on Caltech validation set), sorted by number of layers before soft-max. DAG: directed acyclic graph.

and Wang, 2013a). When using the same detection proposals at training and test time, the vanilla CifarNet already improves over both custom-designed `JointDeep` and `SDN`.

Our CifarNet results are surprisingly close to the previously best known pedestrian detector trained on Caltech1x (30.7% MR versus `SpatialPooling` 29.2% MR (Paisitkriangkrai *et al.*, 2014)).

4.4.3 Exploring different architectures

Encouraged by our initial results, we proceed to explore different parameters for the CifarNet architecture.

Number and size of convolutional filters. Using the Caltech validation set we perform a swipe of convolutional filter sizes (3×3, 5×5, or 7×7 pixels) and number of filters at each layer (16, 32, or 64 filters). We include the full table in the supplementary material. We observe that using large filter sizes hurts quality, while the varying the number of filters shows less impact. Although some fluctuation in miss-rate is observed, overall there is no clear trend indicating that a configuration is clearly better than another. Thus, for sake of simplicity, we keep using CifarNet (32-32-64 filters of 5×5 pixel) in the subsequent experiments.

Number and type of layers. In table 4.5 we evaluate the effect of changing the number and type of layers, while keeping other CifarNet parameters fix. Besides convolutional layers (CONV) and fully-connected layers (FC), we also consider locally-connected layers (LC) (Taigman *et al.*, 2014), and concatenating features across layers (CONCAT23) (used in `ConvNet` (Sermanet *et al.*, 2013)). None of the considered architecture changes improves over the original CifarNet.

Input channels	# channels	CifarNet
RGB	3	39.9%
LUV	3	46.5%
G+LUV	4	40.0%
HOG+L	7	36.8%
HOG+LUV	10	40.7%

Table 4.6: Detection quality when changing the input channels network architectures. Results in MR; log-average miss-rate on Caltech validation set. G indicates luminance channel gradient, HOG indicates G plus G spread over six orientation bins (hard-binning). These are the same input channels used by our `SquaresChnFtrs` proposal method.

4.4.4 Input channels

As discussed in section 2.2, the majority of previous convnets for pedestrian detection use gradient and colour features as input, instead of raw RGB. In table 4.6 we evaluate the effect of different input features over CifarNet. It seems that HOG+L channel provide a small advantage over RGB.

For purposes of direct comparison with the large networks, in the next sections we keep using raw RGB as input for our CifarNet experiments. We report the CifarNet test set results in section 4.6.

4.5 Large convolutional network

One appealing characteristic of convnets is their ability to scale in size of training data volume. In this section we explore larger networks trained with more data.

We base our experiments on the `R-CNN` (Girshick *et al.*, 2014) approach, which is, at the time this research was conducted, one of the best performer on the Pascal VOC detection task (Everingham *et al.*, 2014). We are thus curious to evaluate its performance for pedestrian detection.

4.5.1 Surrogate tasks for improved detections

The `R-CNN` approach ("Regions with CNN features") wraps the large network previously trained for the ImageNet classification task (Krizhevsky *et al.*, 2012), which we refer to as AlexNet (see figure 4.4). We use "AlexNet" as shorthand for "`R-CNN` with AlexNet" with the distinction made clear by the context. During `R-CNN` training AlexNet is fine-tuned for the detection task, and in a second step, the softmax output is replaced by a linear SVM. Unless otherwise specified, we use the default parameters of the open source, Caffe-based `R-CNN` implementation[7]. Like in the previous sections, we use `SquaresChnFtrs` for detection proposals. For consistency with other AlexNet

[7] https://github.com/rbgirshick/rcnn

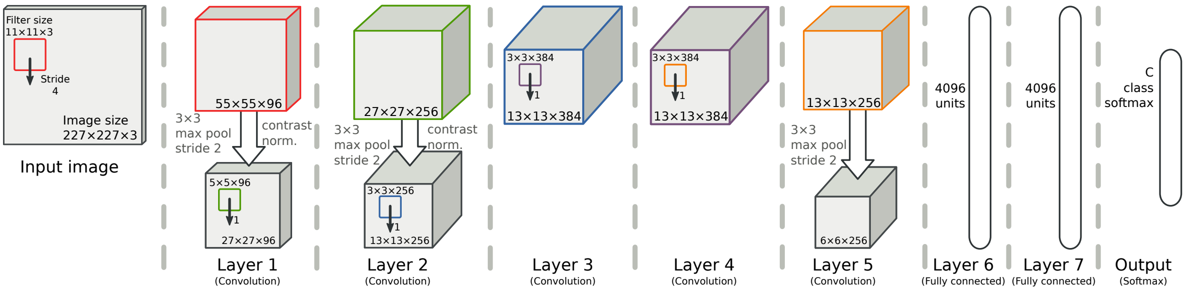


Figure 4.4: Illustration of the AlexNet architecture, $\sim 6 \cdot 10^7$ parameters.

experiments in the literature, we use the default RGB and 227 × 227 input size (the optimal CifarNet parameters might not apply to the larger AlexNet).

Pre-training. If we only train the top layer SVM, without fine-tuning the lower layers of AlexNet, we obtain 39.8% MR on the Caltech test set. This is already surprisingly close to the previously best convnet for the task (`SDN` 37.9% MR). When fine-tuning all layers on Caltech, the test set performance increases dramatically, reaching 25.9% MR. This confirms the effectiveness of the general `R-CNN` recipe for detection (train AlexNet on ImageNet, fine-tune for the task of interest).

In table 4.7 we investigate the influence of the pre-training task by considering AlexNets that have been trained for scene recognition (Zhou *et al.*, 2014) ("Places", see section 4.2) and on both Places and ImageNet. "Places" provides results close to ImageNet, suggesting that the exact pre-training task is not critical and that there is nothing special about ImageNet.

Caltech10x. Due to the large number of parameters of AlexNet, we consider providing additional training data using Caltech10x for fine-tuning the network (see section 4.2). Despite the strong correlation across training samples, we do observe further improvement (see table 4.7). Interestingly, the bulk of the improvement is due to more pedestrians (Positives10x, uses positives from Caltech10x and negatives from Caltech1x). Our top result, 23.3% MR, makes our AlexNet setup the best reported single-frame detector on Caltech (i.e. no optical flow) at the time in which this research was conducted.

4.5.2 Caltech-only training

To compare with CifarNet, and to verify whether pre-training is necessary at all, we train AlexNet "from scratch" using solely the Caltech training data. We collect results in table 4.7.

Training AlexNet solely on Caltech, yields 32.4% MR, which improves over the proposals (`SquaresChnFtrs` 34.8% MR) and the previous best known convnet on Caltech (`SDN` 39.8% MR). Using Caltech10x further improves the performance, down to 27.5% MR.

AlexNet training	Fine-tuning	SVM training	Test MR
Random	none	Caltech1x	86.7%
ImageNet	none	Caltech1x	39.8%
Places+Imagenet	Caltech1x	Caltech1x	30.1%
Places			27.0%
ImageNet			25.9%
ImageNet	Positives10x	Positives10x	23.8%
	Caltech10x	Caltech10x	*23.3%*
Caltech1x	-	Caltech1x	32.4%
	-	Caltech10x	32.2%
Caltech10x	-	Caltech1x	*27.4%*
	-	Caltech10x	*27.5%*
SquaresChnFtrs (chapter 3)			34.8%

Table 4.7: Detection quality when using different training data in different training stages of the AlexNet: initial training of the convnet, optional fine-tuning of the convnet, and the SVM training. Positives10x: positives from Caltech10x and negatives from Caltech1x. Detection proposals provided by SquaresChnFtrs, result included for comparison. See section 4.5.1 and 4.5.2 for details.

Although these numbers are inferior to the ones obtained with ImageNet pre-training (23.3% MR, see table 4.7), we can get surprisingly competitive results using only pedestrian data despite the 10^7 free parameters of the AlexNet model. At the time when we published this study (Hosang *et al.*, 2015), AlexNet with Caltech10x is second best single-frame pedestrian detector on Caltech (best was LDCF 24.8% MR, which also uses Caltech10x).

4.5.3 Additional experiments

How many layers? So far all experiments use the default parameters of R-CNN. Previous works have reported that, depending on the task, using features from lower AlexNet layers can provide better results (e.g. Agrawal *et al.*, 2014; Razavian *et al.*, 2014; Azizpour *et al.*, 2015). Table 4.8 reports Caltech validation results when training the SVM output layer on top of layers four to seven (see figure 4.4). We report results when using the default parameter settings and parameters that have been optimised by grid search (parameters are SVM regularisation and negative example overlap).

We observe a negligible difference between default and optimized parameter (at most 1 percent point). Results for default parameters exhibit a slight trend of better performance for higher levels. These validation set results indicate that, for pedestrian detection, the R-CNN default parameters are a good choice overall.

Parameters	fc7	fc6	pool5	conv4
Default	32.2%	32.5%	33.4%	42.7%
Best	32.0%	31.8%	32.5%	42.4%

Table 4.8: Detection quality when training the R-CNN SVM over different layers of the finetuned CNN. Results in MR; log-average miss-rate on Caltech validation set. “Best parameters” are found by exhaustive search on the validation set.

Effect of proposal method. When comparing the performance of AlexNet fine-tuned on Caltech1x to the proposal method, we see an improvement of 9 pp (percent points) in miss-rate. In table 4.9 we study the impact of using weaker or stronger proposals. Both `ACF` (Dollár *et al.*, 2014) and `SquaresChnFtrs` (chapter 3) provide source code, allowing us to generate training proposals. `Katamari` (chapter 3) and `SpatialPooling+` (Paisitkriangkrai *et al.*, 2014) are top performers on the Caltech dataset, both using optical flow, i.e. additional information at test time. There is a ∼10 pp gap between the detectors `ACF`, `SquaresChnFtrs`, and `Katamari`/`SpatialPooling`, allowing us to cover different operating points.

The results in table 4.9 indicate that, despite the 10 pp gap, there is no noticeable difference between AlexNet models trained with `ACF` or `SquaresChnFtrs`. It is seems that as long as the proposals are not random (see top row of table 4.1), the obtained quality is rather stable. The results also indicate that the quality improvement from AlexNet saturates around ∼22% MR. Using stronger proposals does not lead to further improvement. This means that the discriminative power of our trained AlexNet is on par with the previously best known models on the Caltech dataset, but does not overtake them.

KITTI test set. In figure 4.5 we show performance of the AlexNet in context of the KITTI pedestrian detection benchmark (Geiger *et al.*, 2012). The network is pre-trained on ImageNet and fine-tuned using KITTI training data. `SquaresChnFtrs` reaches 44.4% AP (average precision), which AlexNet can improve to 50.1% AP. These are the first published results for convnets on the KITTI pedestrian detection dataset.

Albeit the ranking with `SpatialPooling` changes, it should be noted that a) the two datasets use different evaluation metrics, b) the two datasets are more dissimilar than they seem on the surface (see table 3.2), and c) overall AlexNet results on KITTI remain satisfactory; using proposals with higher recall might further improve results.

4.5.4 Error analysis

Results from the previous section are encouraging, but not as good as could be expected from looking at improvements on Pascal VOC. So what bounds performance? The proposal method? The localization quality of the convnet?

Looking at the highest scoring false positives paints a picture of localization errors of the proposal method, the `R-CNN`, and even the ground truth. To quantify this effect

Fine-tuning	Training proposals	Testing proposals	Test MR	Δ vs. proposals
1×	ACF	ACF	34.5%	9.7%
	SCF	ACF	34.3%	9.9%
	ACF	SCF	26.9%	7.9%
	SCF	SCF	25.9%	8.9%
	ACF	Katamari	25.1%	−2.6%
	SCF	Katamari	24.2%	−1.7%
10×	SCF	LDCF	23.4%	1.4%
	SCF	SCF	23.3%	11.5%
	SCF	SP+	22.0%	−0.1%
	SCF	Katamari	21.6%	0.9%
ACF (Dollár *et al.*, 2014)			44.2%	
SCF: SquaresChnFtrs (chapter 3)			34.8%	
LDCF (Nam *et al.*, 2014)			24.8%	
Katamari (chapter 3)			22.5%	
SP+: SpatialPooling+ (Paisitkriangkrai *et al.*, 2016)			21.9%	

Table 4.9: Effect of proposal methods on detection quality of R-CNN. 1×/10× indicates fine-tuning on Caltech or Caltech10x. Test MR: log-average miss rate on Caltech test set. Δ: the improvement in MR of the rescored proposals over the test proposals alone.

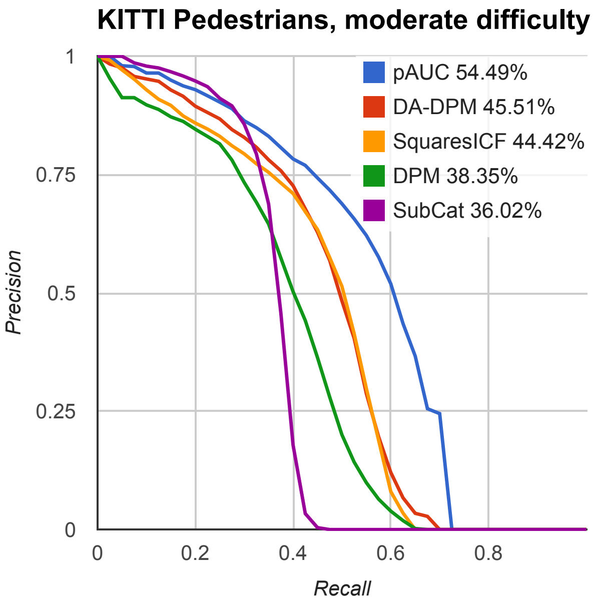


Figure 4.5: AlexNet over on KITTI test set.

we rerun the Caltech evaluation but remove all false positives that touch an annotation. This experiment provides an upper bound on performance when solving localisation issues in detectors and doing perfect non-maximum suppression. We see a surprisingly consistent improvement for all methods of up to 2% MR (see supplementary material). This means that the intuition we gathered from looking at false positives is wrong and actually almost all of the mistakes that worsen the MR are actually background windows that are mistaken for pedestrians. What is striking about this result is that this is not just the case for our R-CNN experiments on detection proposals but also for methods that are trained as a sliding window detector.

4.6 Small or big convnet?

Since we have analysed the CifarNet and AlexNet separately, we compare their performance in this section side by side. Table 4.10 shows performance on the Caltech test set for models that have been trained only on Caltech1x and Caltech10x. With less training data the CifarNet reaches 30.7% MR, performing 2 percent points better than the AlexNet. On Caltech10x, we find the CifarNet performance improved to 28.4%, while the AlexNet improves to 27.1% MR. The trend confirms the intuition that models with lower capacity saturate earlier when increasing the amount of training data than models with higher capacity. We can also conclude that the AlexNet would profit from better regularisation when training on Caltech1x.

Timing. The runtime during detection is about 3ms per proposal window. This is too slow for sliding window detection, but given a fast proposal method that has high recall with less than 100 windows per image, scoring takes about 300ms per image. In our experience `SquaresChnFtrs` runs in 2s per image, so proposing detections takes most of the detection time.

Architecture	#	Test MR	
training	parameters	Caltech1x	Caltech10x
CifarNet	$\sim 10^5$	30.7%	28.4%
MediumNet	$\sim 10^6$	–	27.9%
AlexNet	$\sim 10^7$	32.4%	27.5%
`SquaresChnFtrs` (chapter 3)		34.8%	

Table 4.10: Selection of results (presented in previous sections) when training different networks using Caltech training data only. MR: log-average miss-rate on Caltech test set.

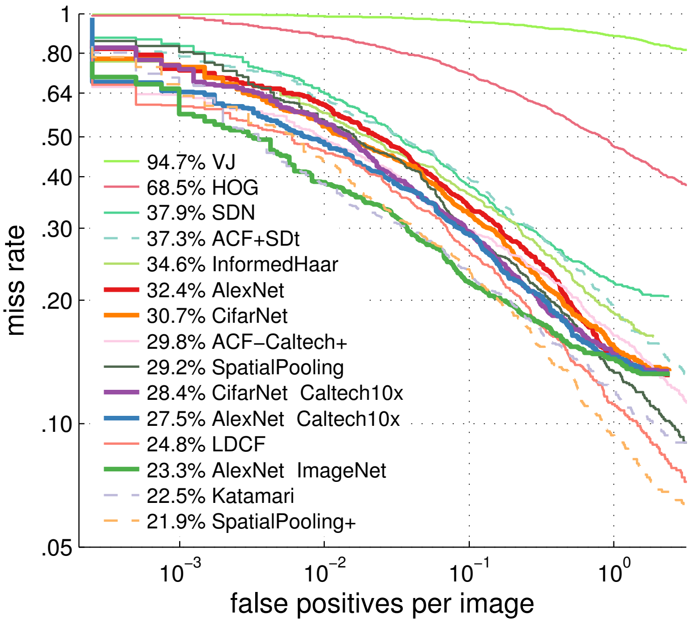


Figure 4.6: Comparison of our key results (thick lines) with published methods on Caltech test set. Methods using optical flow are dashed.

4.7 Takeaways

Previous work suggests that convnets for pedestrian detection underperform, despite having involved architectures (see chapter 3 for a survey of pedestrian detection). In this chapter we showed that neither has to be the case. We present a wide range of experiments with two off-the-shelf models that reach competitive performance: the small CifarNet and the big AlexNet.

We present two networks that are trained on Caltech only, which outperform all previously published convnets on Caltech. The CifarNet shows better performance than related work, even when using the same training data as the respective methods (section 4.4.2). Despite its size, the AlexNet also improves over previous convnets even when it is trained on Caltech only (section 4.5.2).

At time of publication we advanced the state of the art for pedestrian detectors that have been trained on Caltech1x and Caltech10x. The CifarNet was the best single-frame pedestrian detector that has been trained on Caltech1x (section 4.4.2), while AlexNet was the best single-frame pedestrian detector trained on Caltech10x (section 4.5.2).

In figure 4.6, we include all previously published methods on Caltech for the comparison, which also adds methods that use additional information at test time. The AlexNet that has been pre-trained on ImageNet reaches competitive results to the best previously published methods, but without using additional information at test time (section 4.5.1).

We report first results for convnets on the KITTI pedestrian detection benchmark. The AlexNet improves over the proposal method alone, delivering encouraging results to further push KITTI performance with convnets.

4.8 Conclusion

We have presented extensive and systematic experimental evidence on the effectiveness of convnets for pedestrian detection. Compared to previous convnets applied to pedestrian detection our approach avoids custom designs. When using the exact same proposals and training data as previous approaches our “vanilla” networks outperform previous results.

We have shown that with pre-training on surrogate tasks, convnets can reach top performance on this task. Interestingly we have shown that even without pre-training competitive results can be achieved, and this result is quite insensitive to the model size (from 10^5 to 10^7 parameters). Our experiments also detail which parameters are most critical to achieve top performance. At the time of publication of this study, we report the best known results for convnets on both the challenging Caltech and KITTI datasets.

Our experience with convnets indicates that they show good promise on pedestrian detection, and that reported best practices do transfer to this task. That being said, on this more mature field we do not observe the large improvement seen on datasets such as Pascal VOC and ImageNet.

5 Towards human performance pedestrian detection

We saw how crucial feature engineering historically was for driving performance of pedestrian detection in chapter 3. In chapter 4, we showed that standard convolutional neural networks (convnets) are a powerful tool for learning features for pedestrian detection directly from RGB images. Since the publication of that work, more research on convnets for pedestrian detection has enabled significant improvement without signs for slowing down.

Encouraged by the recent progress in pedestrian detection, we investigate the gap between current state-of-the-art methods and the "perfect single frame detector". We enable our analysis by creating a human baseline for pedestrian detection (over the Caltech dataset). After manually clustering the frequent errors of a top detector, we characterise both localisation and background-versus-foreground errors.

To address localisation errors we study the impact of training annotation noise on the detector performance, and show that we can improve results even with a small portion of sanitised training data. To address background/foreground discrimination, we study convnets for pedestrian detection, and discuss which factors affect their performance.

Other than our in-depth analysis, we report top performance on the Caltech dataset, and provide a new sanitised set of training and test annotations.

An earlier version of this work was published at CVPR (Zhang *et al.*, 2016b) and this revision is under minor revision at PAMI. Shanshan Zhang was the lead author and provided most experiments, Rodrigo Benenson and Mohamed Omran provided annotations, and Jan Hosang contributed the AlexNet experiments in section 5.5.2 and the localisation/background analysis in section 5.3.2.2 and 5.5.2.

5.1 Introduction

Despite the extensive research on pedestrian detection, recent papers still show significant improvements, suggesting that a saturation point has not yet been reached. In this chapter we analyse the gap between the state of the art and a newly created human baseline (section 5.3.1). The results indicate that there is still a ten fold improvement to be made before reaching human performance. We aim to investigate which factors will help close this gap.

We analyse failure cases of top performing pedestrian detectors and diagnose what should be changed to further push performance. We show several different analysis (section 5.3.2), including human inspection, automated analysis of problem cases (e.g. blur, contrast), and oracle experiments. Our results indicate that localisation is an important

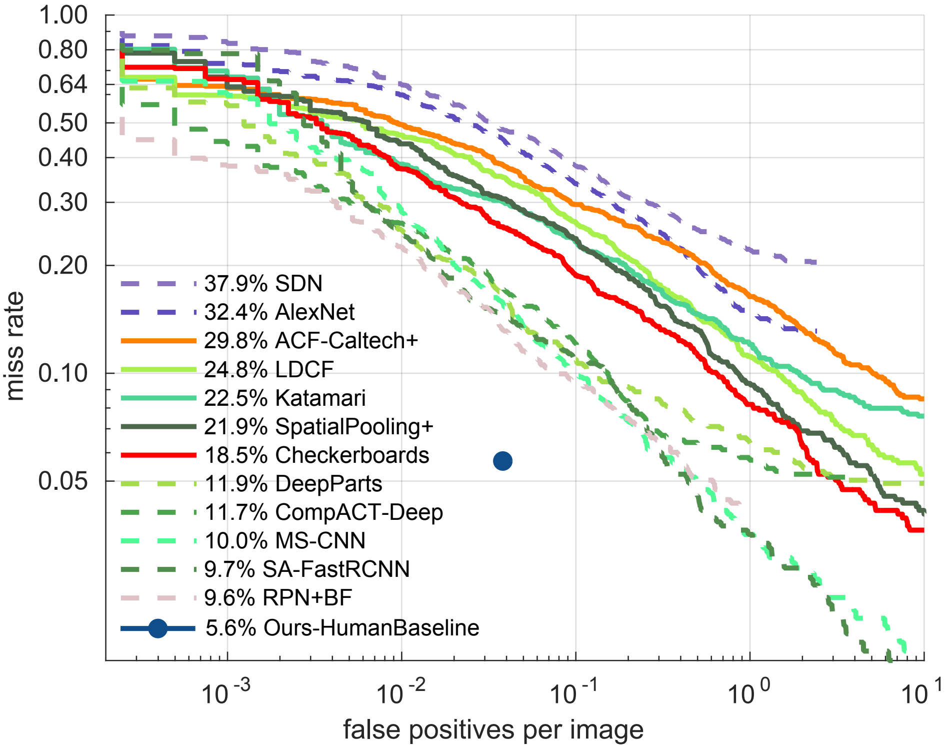


Figure 5.1: Overview of the top results on the Caltech pedestrian benchmark. At ∼95% recall, state-of-the-art detectors make ten times more errors than the human baseline.

source of high confidence false positives. We address this aspect by improving the training set alignment quality, both by manually sanitising the Caltech training annotations and via algorithmic means for the remaining training samples (sections 5.4 and 5.5.1).

To address background versus foreground discrimination, we study convnets for pedestrian detection, and discuss which factors affect their performance (section 5.5.2).

5.1.1 Contributions

Our key contributions are as follows:

1. We provide a detailed analysis of a state-of-the-art pedestrian detector, providing insights into failure cases.

2. We provide a human baseline for the Caltech Pedestrian Benchmark; as well as a sanitised version of the annotations to serve as new, high quality ground truth for the training and test sets of the benchmark. This data is public[8].

3. We analyse the effects of training data quality. More specifically we quantify how much better alignment and fewer annotation mistakes can improve performance.

[8] http://www.mpi-inf.mpg.de/pedestrian_detection_cvpr16

4. Using the insights of the analysis, we explore variants of top performing methods: filtered channel feature detector (Zhang *et al.*, 2015b) and R-CNN detector (Girshick *et al.* (2014) and chapter 4), and show improvements over the baselines.

5.2 Preliminaries

Before delving into our analysis, let us describe the datasets in use, their metrics, and our baseline detectors.

5.2.1 Pedestrian detection benchmarks

Amongst existing pedestrian datasets (Dalal and Triggs, 2005; Ess *et al.*, 2008; Enzweiler and Gavrila, 2009), KITTI (Geiger *et al.*, 2012) and Caltech (Dollár *et al.*, 2012b) are currently the most popular ones, which will be used for analysis in this chapter.

Caltech. See section 3.2.1 for details on the Caltech dataset.

KITTI. The KITTI dataset (Geiger *et al.*, 2012) was captured by driving around the mid-size city of Karlsruhe, in rural areas and on highways. The pedestrian detection benchmark consists of 7481 training images and 7518 test images. Since the ground truth annotations for test set are not publicly available, the analysis for KITTI is implemented by splitting the public training set into train/validation (~4k/2k images) sets.

5.2.2 Evaluation metrics

MR_{-2}, MR_{-4}. In the standard Caltech evaluation (Dollár *et al.*, 2012b) the miss rate (MR) is averaged over the low precision range of $[10^{-2}, 10^{0}]$ FPPI (false positives per image). This metric does not reflect well improvements in localisation errors (lowest FPPI range). Aiming for a more complete evaluation, we extend the evaluation FPPI range from traditional $[10^{-2}, 10^{0}]$ to $[10^{-4}, 10^{0}]$, we denote these MR_{-2} and MR_{-4}. We expect the MR_{-4} metric to become more important as detectors get stronger.

MR^{O}, MR^{N}. In section 5.4 we introduce new annotations on the test set. We show evaluations on both original and new annotations for more comprehensive comparison. O stands for "original annotations", and N stands for "new annotations".

In total, we use four evaluation metrics: MR^{N}_{-2} , MR^{N}_{-4} , MR^{O}_{-2} and MR^{O}_{-4} , for Caltech experiments in this chapter.

5.2.3 Filtered channel feature detectors

For the analysis in this chapter we consider the `Checkerboards` detector Zhang *et al.* (2015b), which is the best method on the Caltech benchmark while this research was conducted. Note `Checkerboards` is the top detector in ICF family. Although some

Filter type	MR^O_{-2}
ACF (Dollár *et al.*, 2014)	44.2
SquaresChnFtrs (Benenson *et al.*, 2014)	34.8
LDCF (Nam *et al.*, 2014)	24.8
RotatedFilters	19.2
Checkerboards (Zhang *et al.*, 2015b)	18.5

Table 5.1: The filter type determines the ICF methods quality.

Base detector	MR^O_{-2}	+Context $\Delta\mathrm{MR}^O_{-2}$	+Flow $\Delta\mathrm{MR}^O_{-2}$
Orig. 2Ped (Ouyang and Wang, 2013b)	48	+ 5	/
Orig. SDt (Park *et al.*, 2013)	45	/	+ 8
SquaresChnFtrs (chapter 3)	35	+ 5	+ 4
Checkerboards (Zhang *et al.*, 2015b)	19	+ 0	+ 1

Table 5.2: Detection quality gain of adding context (Ouyang and Wang, 2013b) and optical flow (Park *et al.*, 2013), as function of the base detector.

recently proposed top methods (shown in figure 5.1) are convnets based, most of them still use ICF detectors to generate proposals, e.g. DeepParts Tian *et al.* (2015a), CompACT-Deep Cai *et al.* (2015), and SA-FastRCNN (Li *et al.*, 2015). Therefore, the analysis and insights on `Checkerboards` are also applicable to other top methods.

The `Checkerboards` detector (Zhang *et al.*, 2015b) is a generalisation of the Integral Channels Feature detector (ICF, Dollár *et al.*, 2009a), which filters the HOG+LUV feature channels before feeding them into a boosted decision forest.

We compare the performance of several detectors from the ICF family in table 5.1, where we can see a big improvement from 44.2% to 18.5% MR^O_{-2} by introducing filters over the feature channels and optimising the filter bank.

Rotated filters. For the experiments involving training new models (in section 5.5.1) we use `RotatedFilters` detector, which is a simplified variant of `LDCF` (Nam *et al.*, 2014). As shown in table 5.1, `RotatedFilters` are significantly better than the original `LDCF`, and only 1 pp (percent point) worse than `Checkerboards`, yet run 6× faster at training and test time.

Additional cues. The review in chapter 3 showed that context and optical flow information can help improve detections. However, as the detector quality improves (table 5.1) the returns obtained from these additional cues erode (table 5.2). Without re-engineering such cues, gains in detection must come from the core detector. Therefore, we only consider pure ICF detectors without using additional cues for analysis in this chapter.

5.2.4 Convnet detectors

In the standard R-CNN framework (Girshick *et al.*, 2014), external methods are used to generate detection proposals, which are then fed into convnets for feature extraction and classification. Such a two-stage strategy saves computation by reducing the number of windows for convnets to process, but on the other hand it let the final detection results affected by the proposal quality. Generally, more proposals are helpful to reach a higher recall, but also increase the chance for convnets to make mistakes.

As reported in chapter 4 and Tian *et al.* (2015b), current top performing convnet methods are sensitive to the underlying detection proposals, thus we first focus on the proposals by optimising the filtered channel feature detectors (more on convnets in section 5.5.2).

5.3 Analysing the state of the art

In this section we estimate a lower bound on the remaining progress available, analyse the mistakes of current pedestrian detectors, and propose new annotations to better measure future progress.

5.3.1 Are we reaching saturation?

Progress on pedestrian detection has been showing no sign of slowing in recent years (Zhang *et al.*, 2015b; Tian *et al.*, 2015b; Benenson *et al.*, 2014), despite recent impressive gains in performance. How much progress can still be expected on current benchmarks? To answer this question, we propose to use a human baseline as lower bound on errors. We asked domain experts to manually "detect" pedestrians in the Caltech test set; machine detection algorithms should be able to at least reach human performance and, eventually, superhuman performance.

Human baseline protocol. To ensure a fair comparison with existing detectors, most of which operate over a single image at a time, we focus on the single frame monocular detection setting. Frames are presented to annotators in random order, and without access to surrounding frames from the source videos. Annotators have to rely on pedestrian appearance and single-frame context rather than (long-term) motion cues.

The Caltech benchmark normalises the aspect ratio of all detection boxes to 0.41 (Dollár *et al.*, 2012b). Thus our human annotations are done by drawing a line from the top of the head to the point between both feet. A bounding box is then automatically generated such that its centre coincides with the centre point of the manually-drawn axis, see illustration in figure 5.2. This procedure ensures the box is well centred on the subject (which is hard to achieve when marking a bounding box).

To check for consistency among the two annotators, we let both annotate a subset of test images ($\sim 10\%$) and evaluated these separately. With an Intersection over Union (IoU) ≥ 0.8 matching criterion, the results were identical up to a single bounding box.

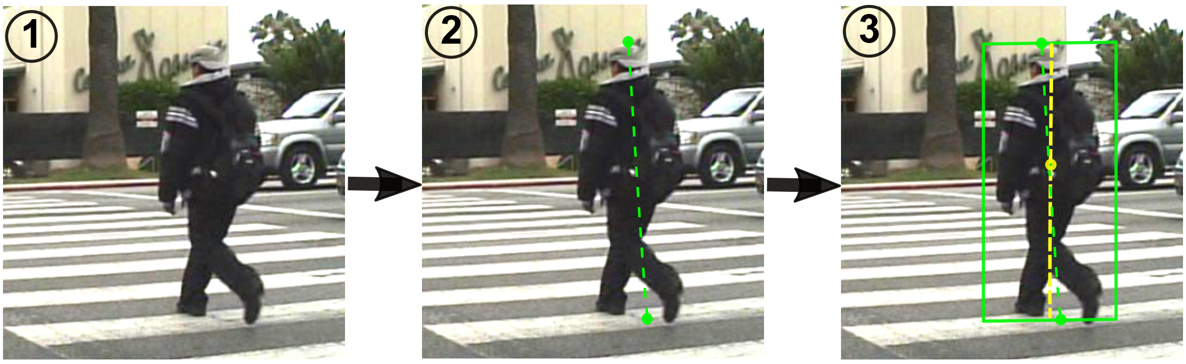


Figure 5.2: Illustration of bounding box generation for human baseline. The annotator only needs to draw a line from the top of the head to the central point between both feet, a tight bounding box is then automatically generated.

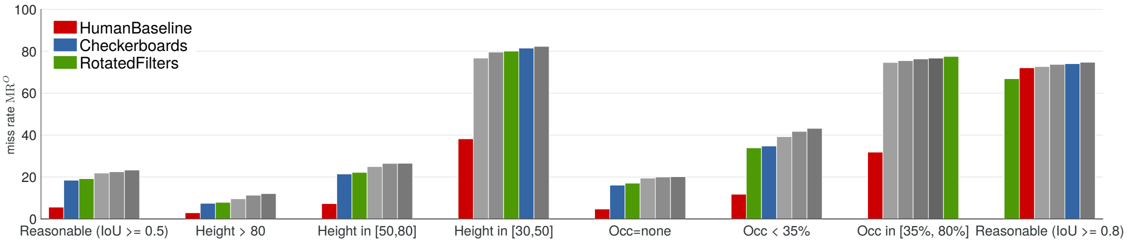


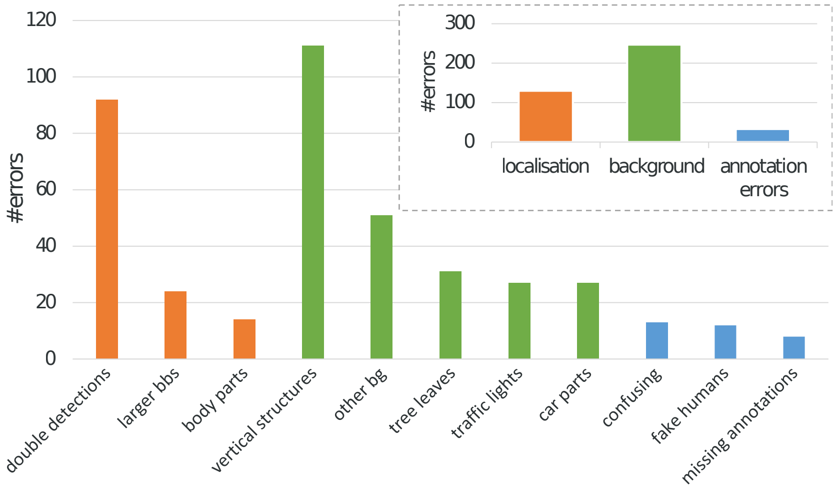
Figure 5.3: Detection quality (log-average miss rate) for different test set subsets. Each group shows the human baseline, the Checkerboards (Zhang *et al.*, 2015b) and RotatedFilters detectors, as well as the next top three (unspecified) methods (different for each setting).

In figure 5.3, we compare our human baseline with Checkerboards, RotatedFilters and other on par methods on different subsets of the test data (varying height ranges and occlusion levels). We find that the human baseline widely outperforms state-of-the-art detectors in all settings[9]. We also notice the gap between human baseline and state-of-the-art detectors is especially large for harder cases, e.g. small scale and heavy occlusion.

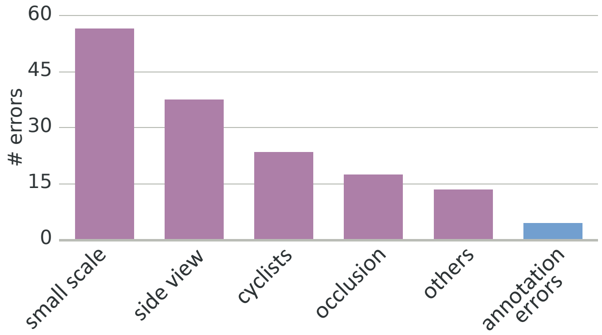
Figure 5.3 also shows that Checkerboards and RotatedFilters perform well across all subsets. In the few cases where they are not top ranked, all methods exhibit low detection quality. Checkerboards is not optimised for the most common case on the Caltech dataset, but nevertheless shows good performance across a variety of situations and is thus an interesting method to analyse.

Conclusion. We are not reaching saturation yet; there is still room for improvement for automatic methods.

[9] Except for IoU ≥ 0.8. This is due to inaccuracies of the ground truth, discussed in section 5.4.



(a) False positive sources



(b) False negative sources

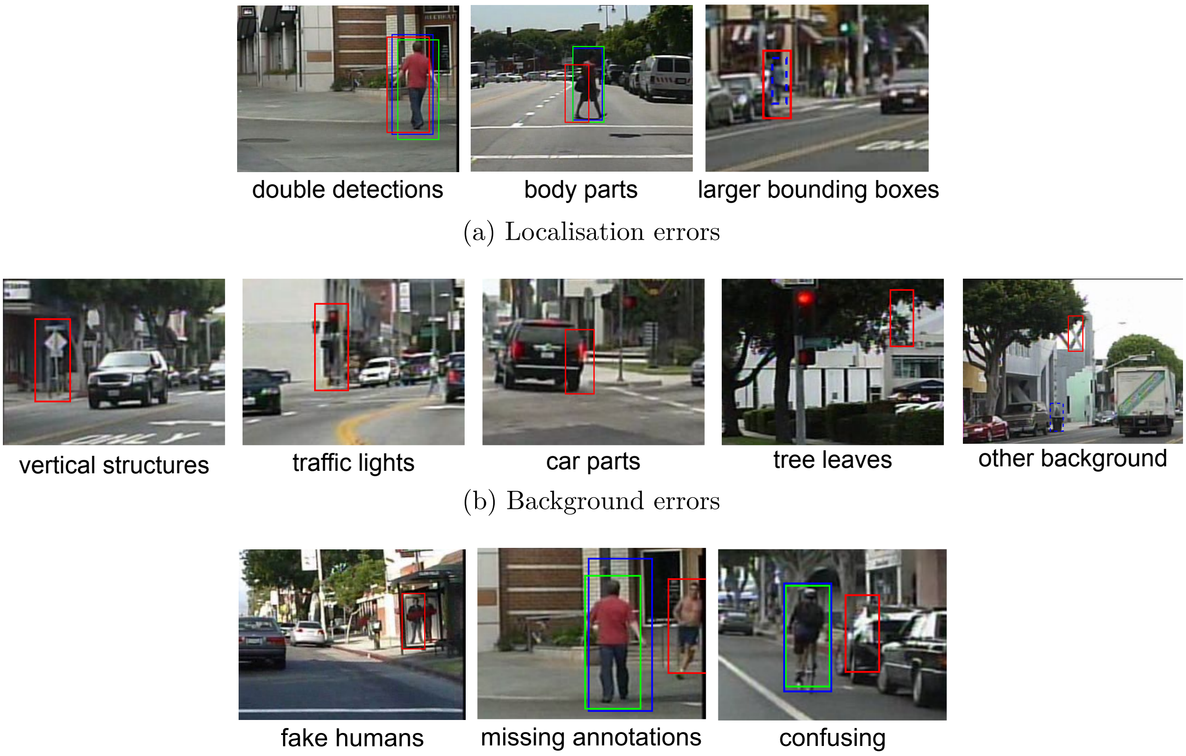
Figure 5.4: Error sources of `Checkerboards` (Zhang *et al.*, 2015b) on the Caltech test set.

5.3.2 Failure analysis

Since there is room for improvement for existing detectors, we want to know: when do they fail? In this section we analyse detection mistakes of `Checkerboards`, which obtains top performance on most subsets of the test set (see figure 5.3). Since most top methods of figure 5.1 are of the ICF family, we expect a similar behaviour for them too. Methods using convnets with proposals based on ICF detectors will also be affected.

5.3.2.1 Error sources

There are two types of errors a detector can do: false positives (detections on background, multiple detections, or poorly localised detections) and false negatives (low-scoring or missing pedestrian detections). In this analysis, we look into false positive and false negative detections at 0.1 false positives per image (FPPI, 1 false positive every 10 images), and manually cluster them into visually distinctive groups. A total of 402 false positive and 148 false negative detections (missing recall) are categorised by error type, as shown in figure 5.4.



(a) Localisation errors

(b) Background errors

(c) Annotation errors

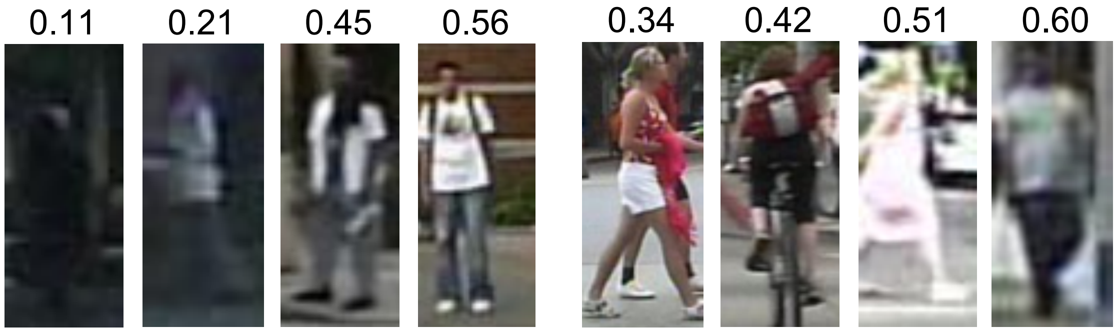
Figure 5.5: Example false positives (`Checkerboards`) from different sources. False positives in red, original annotations in blue, ignore annotations in dashed blue and true positives in green .

False positives. After inspection, all false positives are assigned to one of eleven categories, shown in figure 5.4a. These categories fall into three groups: localisation, background, and annotation errors. We show some examples for each category in figure 5.5a, 5.5b, and 5.5c. Localisation errors are defined as false detections overlapping with ground truth bounding boxes, while background errors have zero overlap with any ground truth annotation.

Background errors are most common, mainly vertical structures (e.g. figure 5.5b), tree leaves, and traffic lights. This indicates that the detectors need to be extended with a better *vertical context*, providing visibility over larger structures and a rough height estimate. In section 5.5.2 we explore how to better handle background errors by using convnets, which has a larger receptive filed (i.e. more context involved) than `Checkerboards`.

Localisation errors are dominated by double detections (high scoring detections covering the same person, e.g. the first two examples in figure 5.5a). This indicates that improved detectors need to have more localised responses (peakier score maps) and/or a different non-maxima suppression strategy. In sections 5.4 and 5.5.1 we explore how to improve the detector localisation.

The annotation errors are mainly missing ignore regions, and a few missing person annotations. In section 5.4 we revisit the Caltech annotations.



(a) Contrast (b) Blur

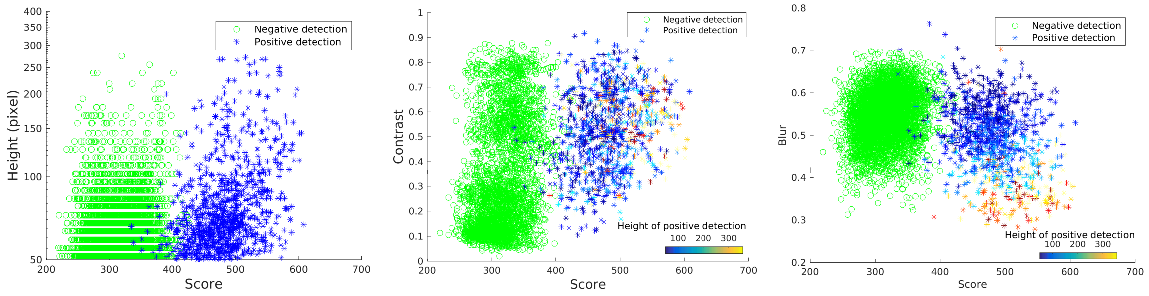
Figure 5.6: Examples for images with different levels of contrast/blur. The number on top of each image indicates the contrast/blur measure.

False negatives. Our clustering results in figure 5.4b show the well known difficulty of detecting small and occluded objects. We hypothesise that low scoring side-view persons and cyclists may be due to a dataset bias, i.e. these cases are under-represented in the training set (most persons are non-cyclist walking on the side-walk, parallel to the car). Augmenting the training set with external images for these cases might be an effective strategy.

To better understand the issue with small pedestrians, we measure size, blur, and contrast for each (true or false) detection. We observe that small persons are commonly saturated (over or under exposed) and blurry, and thus hypothesise that this might be an underlying factor for weak detection (other than simply having fewer pixels to make the decision).

Scale, blur, or contrast? For false negatives, a major source of errors is small scale, but we also find that small pedestrians are often of low contrast or blurred.
To enable our analysis regarding blur and contrast, we define two automated measures. Contrast is measured via the difference between the top and bottom quantiles of the grey scale intensity of the pedestrian patch; the blur is measured as the difference between the input and its blurred patch, which is generated by applying a mean filter on top of the input image (Crete-Roffet *et al.*, 2007). Note all patches are re-scaled to our model size (120 × 60) for blur measure computation. Both of the contrast and blur measures are in [0, 1], where a higher value indicates a higher degree of contrast or blur. Figures 5.6a and 5.6b show pedestrians ranked by our contrast and blur measures. One can observe that our quantitative measures correlate well with human notions of blur and contrast.

In order to investigate the three factors separately, we observe the correlation between size/contrast/blur and score, as shown in figure 5.7. We can see that the overlap between false positive and true positive is equally distributed across different levels of contrast and blur, while for scale, the overlap is quite high at small scale. Thus we conclude that small scale itself is the main factor negatively impacting detection quality and that high blur and low contrast are not.



(a) Size versus score (b) Contrast versus score (c) Blur versus score

Figure 5.7: Correlation between size/contrast/blur and score.

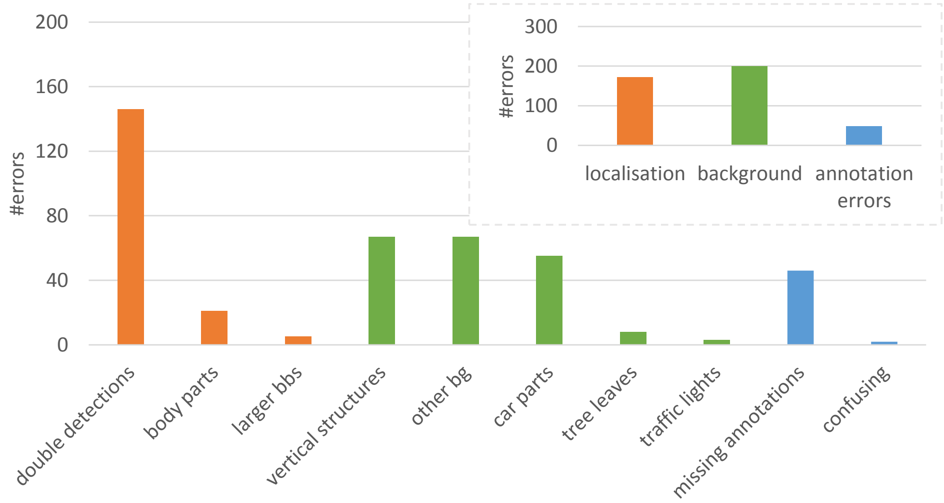


Figure 5.8: False positive sources of `Checkerboards` (Zhang *et al.*, 2015b) on the KITTI validation set.

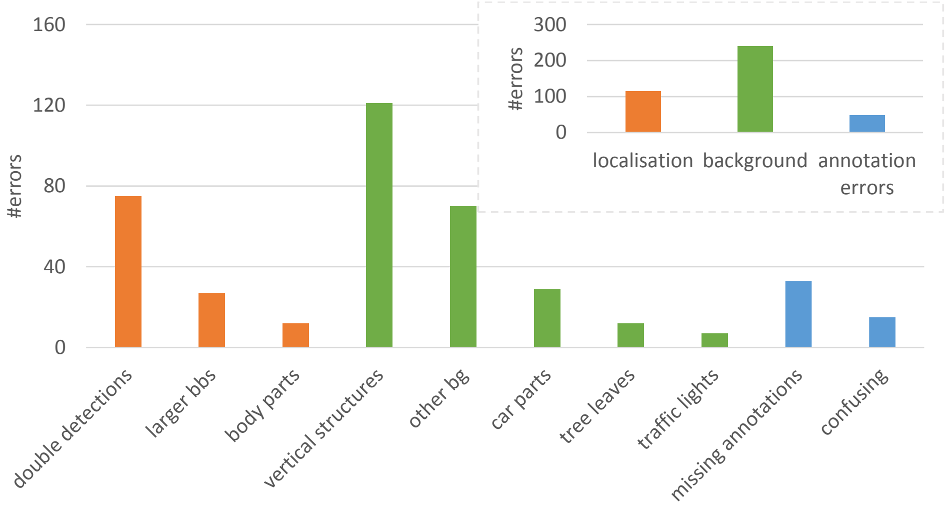


Figure 5.9: False positive sources of RPN+BF on the Caltech test set.

Discussion. In order to verify the universality of the error sources across different datasets and detectors, we implement the same analysis for the `Checkerboards` detector on the KITTI dataset (Geiger *et al.*, 2012) and for another state-of-the-art detector RPN+BF (Zhang *et al.*, 2016a) on the Caltech test set.

While comparing the statistics shown in figure 5.4a, 5.8 and 5.9, we observe similar trends for the error sources, e.g. double detections, vertical structures, annotation errors.

Conclusion. Our analysis shows that false positive errors have well defined sources that can be specifically targeted with the strategies suggested above. A fraction of the false negatives are also addressable, although the small and occluded pedestrians remain a hard and significant problem.

5.3.2.2 *Oracle test cases*

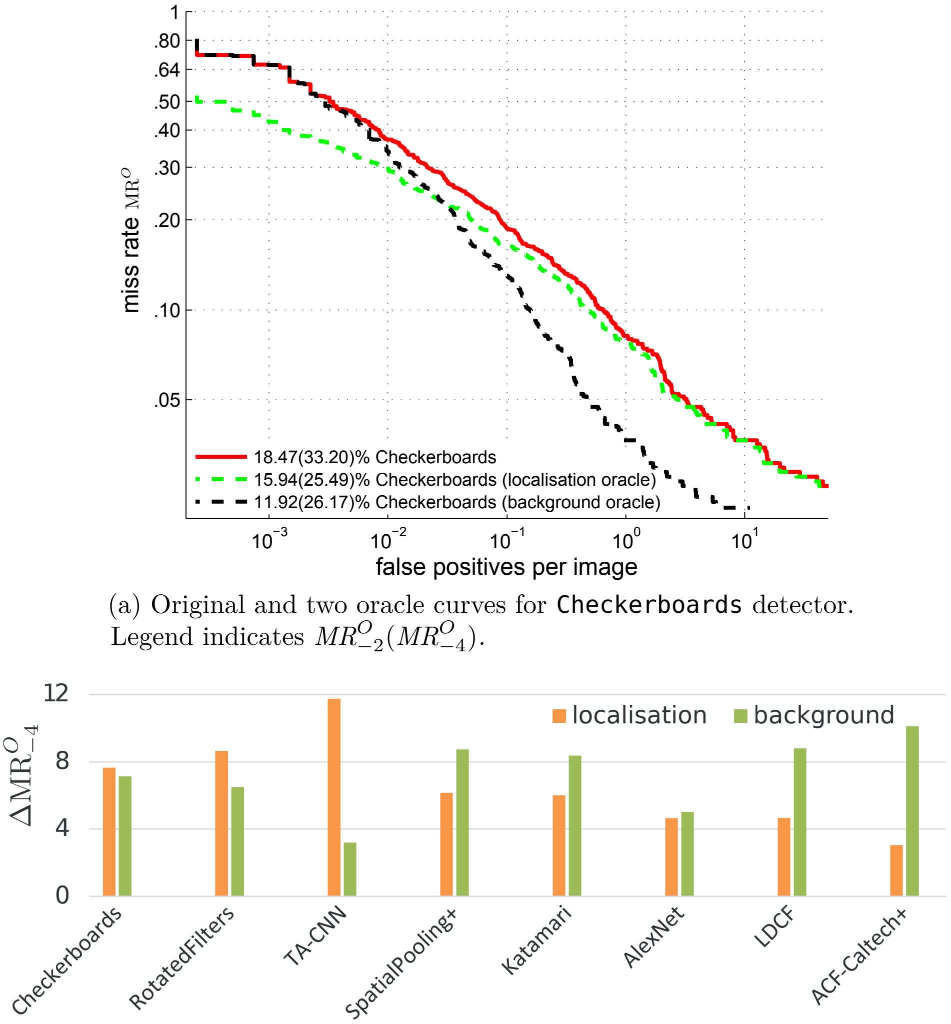
The analysis of section 5.3.2.1 focused on error counts. For area-under-the-curve metrics, such as the ones used for Caltech evaluation, high-scoring errors matter more than low-scoring ones. In this section we directly measure the impact of localisation and background-vs-foreground errors on the detection quality metric (log-average miss-rate) by using oracle test cases.

In the oracle case for localisation, all false positives that overlap with ground truth are ignored for evaluation. In the oracle tests for background-vs-foreground, all false positives that do not overlap with ground truth are ignored.

Figure 5.10a shows that fixing localisation mistakes improves performance in the low FPPI region; while fixing background mistakes improves results in the high FPPI region. In figure 5.10b we show the gains to be obtained in MR^{O}_{-4} terms by fixing localisation or background issues. When comparing the eight top performing methods we find that most methods would boost performance significantly by fixing either problem. It is important to note that localisation and background errors together comprise all false positives. If we remove both types the only mistakes that remain stem from missing recall and the result would be a horizontal line with very low miss rate. However, due to the log-log nature of the numbers, the sum of localisation and background deltas do not add up to the total miss-rate.

We also show some examples of objects with similar scores in figure 5.11. In both low-scoring and high-scoring groups, we can see both pedestrians and background objects, which shows that the detector fails to rank foreground and background adequately.

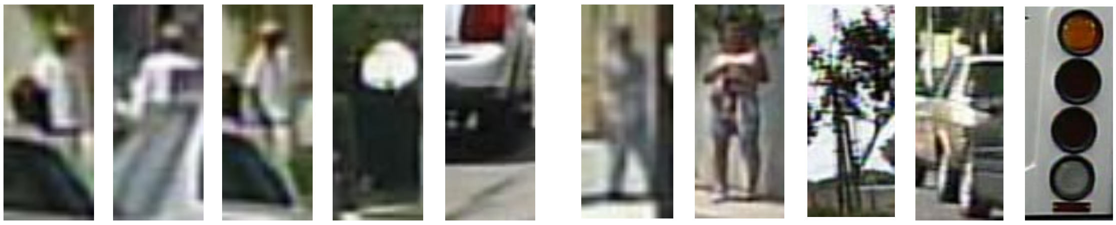
Conclusion. For most top performing methods localisation and background-vs-foreground errors have equal impact on the detection quality.



(a) Original and two oracle curves for `Checkerboards` detector. Legend indicates $MR^O_{-2}(MR^O_{-4})$.

(b) Comparison of miss-rate gain ($\Delta\mathrm{MR}^O_{-4}$) for top performing methods.

Figure 5.10: Oracle cases evaluation over Caltech test set. Both localisation and background-versus-foreground show important room for improvement.



(a) Low-scoring objects

(b) High-scoring objects

Figure 5.11: Failure cases of `Checkerboards` detector (Zhang *et al.*, 2015b). Each group shows image patches of similar scores: some background objects are of high scores; while some persons are of low scores. We aim to understand when the detector fails through analysis.

5.4 Improved Caltech annotations

When evaluating our human baseline and other methods with a strict IoU ≥ 0.8 in figure 5.3, we notice that the performance drops. The original annotation protocol is based on interpolating sparse annotations across multiple frames (Dollár *et al.*, 2012b), and these sparse annotations are not necessarily located on the evaluated frames. After close inspection we notice that this interpolation generates a systematic offset in the annotations. Humans walk with a natural up and down oscillation that is not modelled by the linear interpolation used, thus in most frames have shifted bounding box annotations. This effect is not noticeable when using the forgiving IoU ≥ 0.5 criterion, however such noise in the annotations is problematic when aiming to improve object localisation.

These localisation issues together with the annotation errors detected in section 5.3.2.1 motivated us to create a new set of improved annotations for the Caltech pedestrians dataset. Our aim is two-fold: on one side we want to provide a more accurate evaluation of the state of the art, in particular an evaluation suitable to close the "last 20%" of the problem. On the other side, we want to have high quality training annotations and evaluate how much improved annotations lead to better detections. We evaluate the second aspect in section 5.5.1.

5.4.1 New annotation protocol

Our new annotations are done both on the test and training 1× set, and focus on high quality. The annotators are allowed to look at the full video to decide if a person is present or not, they are requested to mark ignore regions in areas covering crowds, human shapes that are not persons (posters, statues, etc.), and in areas that could not be decided as certainly not containing a person. Each person annotation is done by drawing a line from the top of the head to the point between both feet, the same as human baseline. The annotators must hallucinate head and feet if these are not visible. When the person is not fully visible, they must also annotate a rectangle around the largest visible region. This allows to estimate the occlusion level in a similar fashion as the original annotations. The new annotations do share some bounding boxes with the human baseline (when no correction was needed), thus the human baseline cannot be used to do analysis across different IoU thresholds over the new test set.

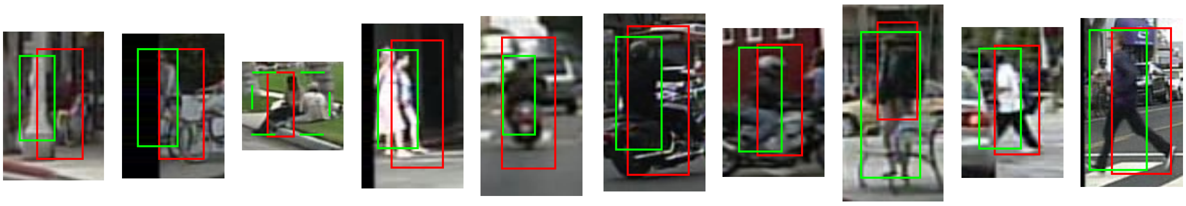
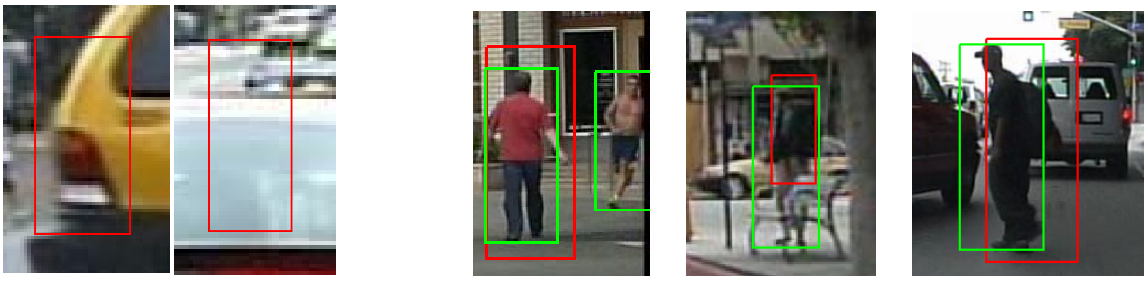


Figure 5.12: Examples of differences between original (red) and new annotations (green). Ignore regions are drawn with dashed lines. These are the top 10 annotations, sorted from smallest to largest IoU between original and new annotations.



(a) False annotations

(b) Poor alignment

Figure 5.13: Examples of errors in original annotations. New annotations in green, original ones in red.

Detector	Training data	Median IoU^O	Median IoU^N
Roerei (Benenson *et al.*, 2013)	INRIA	0.76	*0.84*
RotatedFilters	Orig. 10×	*0.80*	0.77
RotatedFilters	New 10×	0.76	*0.85*

Table 5.3: Median IoU of true positives for detectors trained on different data, evaluated on original and new Caltech test. Models trained on INRIA align well with our new annotations, confirming that they are more precise than previous ones.

In summary, our new annotations differ from the human baseline in the following aspects: both training and test sets are annotated, ignore regions and occlusions are also annotated, full video data is used for decision, and multiple revisions of the same image are allowed.

After creating a full independent set of annotations, we consolidated the new annotations by cross-validating with the old annotations. Any correct old annotation not accounted for in the new set, was added too.

We show some examples of differences between original and new annotations in figure 5.12. Our new annotations correct several types of errors in the existing annotations, such as misalignments (figure 5.13b), missing annotations (false negatives), false annotations (false positives, figure 5.13a), and the inconsistent use of "ignore" regions.

5.4.2 Better alignment

Figure 5.14 and table 5.3 show quantitative evidence that our new annotations are more precisely localised than the original ones.

In table 5.3 we summarise the alignment quality of a detector via the median IoU between true positive detections and a given set of annotations. When evaluating with the original annotations ("median IoU^O" column in table 5.3), only the model trained with original annotations has good localisation. However, when evaluating with the new annotations ("median IoU^N" column) *both* the model trained on INRIA data, and on the new annotations reach high localisation accuracy. This indicates that our new annotations are indeed better aligned, just as INRIA annotations are better aligned than Caltech.

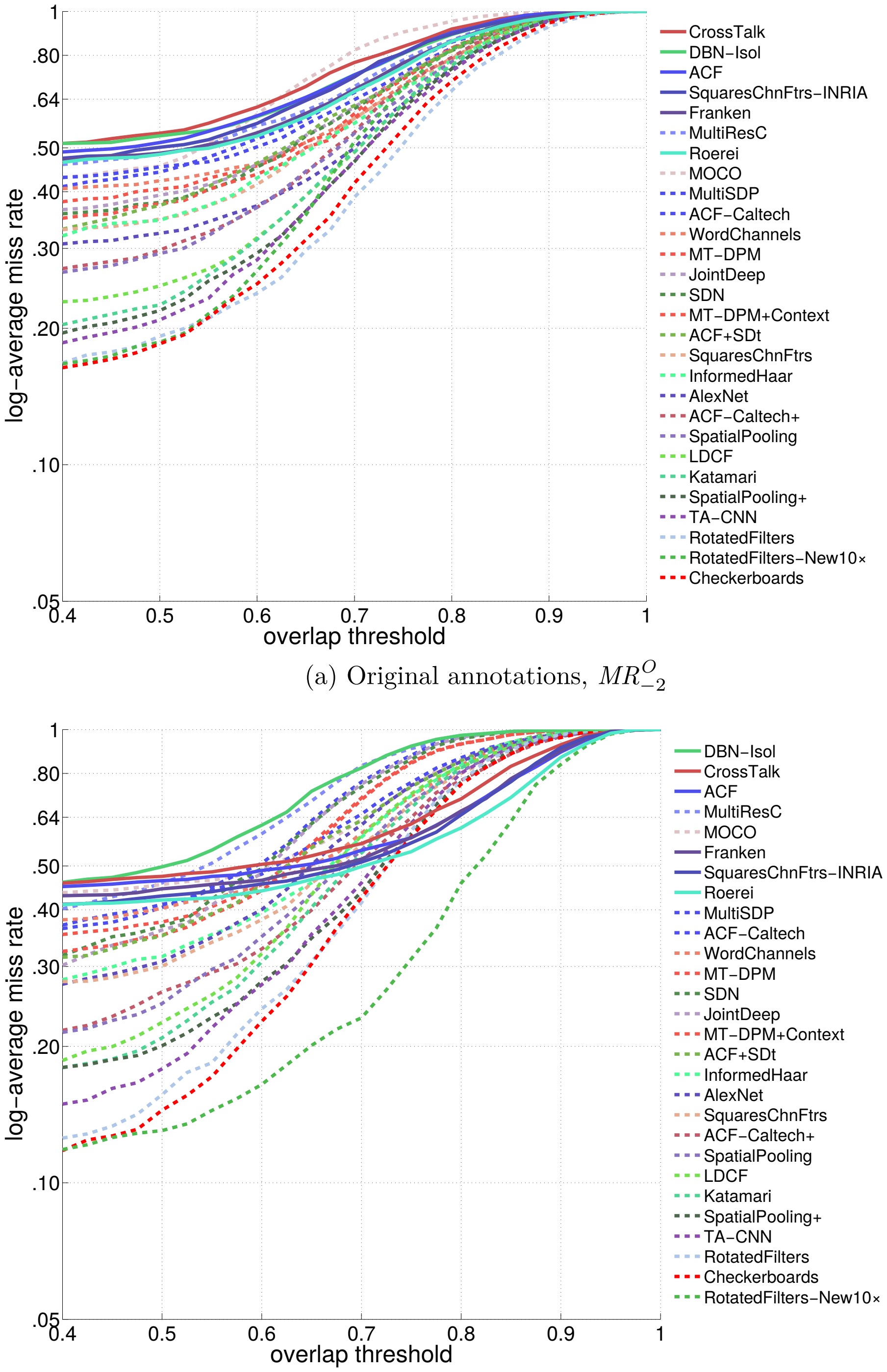
MR versus IoU. Figure 5.14 provides more details about table 5.3. It plots $\mathrm{MR}^{\mathrm{O}}_{-2}$ and $\mathrm{MR}^{\mathrm{N}}_{-2}$ of top performing methods versus the overlap criterion for accepting detections as true positives (IoU threshold). The standard evaluation uses IoU threshold 0.5. On these plots methods trained on INRIA have continuous lines, methods trained on Caltech dashed ones (see also figure 5.15).

In figure 5.14a (original annotations) the ranking of the methods remains stable as the overlap threshold becomes stricter (consistent with the observations in Dollár *et al.* (2012b)). Interestingly, we observe a different trend in figure 5.14b, where all methods are evaluated on new annotations ($\mathrm{MR}^{\mathrm{N}}_{-2}$). Those methods trained on INRIA, albeit having a poor performance at IoU $= 0.5$, perform comparatively well at higher IoU, eventually overpassing all methods trained on original Caltech data. We attribute this to the fact that INRIA training data is of better quality (better aligned training samples), and thus the detectors have learnt to localise better.

This difference in trend between original and new annotations confirms that our improved annotations are better with respect to localisation. Table 5.3 summarises this observation. Section 5.5.1 describes the `RotatedFilters-New10`× entry.

5.4.3 Ranking

Figure 5.15 presents the ranking of all published Caltech methods when evaluated on $\mathrm{MR}^{\mathrm{N}}_{-2}$ (proposed new annotations). Although there are a few changes in ranking (e.g. `JointDeep` versus `SDN`) compared to MR^{O}_{-2} (original annotations) evaluation, the overall trend is preserved. This is a good sign that the improved annotations are not a radical departure from previous ones. As discussed beforehand, improved annotations matter most for future methods (going further down in MR) and for the low FPPI region of the curves (high confidence mistakes).



(a) Original annotations, MR^{O}_{-2}

(b) New annotations, MR^{N}_{-2}

Figure 5.14: Plot of log-average miss rate versus overlap threshold (IoU) for the top-performing methods on the "reasonable" experimental setting. While evaluated on the new annotations, methods trained on INRIA (represented with solid curves) behave better than methods trained on Caltech original annotations when we apply a stricter overlap criterion.

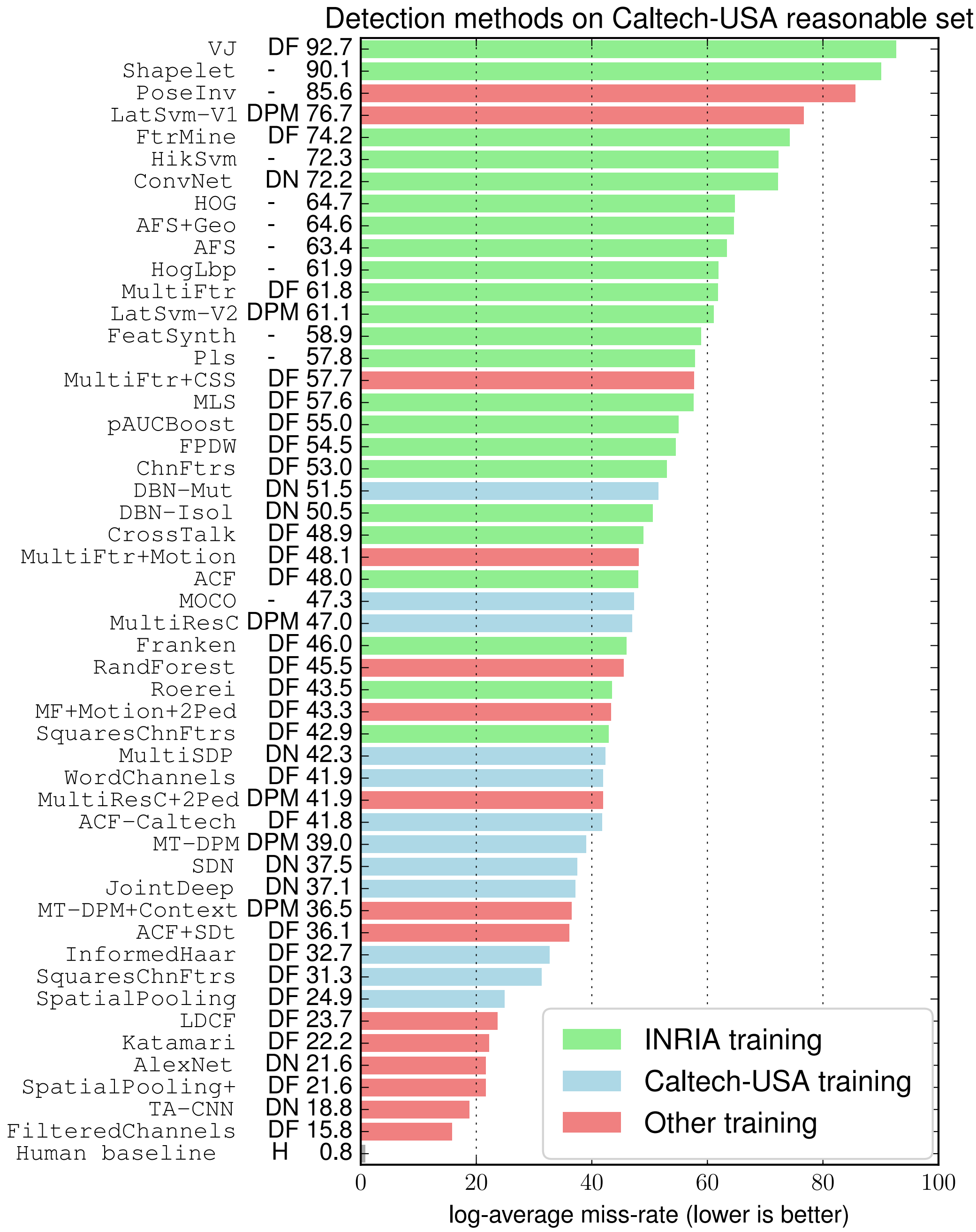


Figure 5.15: Ranking of Caltech methods (CVPR 2015 snapshot) on new annotations MR^N_{-2}. DF: decision forest, DPM: deformable parts model, DN: deep network.

5.5 Improving the state of the art

In this section we leverage the insights of the analysis, to improve localisation and background-versus-foreground discrimination of our baseline detector.

5.5.1 Impact of training annotations

With new annotations at hand we want to understand what is the impact of annotation quality on detection quality. We will train `ACF` (Dollár *et al.*, 2014) and `RotatedFilters` models using different training sets and evaluate on both original and new annotations (i.e. MR^O_{-2}, MR^O_{-4} and MR^N_{-2}, MR^N_{-4}). The above two detectors are selected for the subsequent experiments as they are trained via boosting and thus inherently sensitive to annotation noise.

Pruning benefits. Table 5.4 shows results when trained with original, new and pruned annotations (using a $5/6 + 1/6$ training and validation split of the full training set). As expected, models trained on original/new and tested on original/new perform better than training and testing on different annotations. To understand better what the new annotations bring to the table, we build a hybrid set of annotations. Pruned annotations is a mid-point that allows to decouple the effects of removing errors and improving alignment.

Pruned annotations are generated by matching new and original annotations (IoU $\geq$ 0.5), marking as ignore region any original annotation absent in the new ones, and adding any new annotation absent in the original ones.

From original to pruned annotations the main change is removing annotation errors, from pruned to new, the main change is better alignment. From table 5.4 both `ACF` and `RotatedFilters` benefit from removing annotation errors, even in MR^O_{-2}. This indicates that our new training set is better sanitised than the original one.

We see in MR^N_{-2} that the stronger detector benefits more from better data, and that the largest gain in detection quality comes from removing annotation errors.

Detector	Anno. variant	MR^O_{-2}	MR^N_{-2}
`ACF` (Dollár *et al.*, 2014)	Original	*36.90*	40.97
	Pruned	36.41	35.62
	New	41.29	*34.33*
`RotatedFilters`	Original	*28.63*	33.03
	Pruned	23.87	25.91
	New	31.65	*25.74*

Table 5.4: Effects of different training annotations on detection quality on validation set (1× training set). Italic numbers have matching training and test sets. Both detectors improve on the original annotations, when using the “pruned” variant (see §5.5.1).

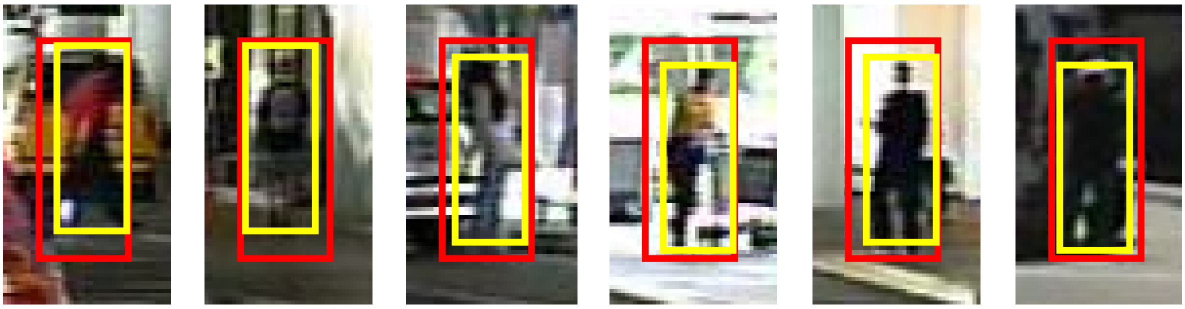


Figure 5.16: Examples of automatically aligned ground truth annotations. Red/yellow→ before/after alignment.

1× data	10× data aligned with	MR^O_{-2} (MR^O_{-4})	MR^N_{-2} (MR^N_{-4})
Orig.	Ø	19.20 (34.28)	17.22 (31.65)
Orig.	Orig. 10×	19.16 (32.28)	15.71 (28.13)
Orig.	New ½×	16.97 (28.01)	14.54 (25.06)
New	New 1×	16.77 (29.76)	12.96 (22.20)

Table 5.5: Detection quality of `RotatedFilters` on test set when using different aligned training sets. All models trained with Caltech 10×, composed with different $1\times + 9\times$ combinations.

Alignment benefits. The detectors from the ICF family benefit from training with increased training data (Nam *et al.*, 2014; Zhang *et al.*, 2015b), using 10× data is better than 1× (see section 5.2.1). To leverage the 9× remaining data using the new 1× annotations we train a model over the new annotations and use this model to re-align the original annotations over the 9× portion. Because the new annotations are better aligned, we expect this model to be able to recover slight position and scale errors in the original annotations. Figure 5.16 shows example results of this process.

Table 5.5 reports results using the automatic alignment process, and a few degraded cases: using the original 10×, self-aligning the original 10× using a model trained over original 10×, and aligning the original 10× using only a fraction of the new annotations (without replacing the 1× portion). The results indicate that using a detector model to improve overall data alignment is indeed effective, and that better aligned training data leads to better detection quality (both in MR^O and MR^N). This is in line with the analysis of section 5.3.2. Already using a model trained on ½ of the new annotations for alignment, leads to a stronger model than obtained when using original annotations.

We name the `RotatedFilters` model trained using the new 1× annotations and the aligned 9× data, `RotatedFilters-New10`×. This model also reaches high median true positives IoU in table 5.3, indicating that indeed it obtains more precise detections at test time.

Test proposals	Proposal	+AlexNet	+VGG	+bbox reg & NMS
ACF (Dollár *et al.*, 2014)	48.0%	28.5%	22.8%	20.8%
SquaresChnFtrs (Benenson *et al.*, 2014)	31.3%	21.2%	15.9%	14.7%
LDCF (Nam *et al.*, 2014)	23.7%	21.6%	16.0%	13.7%
RotatedFilters	17.2%	21.5%	17.8%	13.8%
Checkerboards (Zhang *et al.*, 2015b)	16.1%	21.0%	15.3%	11.1%
RotatedFilters-New10×	13.0%	17.2%	11.7%	10.0%

Table 5.6: Detection quality of convnets with different proposals. Grey numbers indicate worse results than the input proposals. All numbers are MR^N_{-2} on the Caltech test set. The last column indicates bounding box regression followed by a second non-maximum suppression is applied after VGG re-scoring.

Conclusion. Using high quality annotations for training improves the overall detection quality, thanks both to improved alignment and to reduced annotation errors.

5.5.2 Convnets for pedestrian detection

The results of section 5.3.2 indicate that there is room for improvement by focusing on the core background versus foreground discrimination task (the "classification part of object detection"). Chapter 4 and recent work (Tian *et al.*, 2015b) showed competitive performance with convolutional neural networks (convnets) for pedestrian detection. We include convnets into our analysis, and explore to what extent performance is driven by the quality of the detection proposals.

AlexNet and VGG. We consider two convnets. 1) The AlexNet from chapter 4, and 2) The VGG16 model from Girshick (2015). Both are pre-trained on ImageNet and fine-tuned over Caltech 10× (original annotations) using SquaresChnFtrs proposals. Both networks are based on open source, and both are instances of the R-CNN framework (Girshick *et al.*, 2014). Albeit their training/test time architectures are slightly different (R-CNN versus Fast R-CNN), we expect the result differences to be dominated by their respective discriminative power (VGG16 improves 8 pp in mAP over AlexNet in the Pascal detection task (Girshick *et al.*, 2014)).

Table 5.6 shows that as the quality of the detection proposals improves, AlexNet fails to provide a consistent gain, eventually worsening the results of our ICF detectors (similar observation in chapter 4). Similarly VGG provides large gains for weaker proposals, but as the proposals improve, the gain from the convnet re-scoring eventually stalls.

After closer inspection of the resulting curves, we notice that both AlexNet and VGG push background instances to lower scores, and at the same time generate a large number of high scoring false positives. This observation motivates us to have a look at the distribution of proposals. We then find that ICF detectors are able to provide proposals with high recall, but at a price of introducing a lot of false positives around

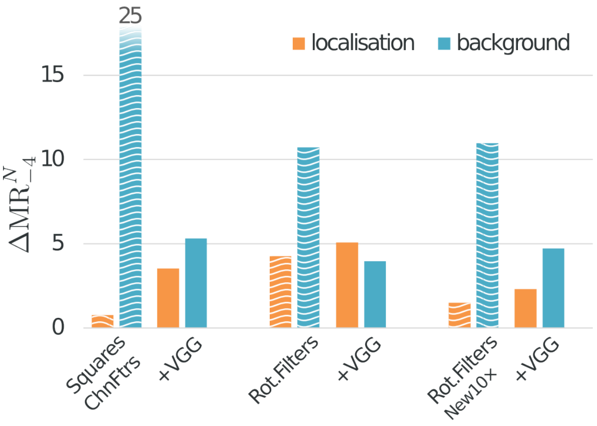


Figure 5.17: Oracle case analysis of proposals + convnets (after second NMS). Miss-rate gain, ΔMR^{O}_{-4}. The convnet significantly improves background errors, while slightly increasing localisation ones.

Detector aspect	MR^{O}_{-2} (MR^{O}_{-4})	MR^{N}_{-2} (MR^{N}_{-4})
`Checkerboards`	18.47 (33.20)	15.81 (28.57)
`RotatedFilters`	19.20 (34.28)	17.22 (31.65)
+ Alignment §5.5.1	16.97 (28.01)	14.54 (25.06)
+ New annotations §5.5.1	16.77 (29.76)	12.96 (22.20)
+ VGG §5.5.2	16.61 (34.79)	11.74 (28.37)
+ bbox reg & NMS	*14.16 (28.39)*	*10.00 (20.77)*

Table 5.7: Step by step improvements from previous best method `Checkerboards` to `RotatedFilters-New10x+VGG`.

the objects (see figure 5.18). However convnets have difficulties giving low scores to these windows surrounding the true positives. In other words, despite their fine-tuning, the convnet score maps are “blurrier” than the proposal ones. We hypothesise this is an intrinsic limitation of the AlexNet and VGG architectures, due to their internal feature pooling. Obtaining “peakier” responses from a convnet most likely will require using rather different architectures, possibly more similar to the ones used for semantic labelling or boundaries estimation tasks which require pixel-accurate output.

Fortunately, we can compensate for the lack of spatial resolution in the convnet scoring by using bounding box regression. Adding bounding regression over VGG, and applying a second round of non-maximum suppression (first NMS on the proposals, second on the regressed boxes), has the effect of “contracting the score maps”. Neighbour proposals that before generated multiple strong false positives, now collapse into a single high scoring detection. We use the usual IoU ≥ 0.5 merging criterion for the second NMS.

The last column of table 5.6 shows that bounding box regression + NMS is effective at providing an additional gain over the input proposals, even for our best detector `RotatedFilters-New10×`. On the original annotations `RotatedFilters-`

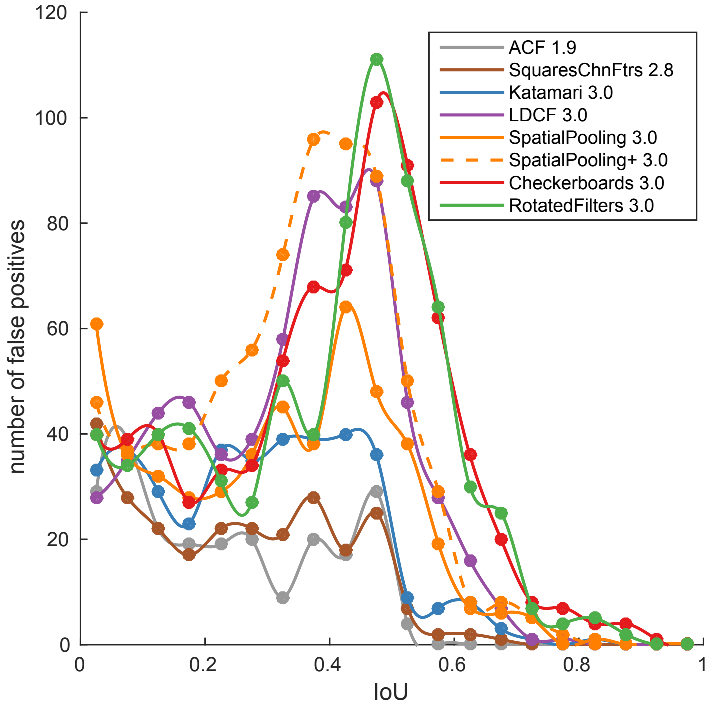


Figure 5.18: Distribution of overlap between false positives and ground truth, for different ICF detectors. The curves are histograms with coarse IoU bins (0 overlap case omitted). Number in the legend indicates the average number of proposals per image (after filtering to reach ~3 proposals per image on average). Note that most detectors have many false positives nearby true detections.

`-New10`×`+VGG` reaches 14.2% MR^{O}_{-2}, which improves over chapter 4 and Tian *et al.* (2015b). The comparison with other state-of-the-art detectors are shown in figure 5.19.

Our best performing detector `RotatedFilters-New10`×`+VGG` runs on a 640 × 480 image for ~3.5 seconds, including the ICF sliding window detection and VGG re-scoring. Training times are counted 1~2 days for the `RotatedFilters` detector, and 1~2 days for VGG fine-tuning. We compare the runtime versus performance for different detectors in table 5.8. All detectors are tested on the same hardware: Intel Xeon E5-2680 2.70GHz CPU; and Tesla K40 GPU. Although `RotatedFilters-New10x+VGG` runs slower than previous ICF detectors, it reduces the errors by a large margin.

Figure 5.17 repeats the oracle tests of section 5.3.2.2 over our convnet results. We make comparisons for three convnet detectors and their corresponding ICF proposal methods, to observe how localisation and background errors change after VGG re-scoring. One can see that for each proposal method, VGG significantly cuts down the background errors, while at the same time slightly increases the localisation errors. These comparisons verify that convnets are of strong discriminative ability against background objects, but on the other hand also demonstrate that convnets fail to shrink down off-aligned false positives around the true detections.

	Runtime (seconds)			MR_{-2}^{N}
	CPU	GPU	Total	
ACF	0.1	/	0.1	27.6
`Checkerboards`	3.0	/	3.0	15.8
`RotatedFilters-New10x`	2.5	/	2.5	13.0
`RotatedFilters-New10x+VGG`	2.5	1.0	3.5	10.0

Table 5.8: Comparison of runtime versus performance for different detectors on the Caltech benchmark. Runtime is the average test time on one 640 × 480 image.

Conclusion. Although convnets have strong results in image classification and general object detection, they seem to have limitations when producing well localised detection scores around small objects. Bounding box regression (and NMS) is a key ingredient to side-step this limitation with current architectures. Even after using a strong convnet, background-versus-foreground remains the main source of errors; suggesting that there is still room for improvement on the raw classification power of the neural network.

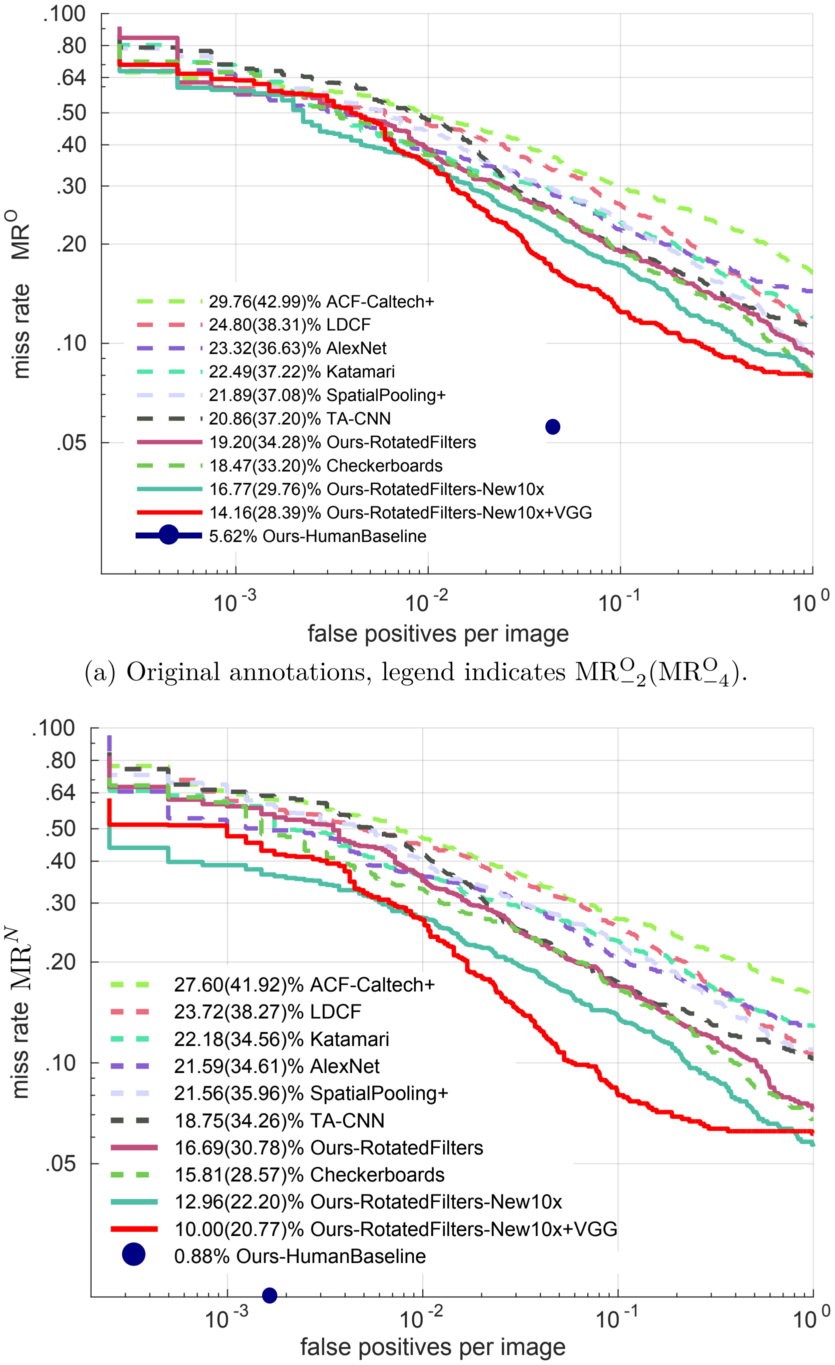
5.6 Conclusion

In this chapter, we make great efforts on analysing the failures for a top-performing detector on Caltech and KITTI datasets. Via our human baseline we have quantified a lower bound on how much improvement there is to be expected. There is a 10× gap in terms of errors still to be closed. To better measure the next steps in detection progress, we have provided new sanitised Caltech training and test set annotations.

Our failure analysis of a top performing method has shown that most of its mistakes are well characterised. The error characteristics lead to specific suggestions on how to engineer better detectors (mentioned in section 5.3.2; e.g. data augmentation for side-view persons, or extending the detector receptive field along the vertical axis).

We have partially addressed some of the issues by measuring the impact of better annotations on localisation accuracy, and by investigating the use of convnets to improve the background to foreground discrimination. Our results indicate that significantly better alignment can be achieved with properly trained ICF detectors, and that, for pedestrian detection, convnet struggle with localisation issues, which can be partially addressed via bounding box regression. Both on original and new annotations, the described detection approach reaches top performance, see progress in table 5.7.

We hope the insights and data provided in this work will guide the path to close the gap between machines and humans in the pedestrian detection task.



(a) Original annotations, legend indicates $\text{MR}^{\text{O}}_{-2}(\text{MR}^{\text{O}}_{-4})$.

(b) New annotations, legend indicates $\text{MR}^{\text{N}}_{-2}(\text{MR}^{\text{N}}_{-4})$.

Figure 5.19: Performance of top detectors evaluated on original and new annotations.

Part II

GENERAL OBJECT DETECTION

General object class detection faces similar challenges as pedestrian detection (Part I). Yet general objects typically have significantly different statistics in several aspects. While the vast majority of pedestrians are upright walking people and thus have a fixed aspect ratio, other object classes can have a very large range of of aspect ratios, which increases the search space of magnitudes. Many object classes exhibit diverse geometries and appearances, much larger than the articulation of pedestrians. Further, pedestrian detection benchmarks often focus on the autonomous driving scenario as this a major application. The scene geometric statistics of a camera in a driving car are a bias that is not present in general object detection benchmark, which consist of unconstrained photo collections, where the main bias is how humans compose photographs and what they like to photograph. For these reasons, not all insights and techniques transfer well between pedestrian detection and general object detection.

To address speed concerns it is necessary to either speed up the classifier or reduce the search space. A common technique to cut down the search space are detection proposals, which we analyse in chapter 6.

In chapters 7 and 8 we present work on learning non-maximum suppression (NMS) with neural networks. NMS is a ubiquitous post processing step in almost all detectors (both general object detectors and pedestrian detectors) that removes redundant detections that belong to the same object. In fact it is *so* ubiquitous and essential to how we think about a detector, publications typically do not even mention that they use it.

What makes for effective detection proposals? 6

Current top performing object detectors employ detection proposals to guide the search for objects, thereby avoiding exhaustive sliding window search across images. Despite the popularity and widespread use of detection proposals, it is unclear which trade-offs are made when using them during object detection. We provide an in-depth analysis of twelve proposal methods along with four baselines regarding proposal repeatability, ground truth annotation recall on PASCAL, ImageNet, and COCO, and their impact on DPM, R-CNN, and Fast R-CNN detection performance. Our analysis shows that for object detection improving proposal localisation accuracy is as important as improving recall. We introduce a novel metric, the average recall (AR), which rewards both high recall and good localisation and correlates surprisingly well with detection performance. Our findings show common strengths and weaknesses of existing methods, and provide insights and metrics for selecting and tuning proposal methods.

The work has been published at BMVC (Hosang *et al.*, 2014) with an oral presentation and later revised and published at PAMI (Hosang *et al.*, 2016a). Jan Hosang was the lead author and contributed all experiments.

6.1 Introduction

Until recently, the most successful approaches to object detection utilised the well known “sliding window” paradigm (Papageorgiou and Poggio, 2000; Viola and Jones, 2004; Felzenszwalb *et al.*, 2010), in which a computationally efficient classifier tests for object presence in every candidate image window. Sliding window classifiers scale linearly with the number of windows tested, and while single-scale detection requires classifying

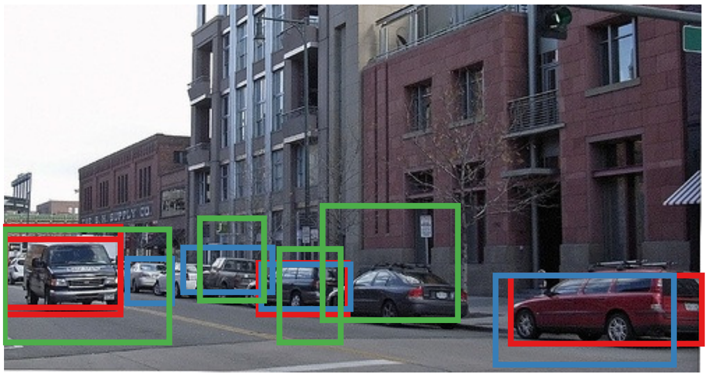


Figure 6.1: What makes object detection proposals effective?

around $10^4 - 10^5$ windows per image, the number of windows grows by an order of magnitude for multi-scale detection. Modern detection datasets (Everingham *et al.*, 2014; Deng *et al.*, 2009; Lin *et al.*, 2014) also require the prediction of object aspect ratio, further increasing the search space to $10^6 - 10^7$ windows per image.

The steady increase in complexity of the core classifiers has led to improved detection quality, but at the cost of significantly increased computation time per window (Wang *et al.*, 2013; Cinbis *et al.*, 2013; Girshick *et al.*, 2014; He *et al.*, 2014; Szegedy *et al.*, 2015b). One approach for overcoming the tension between computational tractability and high detection quality is through the use of "detection proposals" (Alexe *et al.*, 2010; Carreira and Sminchisescu, 2010; Endres and Hoiem, 2010; van de Sande *et al.*, 2011). Under the assumption that all objects of interest share common visual properties that distinguish them from the background, one can design or train a method that, given an image, outputs a set of proposal regions that are likely to contain objects. If high object recall can be reached with considerably fewer windows than used by sliding window detectors, significant speed-ups can be achieved, enabling the use of more sophisticated classifiers.

At the time of early 2015, top performing object detectors for PASCAL (Everingham *et al.*, 2014) and ImageNet (Deng *et al.*, 2009) all use detection proposals (Wang *et al.*, 2013; Girshick *et al.*, 2014; Szegedy *et al.*, 2015b; He *et al.*, 2014; Cinbis *et al.*, 2013; Girshick, 2015). In addition to allowing for use of more sophisticated classifiers, the use of detection proposals alters the data distribution that the classifiers handle. This may also improve detection quality by reducing spurious false positives.

Most papers on generating detection proposals perform fairly limited evaluations, comparing results using only a subset of metrics, datasets, and competing methods. In this work, we aim to revisit existing work on proposals and compare most publicly available methods in a unified framework. While this requires us to carefully re-examine the metrics and settings for evaluating proposals, it allows us to better understand the benefits and limitations of current methods.

The contributions of this chapter are as follows:

- In section 6.2, we provide a systematic overview of detection proposal methods and define simple baselines that serve as reference points. We discuss the taxonomy of proposal methods, and describe commonalities and differences of the various approaches.

- In section 6.3, we introduce the notion of proposal repeatability, discuss its relevance when considering proposals for detection, and measure the repeatability of existing methods. The results are somewhat unexpected.

- In section 6.4, we study object recall on the PASCAL VOC 2007 test set (Everingham *et al.*, 2014), and for the first time, over the larger and more diverse ImageNet 2013 (Deng *et al.*, 2009) and MS COCO 2014 (Lin *et al.*, 2014) validation sets. The latter allows us to examine possible biases towards PASCAL objects categories. Overall, these experiments are substantially broader in scope than previous work, both in the number of methods evaluated and datasets used.

- In section 6.5, we evaluate the influence of different proposal methods on DPM (Felzenszwalb *et al.*, 2010), R-CNN (Girshick *et al.*, 2014), and Fast R-CNN (Girshick, 2015) detection performance. Based on our results, we introduce a novel evaluation metric, the average recall (AR). We show that AR is highly correlated with detector performance, more so than previous metrics, and we advocate AR to become the standard metric for evaluating proposals. Our experiments provide the first clear guidelines for selecting and tuning proposal methods for object detection.

All evaluation scripts and method bounding boxes used in this work are publicly available to facilitate the reproduction of our evaluation[10]. The results presented in this chapter summarise results of over 500 experiments on multiple data sets and required multiple months of CPU time.

6.2 Detection proposal methods

Detection proposals are similar in spirit to interest point detectors (Tuytelaars and Mikolajczyk, 2008; Mikolajczyk *et al.*, 2005). Interest points allow for focusing attention to the most salient and distinctive locations in an image, greatly reducing computation for subsequent tasks such as classification, retrieval, matching, and detection. Likewise, object proposals considerably reduce computation compared to the dense (sliding window) detection framework by generating candidate proposals that may contain objects. This in turn enables use of expensive classifiers per window (Wang *et al.*, 2013; Girshick *et al.*, 2014; Szegedy *et al.*, 2015b; He *et al.*, 2014; Cinbis *et al.*, 2013).

It is worthwhile noting that interest points were dominant when computing feature descriptors densely was prohibitive. However, with improved algorithmic efficiency and increased computational power, it is now standard practice to use dense feature extraction (Tuytelaars, 2010). The opposite trend has occurred in object detection, where the dense sliding window framework has been overtaken by use of proposals. We aim to understand if detection proposals improve detection accuracy or if their use is strictly necessary for computational reasons. While in this work we focus on the impact of proposals on detection, proposals have applications beyond object detection, as we discuss in section 6.6.

Two general approaches for generating object proposals have emerged: *grouping methods* and *window scoring methods*. These are perhaps best exemplified by the early and well known `SelectiveSearch` (van de Sande *et al.*, 2011) and `Objectness` (Alexe *et al.*, 2010) proposal methods. We survey these approaches in section 6.2.1 and section 6.2.2, followed by an overview of alternate approaches in section 6.2.3 and baselines in section 6.2.4. Finally, we consider the connection between proposals and cascades in section 6.2.5 and provide additional method details in section 6.2.6.

The survey that follows is meant to be exhaustive. However, for the purpose of our evaluations, we only consider methods for which source code is available. We cover a diverse set of methods (in terms of quality, speed, and underlying approach). Table

[10] Project page: `http://goo.gl/uMhkAs`

Method	Approach	Outputs Segments	Outputs Score	Control #proposals	Time (sec.)	Repea-tability	Recall Results	Detection Results
`Bing` (Cheng *et al.*, 2014)	Window scoring		✓	✓	0.2	⋆⋆⋆	⋆	·
`CPMC` (Carreira and Sminchisescu, 2012)	Grouping	✓	✓	✓	250	-	⋆⋆	⋆
`EdgeBoxes` (Zitnick and Dollár, 2014)	Window scoring		✓	✓	0.3	⋆⋆	⋆⋆⋆	⋆⋆⋆
`Endres` (Endres and Hoiem, 2014)	Grouping	✓	✓	✓	100	-	⋆⋆⋆	⋆⋆
`Geodesic` (Krähenbühl and Koltun, 2014)	Grouping	✓		✓	1	⋆	⋆⋆⋆	⋆⋆
`MCG` (Arbeláez *et al.*, 2014)	Grouping	✓	✓	✓	30	⋆	⋆⋆⋆	⋆⋆⋆
`Objectness` (Alexe *et al.*, 2012)	Window scoring		✓	✓	3	·	⋆	·
`Rahtu` (Rahtu *et al.*, 2011)	Window scoring		✓	✓	3	·	·	⋆
`RandomizedPrim's` (Manén *et al.*, 2013)	Grouping	✓		✓	1	⋆	⋆	⋆⋆
`Rantalankila` (Rantalankila *et al.*, 2014)	Grouping	✓		✓	10	⋆⋆	·	⋆⋆
`Rigor` (Humayun *et al.*, 2014)	Grouping	✓		✓	10	⋆	⋆⋆	⋆⋆
`SelectiveSearch` (Uijlings *et al.*, 2013)	Grouping	✓	✓	✓	10	⋆⋆	⋆⋆⋆	⋆⋆⋆
`Gaussian`				✓	0	·	·	⋆
`SlidingWindow`				✓	0	⋆⋆⋆	·	·
`Superpixels`		✓			1	⋆	·	·
`Uniform`				✓	0	·	·	·

Table 6.1: Comparison of different detection proposal methods. Grey check-marks indicate that the number of proposals is controlled by indirectly adjusting parameters. Repeatability, quality, and detection rankings are provided as rough summary of the experimental results: "-" indicates no data, "·", "⋆", "⋆⋆", "⋆⋆⋆" indicate progressively better results. These guidelines were obtained based on experiments presented in sections §6.3, §6.4, and §6.5, respectively.

6.1 gives an overview of the 12 selected methods (plus 4 baselines).[11] Table 6.1 also indicates high level information regarding the output of each method and a qualitative overview of the results of the evaluations performed in the remainder of this chapter.

In this chapter we concentrate on class-agnostic proposals for single-frame, bounding box detection. For proposal methods that output segmentations instead of bounding boxes, we convert the output to bounding boxes for the purpose of our evaluation. Methods that operate on videos and require temporal information (e.g. Fragkiadaki *et al.*, 2015) are considered outside the scope of this work.

6.2.1 Grouping proposal methods

Grouping proposal methods attempt to generate multiple (possibly overlapping) segments that are likely to correspond to objects. The simplest such approach would be to directly use the output of any hierarchical image segmentation algorithm, e.g. Gu *et al.* (2009) use the segmentation produced by gPb (Arbeláez *et al.*, 2011). To increase the number of candidate segments, most methods attempt to diversify such hierarchies, e.g. by using multiple low level segmentations (Carreira and Sminchisescu, 2012; Uijlings *et al.*, 2013; Manén *et al.*, 2013) or starting with an over-segmentation and randomising the merge process (Manén *et al.*, 2013). The decision to merge segments is typically based on a diverse set of cues including superpixel shape, appearance cues, and boundary estimates (typically obtained from Arbeláez *et al.* (2011) or Dollár and Zitnick (2015)).

We classify grouping methods into three types according to how they generate proposals. Broadly speaking, methods generate region proposals by grouping superpixels (SP), often using Felzenszwalb and Huttenlocher (2004), solving multiple graph cut (GC) problems with diverse seeds, or directly from edge contours (EC), e.g. from Arbeláez *et al.* (2011) or Dollár and Zitnick (2015). In the method descriptions below the type of each method is marked by SP, GC, or EC accordingly.

We note that while all the grouping approaches have the strength of producing a segmentation mask of the object, we evaluate only the enclosing bounding box proposals.

- **`SelectiveSearch`**$^{\dagger SP}$ (van de Sande *et al.*, 2011; Uijlings *et al.*, 2013) greedily merges superpixels to generate proposals. The method has no learned parameters, instead features and similarity functions for merging superpixels are manually designed. `SelectiveSearch` has been broadly used as the proposal method of choice by many state-of-the-art object detectors, including the R-CNN and Fast R-CNN detectors (Girshick *et al.*, 2014; Girshick, 2015).

- **`RandomizedPrim's`**$^{\dagger SP}$ (Manén *et al.*, 2013) uses similar features as `SelectiveSearch`, but introduces a randomised superpixel merging process in which all probabilities have been learned. Speed is substantially improved.

- **`Rantalankila`**$^{\dagger SP}$ (Rantalankila *et al.*, 2014) proposes a superpixel merging strategy similar to `SelectiveSearch`, but using different features. In a subsequent stage,

[11] We mark the evaluated methods with a '†' in the following listing.

the generated segments are used as seeds for solving graph cuts in the spirit of `CPMC` (see below) to generate more proposals.

- **`Chang`**^{SP} (Chang *et al.*, 2011) combines saliency and `Objectness` with a graphical model to merge superpixels into figure/background segmentations.
- **`CPMC`**$^{\dagger GC}$ (Carreira and Sminchisescu, 2010, 2012) avoids initial segmentations and computes graph cuts with several different seeds and unaries directly on pixels. The resulting segments are ranked using a large pool of features.
- **`Endres`**$^{\dagger GC}$ (Endres and Hoiem, 2010, 2014) builds a hierarchical segmentation from occlusion boundaries and solves graph cuts with different seeds and parameters to generate segments. The proposals are ranked based on a wide range of cues and in a way that encourages diversity.
- **`Rigor`**$^{\dagger GC}$ (Humayun *et al.*, 2014) is a somewhat improved variant of `CPMC` that speeds computation considerably by re-using computation across multiple graph-cut problems and using the fast edge detectors from Lim *et al.* (2013) and Dollár and Zitnick (2015).
- **`Geodesic`**$^{\dagger EC}$ (Krähenbühl and Koltun, 2014) starts from an over-segmentation of the image based on Dollár and Zitnick (2015). Classifiers are used to place seeds for a geodesic distance transform. Level sets of each of the distance transforms define the figure/ground segmentations that are the proposals.
- **`MCG`**$^{\dagger EC}$ (Arbeláez *et al.*, 2014) introduces a fast algorithm for computing multi-scale hierarchical segmentations building on Dollár and Zitnick (2015). Segments are merged based on edge strength and the resulting object hypotheses are ranked using cues such as size, location, shape, and edge strength.

6.2.2 Window scoring proposal methods

An alternate approach for generating detection proposals is to score each candidate window according to how likely it is to contain an object. Compared to grouping approaches these methods usually only return bounding boxes and tend to be faster. Unless window sampling is performed very densely, this approach typically generates proposals with low localisation accuracy. Some methods counteract this by refining the location of the generated windows.

- **`Objectness`**† (Alexe *et al.*, 2010, 2012) is one of the earliest and well known proposal methods. An initial set of proposals is selected from salient locations in an image, these proposals are then scored according to multiple cues including colour, edges, location, size, and the strong "superpixel straddling" cue.
- **`Rahtu`**† (Rahtu *et al.*, 2011) begins with a large pool of proposal regions generated from individual superpixels, pairs and triplets of superpixels, and multiple

randomly sampled boxes. The scoring strategy used by `Objectness` is revisited, and improvements are proposed. Blaschko *et al.* (2013) adds additional low-level features and highlights the importance of properly tuned non-maximum suppression.

- **`Bing`**† (Cheng *et al.*, 2014) uses a simple linear classifier trained over edge features and applied in a sliding window manner. Using adequate approximations a very fast class agnostic detector is obtained (1 ms/image on CPU). However, it was shown that the classifier has minimal influence and similar performance can be obtained *without* looking at the image (Zhao *et al.*, 2014). This image independent method is named `CrackingBing.`
- **`EdgeBoxes`**$^{\dagger EC}$ (Zitnick and Dollár, 2014) also starts from a coarse sliding window pattern, but builds on object boundary estimates (obtained via structured decision forests (Dollár and Zitnick, 2013, 2015)) and adds a subsequent refinement step to improve localisation. No parameters are learned. The authors propose tuning the density of the sliding window pattern and the threshold of the non-maximum suppression to tune the method for different overlap thresholds (see §6.5).
- **`Feng`** (Feng *et al.*, 2011) poses proposal generation as the search for salient image content and introduces new saliency measures, including the ease with which a potential object can be composed from the rest of the image. The sliding window paradigm is used and every location scored according to the saliency cues.
- **`Zhang`** (Zhang *et al.*, 2011) proposes to train a cascade of ranking SVMs on simple gradient features. The first stage has separate classifiers for each scale and aspect ratio; the second stage ranks all proposals from the previous stage. All SVMs are trained using structured output learning to score windows higher that overlap more with objects. Because the cascade is trained and tested over the same set of categories, it is unclear how well this approach generalises across categories.
- **`RandomizedSeeds`** (Van Den Bergh *et al.*, 2013) uses multiple randomised SEED superpixel maps (Van den Bergh *et al.*, 2014) to score each candidate window. The scoring is done using a simple metric similar to "superpixel straddling" from `Objectness`, no additional cues are used. The authors show that using multiple superpixel maps significantly improves recall.

6.2.3 Alternative proposal methods

- **`ShapeSharing`** (Kim and Grauman, 2012) is a non-parametric, data-driven method that transfers object shapes from exemplars into test images by matching edges. The resulting regions are subsequently merged and refined by solving graph cuts.
- **`Multibox`** (Szegedy *et al.*, 2015b; Erhan *et al.*, 2014) trains a neural network to directly regress a fixed number of proposals in the image without sliding the network over the image. Each of the proposals has its own location bias to diversify the location of the proposals. The authors report top results on ImageNet.

6.2.4 Baseline proposal methods

We additionally consider a set of baselines that serve as reference points. Like all evaluated methods described earlier, the following baselines are class independent:

- **`Uniform`**†: To generate proposals, we uniformly sample the bounding box centre position, square root area, and log aspect ratio. We estimate the range of these parameters on the PASCAL VOC 2007 training set after discarding 0.5% of the smallest and largest values, so that our estimated distribution covers 99% of the data.

- **`Gaussian`**†: Likewise, we estimate a multivariate Gaussian distribution for the bounding box centre position, square root area, and log aspect ratio. After calculating mean and covariance on the training set we sample proposals from this distribution.

- **`SlidingWindow`**†: We place windows on a regular grid as is common for sliding window object detectors. The requested number of proposals is distributed across windows sizes (width and height), and for each window size, we place the windows uniformly. This procedure is inspired by the implementation of `Bing` (Cheng *et al.*, 2014; Zhao *et al.*, 2014).

- **`Superpixels`**†: As we will show, superpixels have an important influence on the behaviour of proposal methods. Since five of the evaluated methods build on Felzenszwalb and Huttenlocher (2004), we use it as a baseline: each low-level segment is used as a detection proposal. This method serves as a lower-bound on recall for methods using superpixels.

It should be noted that with the exception of `Superpixels`, all the baselines generate proposal windows independent of the image content. `SlidingWindow` is deterministic given the image size (similar to `CrackingBing`), while the `Uniform` and `Gaussian` baselines are stochastic.

6.2.5 Proposals versus cascades

Many proposal methods utilise image features to generate candidate windows. One can interpret this process as a discriminative one; given such features a method quickly determines whether a window should be considered for detection. Indeed, many of the surveyed methods include some form of discriminative learning (`SelectiveSearch` and `EdgeBoxes` are notable exceptions). As such, proposal methods are related to cascades (Viola and Jones, 2004; Bourdev and Brandt, 2005; Harzallah *et al.*, 2009; Dollár *et al.*, 2012a), which use a fast but inaccurate classifier to discard a vast majority of unpromising proposals. Although traditionally used for class specific detection, cascades can also apply to sets of categories (Torralba *et al.*, 2007; Zehnder *et al.*, 2008).

The key distinction between traditional cascades and proposal methods is that the latter is required to generalise beyond object classes observed during training. So what allows discriminatively trained proposal methods to generalise to unseen categories? A key assumption is that training a classifier for a large enough number of categories is sufficient to generalise to unseen categories (for example, after training on cats and dogs proposals may generalise to other animals). Additionally, the discriminative power of the classifier is often limited (e.g. `Bing` and `Zhang`), thus preventing overfitting to the training classes and forcing the classifier to learn coarse properties shared by all object (e.g. "objects are roundish"). This key distinction is also noted in Chavali *et al.* (2015). We test the generalisation of proposal methods by evaluating on datasets with many additional classes in section 6.4.

6.2.6 Controlling the number of proposals

In this work we will perform an extensive apples-to-apples comparison of the 12 methods (plus 4 baselines) listed in table 6.1. In order to be able to compare amongst methods, for each method we need to control the number of proposals produced per image. By default, the evaluated methods provide variable numbers of detection proposals, ranging from just a few ($\sim 10^2$) to a large number ($\sim 10^5$). Additionally, some methods output sorted or scored proposals, while others do not. Having more proposals increases the chance for high recall, thus for each method in all experiments we attempt to carefully control the number of generated proposals. Details are provided next.

Albeit not all having explicit control over the number of proposals, `Objectness`, `CPMC`, `Endres`, `Selective Search`, `Rahtu`, `Bing`, `MCG`, and `EdgeBoxes` do provide scored or sorted proposals so we can use the top k. `Rantalankila`, `Rigor`, and `Geodesic` provide neither direct control over the number of proposals nor sorted proposals, but indirect control over k can be obtained by altering other parameters. Thus, we record the number of produced proposals on a subset of the images for different parameters and linearly interpolate between the parameter settings to control k. For `RandomizedPrim's`, which lacks any control over the number of proposals, we randomly sample k proposals.

Finally, we observed a number of methods produce duplicate proposals. All such duplicates were removed.

6.3 Proposal repeatability

Training a detector on detection proposals rather than on all sliding windows modifies the appearance distribution of both positive and negative windows. In section 6.4, we look into how well the different object proposals overlap with ground truth annotations of objects, which is an analysis of the positive window distribution. In this section we analyse the distribution of negative windows: if the proposal method does not consistently propose windows on similar image content without objects or with partial objects, the classifier may have difficulty generating scores on negative windows on the test set. As an extreme, motivational example, consider a proposal method that generates proposals containing only objects on the training set but containing both

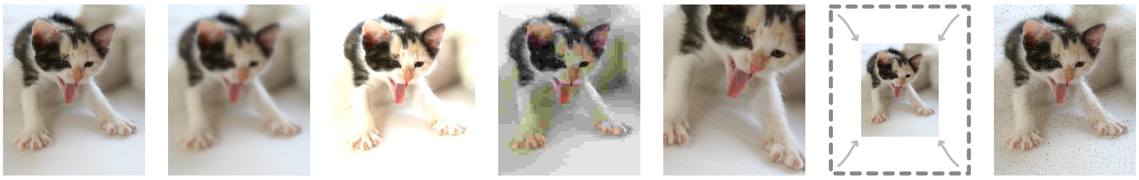


Figure 6.2: Example of the image perturbations considered. Top to bottom, left to right: original, blur, illumination, JPEG artefact, rotation, scale perturbations, and "salt and pepper" noise.

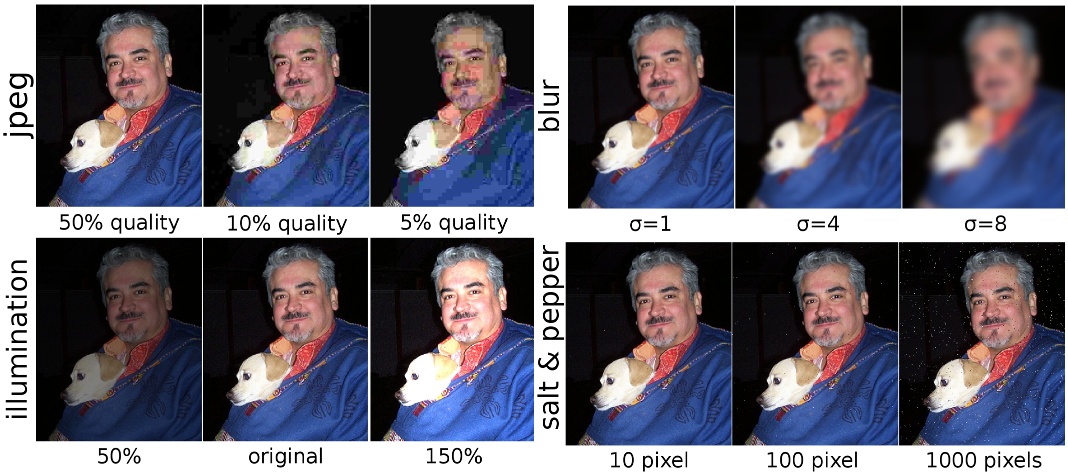


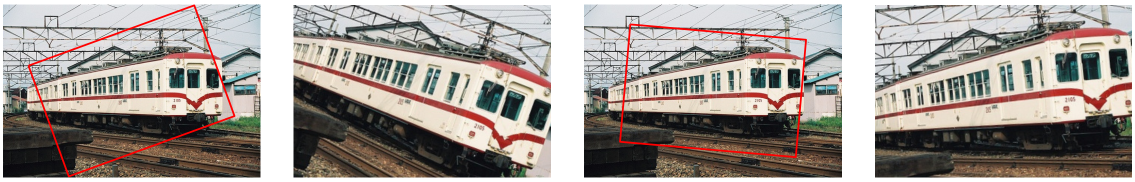
Figure 6.3: Illustration of the perturbation ranges used for the repeatability experiments.

objects and negative windows on the test set. A classifier trained on such proposals would be unable to differentiate objects from background, thus at test time would give useless scores for the negative windows. Thus we expect that a consistent appearance distribution for proposals *on the background* is likewise relevant for a detector.

We call the property of proposals being placed on similar image content the *repeatability* of a proposal method. Intuitively proposals should be repeatable on slightly different images with the same content. To evaluate repeatability we compare proposals that are generated for one image with proposals generated for a slightly modified version of the same image. PASCAL VOC (Everingham *et al.*, 2014) does not contain suitable images. An alternative is the dataset of Mikolajczyk *et al.* (2005), but it only consists of 54 images and even fewer objects. Instead, we opt to apply synthetic transformations to PASCAL images.

6.3.1 Evaluation protocol for repeatability

Our evaluation protocol is inspired by Mikolajczyk *et al.* (2005), which evaluates interest point repeatability. For each image in the PASCAL VOC 2007 test set (Everingham *et al.*, 2014), we generate several perturbed versions. We consider blur, rotation, scale, illumination, JPEG compression, and "salt and pepper" noise (see figures 6.2 and 6.3).



(a) Rotation of 20°. (b) Crop from (a). (c) Rotation of −5°. (d) Crop from (c).

Figure 6.4: Examples of rotation perturbation. (a) shows the largest rectangle with the same aspect as the original image that can fit into the image under a 20° rotation, and (b) the resulting crop. All other rotations are cropped to the same dimensions, e.g. the −5° rotation in (c) to the crop in (d).

For each pair of reference and perturbed images we compute detection proposals with a given method (generating 1000 windows per image). The proposals are projected back from the perturbed into the reference image and then matched to the proposals in the reference image. In the case of rotation, all proposals whose centre lies outside the image after projection are removed before matching. For matching we use the intersection over union (IoU) criterion and we solve the resulting bipartite matching problem greedily for efficiency reasons. Given the matching, we plot the recall for every IoU threshold and *define the repeatability to be the area under this "recall versus IoU threshold" curve between IoU* 0 *and* 1[12]. This is similar to computing the average best overlap (ABO, see §6.A) for the proposals on the reference image. Methods that propose windows at similar locations at high IoU—and thus on similar image content—are more repeatable, since the area under the curve is larger.

One issue regarding such proposal matching is that large windows are more likely to match than smaller ones since the same perturbation will have a larger relative effect on smaller windows. This effect is important to consider since different methods have very different distributions of proposal window sizes as can be seen in figure 6.5a. To reduce the impact of this effect, we bin the original image windows by area into 10 groups, and evaluate the area under the recall versus IoU curve per size group. In figure 6.5b we show the recall versus IoU curve for a small blur perturbation for each of the 10 groups. As expected, large proposals have higher repeatability. In order to measure repeatability independently of the distribution of windows sizes, in all remaining repeatability experiments in figure 6.5 we show the (unweighted) average across the 10 size groups.

We omit the slowest two methods, `CPMC` and `Endres`, due to computational constraints (these experiments require running the detectors ~50 times on the entire PASCAL test set, once for every perturbation).

[12] In contrast to the average recall (AR) used in later sections, we use the area under the entire curve. We are interested in how much proposals change, which is independent of the PASCAL overlap criterion.

6.3.2 Repeatability experiments and results

There are some salient aspects of the result curves in figure 6.5 that need additional explanation. First, not all methods have 100% repeatability when there is no perturbation. This is due to random components in the selection of proposals for several methods. Attempting to remove a method's random component is beyond the scope of this work and could potentially considerably alter the method. A second important aspect is the large drop of repeatability for most methods, even for subtle image changes. We observed that many of the methods based on superpixels are particularly prone to such perturbations. Indeed the `Superpixels` baseline itself shows high sensitivity to perturbations, so the instability of the superpixels likely explains much of this effect. Inversely we notice that methods that are not based on superpixels are most robust to small image changes (e.g. `Bing` and also the baselines that ignore image content).

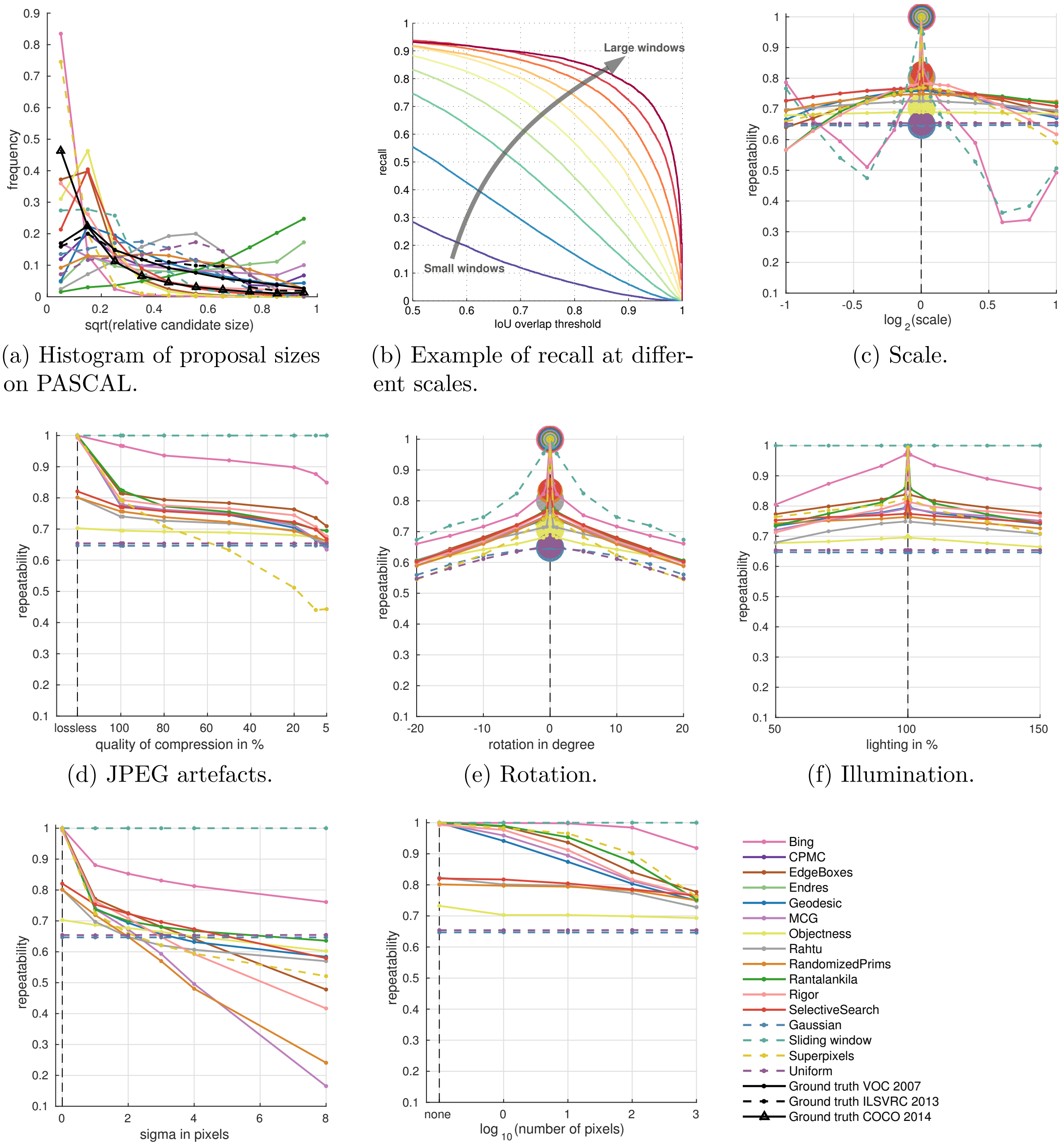
We now discuss the details and effects of each perturbation on repeatability, shown in figure 6.5:

Scale (6.5c). We uniformly sample the scale factor from .5× to 2×, and test additional scales near the original resolution (.9×, .95×, .99×, 1.01×, 1.05×, 1.1×). Upscaling is done with bicubic interpolation and downscaling with anti-aliasing. All methods except `Bing` show a drastic drop with small scale changes, but suffer only minor degradation for larger changes. `Bing` is more robust to small scale changes; however, it is more sensitive to larger changes due to its use of a coarse set of box sizes while searching for candidates (this also accounts for its dip in repeatability at half scales). The `SlidingWindow` baseline suffers from the same effect.

JPEG artefacts (6.5d). To create JPEG artefacts we write the target image to disk with the Matlab function imwrite and specify a quality settings ranging from 5% to 100%, see figure 6.3. Even the 100% quality setting is lossy, so we also include a lossless setting for comparison. Similar to scale change, even slight compression has a large effect and more aggressive compression shows monotonic degradation. Despite using gradient information, `Bing` is most robust to these kind of changes.

Rotation (6.5e). We rotate the image in 5° steps between $-20°$ and 20°. To ensure roughly the same content is visible under all rotations, we construct the largest box with the same aspect as the original image that fits into the image under a 20° rotation and use this crop for all other rotations, see figure 6.4. All proposal methods are equally affected by image rotation. The drop of the `Uniform` and `Gaussian` baselines indicate the repeatability loss due to the fact that we are matching rotated bounding boxes.

Illumination (6.5f). To synthetically alter illumination of an image we changed its brightness channel in HSB colour space. We vary the brightness between 50% and 150% of the original image so that some over and under saturation occurs, see figure 6.3. Repeatability under illumination changes shows a similar trend as under JPEG artefacts.



(a) Histogram of proposal sizes on PASCAL.

(b) Example of recall at different scales.

(c) Scale.

(d) JPEG artefacts.

(e) Rotation.

(f) Illumination.

(g) Blur.

(h) Salt and pepper noise.

Figure 6.5: Repeatability results under various perturbations.

Methods based on superpixels are heavily affected. `Bing` is more robust, likely due to use of gradient information which is known to be fairly robust to illumination changes.

Blur (6.5g). We blur the images with a Gaussian kernel with standard deviations $0 \leq \sigma \leq 8$, see figure 6.3. The repeatability results again exhibit a similar trend although the drop is stronger for a small σ.

Salt and pepper noise (6.5h). We sample between 1 and 1000 random locations in the image and change the colour of the pixel to white if it is a dark and to black otherwise, see figure 6.3. Surprisingly, most methods already lose some repeatability when even a single pixel is changed. Significant degradation in repeatability for the majority of the methods occurs when merely ten pixels are modified.

Discussion. Small changes to an image cause noticeable differences in the set of detection proposals for all methods except `Bing`. The higher repeatability of `Bing` is explained by its sliding window pattern, which has been designed to cover almost all possible annotations with IoU $= 0.5$ (see also `Cracking Bing`(Zhao *et al.*, 2014)). As one cause for poor repeatability we identify the segmentation algorithm on which many methods build. Among all proposal methods, `EdgeBoxes` also performs favourably, possibly because it avoids the hard decision of grouping pixels into superpixels.

We also experimented with repeatability of boxes that touch annotations sufficiently (IoU ≥ 0.5), which showed very similar trends, indicating that the issue of repeatability also applies to proposals that partially cover objects.

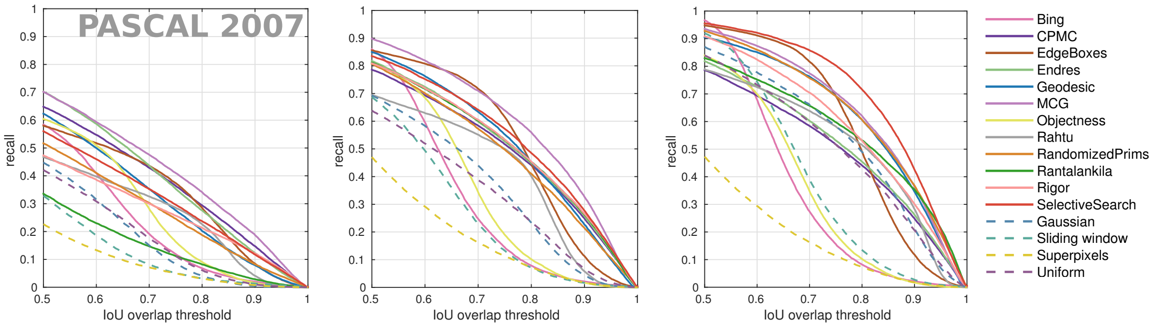
Different applications will be more or less sensitive to repeatability. Our results indicate that if repeatability is a concern, the proposal method should be selected with care. For object detection, another aspect of interest is recall, which we explore in the next section.

6.4 Proposal recall

When using detection proposals for detection it is important to have a good coverage of the objects of interest in the test image, since missed objects cannot be recovered in the subsequent classification stage. Thus it is common practice to evaluate the quality of proposals based on the recall of the ground truth annotations.

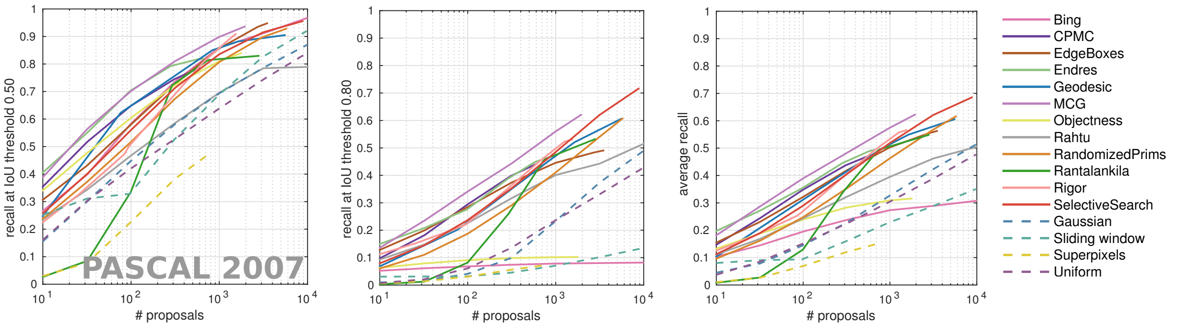
6.4.1 Evaluation protocol for recall

The protocol introduced in Alexe *et al.* (2010) (using the PASCAL VOC 2007 dataset (Everingham *et al.*, 2014)) has served as a guideline for most evaluations in the literature. While previous papers do show various comparisons on PASCAL, the train and test sets vary amongst papers, and the metrics shown tend to favour different methods. We provide an extensive and unified evaluation and show that different metrics result in different rankings of proposal methods (e.g. see figure 6.6b versus 6.7b).



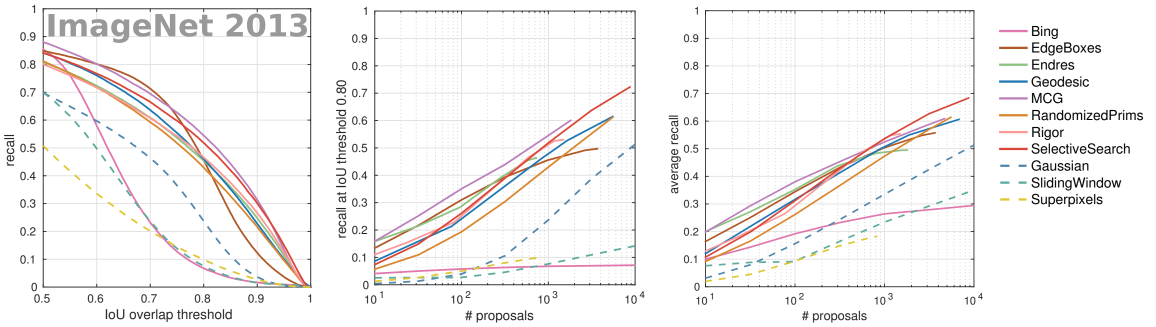
(a) 100 (b) 1 000 (c) 10 000

Figure 6.6: Recall versus IoU threshold on the PASCAL VOC 2007 test set for different number of proposals per image.



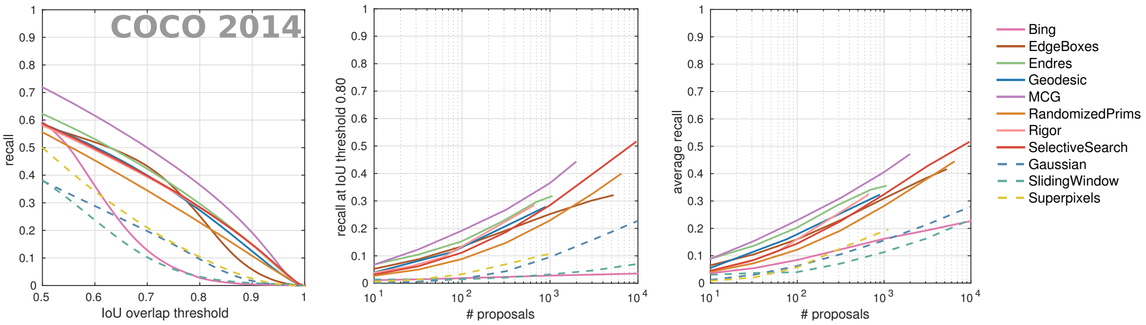
(a) Recall at 0.5 IoU. (b) Recall at 0.8 IoU. (c) Average recall.

Figure 6.7: Recall versus number of proposal windows on the PASCAL VOC 2007 test set.



(a) 1 000 proposals per image. (b) Recall at 0.8 IoU. (c) Average recall.

Figure 6.8: Recall on the ImageNet 2013 validation set.



(a) 1 000 proposals per image. (b) Recall at 0.8 IoU. (c) Average recall (between [0.5, 1] IoU).

Figure 6.9: Recall on the MS COCO 2014 validation set.

Metrics. Evaluating (class agnostic) detection proposals is quite different from traditional class-specific detection (Hoiem *et al.*, 2012a) since most metrics (class confusion, background confusion, precision, etc.) do not apply. Instead, one of the primary metrics for evaluating proposals is, for a fixed number of proposals, the fraction of ground truth annotations covered as the intersection over union (IoU) threshold is varied (figure 6.6). Another common and complementary metric is, for a fixed IoU threshold, proposal recall as the number of proposals is varied (figure 6.7a, 6.7b). Finally, we define and report a novel metric, the average recall (AR) between IoU 0.5 to 1, and plot AR versus number of proposals (figure 6.7c).

PASCAL. We evaluate recall on the full PASCAL VOC 2007 test set (Everingham *et al.*, 2014), which includes 20 object categories present in ~5 000 unconstrained images. For the purpose of proposal evaluation we include all 20 object categories and all ground truth bounding boxes, including "difficult" ones, since our goal is to measure maximum recall. In contrast to (Alexe *et al.*, 2010), we compute a matching between proposals and ground truth, so one proposal cannot cover two objects. Note that while different methods may be trained on different sets of object categories and subsets of data, we believe evaluating on all categories at test time is appropriate as we care about absolute proposal quality. Such an evaluation strategy is further supported as many methods have no training stage, yet provide competitive results (e.g. `SelectiveSearch`).

ImageNet. The PASCAL VOC 2007 test set, on which most proposal methods have been previously evaluated, has only 20 categories, yet detection proposal methods claim to predict proposals for *any* object category. Thus there is some concern that the proposal methods may be tuned to the PASCAL categories and not generalise well to novel categories. To investigate this potential bias, we also evaluate methods on the larger ImageNet (Deng *et al.*, 2009) 2013 validation set, which contains annotations for 200 categories in over ~20 000 images. It should be noted that these 200 categories are *not* fine grained versions of the PASCAL ones. They include additional types of animals (e.g. crustaceans), food items (e.g. hot-dogs), household items (e.g. diapers), and other diverse object categories.

MS COCO. Although ImageNet has 180 more classes than PASCAL, it is still similar in statistics like number of objects per image and size of objects. Microsoft Common Objects in Context (MS COCO) (Lin *et al.*, 2014) has more objects per image, smaller objects, but also fewer object classes (80 object categories). We evaluate the recall of this dataset to further investigate potential biases of proposal methods. We evaluate the recall on all annotations excluding the "crowd" annotations which may mark large image areas including a lot of background.

6.4.2 Recall results

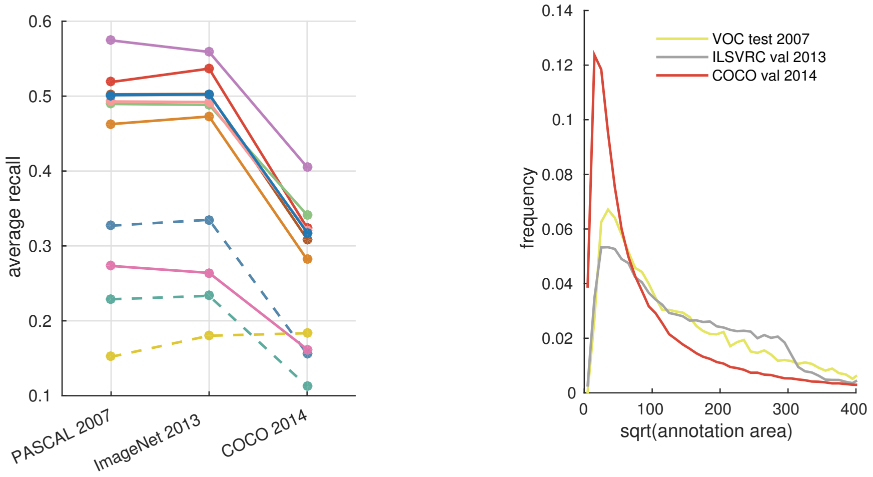
PASCAL Results in figure 6.6 and 6.7 present a consistent trend across the different metrics. `MCG`, `EdgeBoxes`, `SelectiveSearch`, `Rigor`, and `Geodesic` are the best methods across different numbers of proposals. `SelectiveSearch` is surprisingly effective despite being fully hand-crafted (no machine learning involved). When considering less than 10^3 proposals, `MCG`, `Endres`, and `CPMC` provide strong results.

Overall, the methods fall into two groups: well localised methods that gradually lose recall as the IoU threshold increases and methods that only provide coarse bounding box locations, so their recall drops rapidly. All baseline methods, as well as `Bing`, `Rahtu`, `Objectness`, and `EdgeBoxes` fall into the latter category. `Bing` in particular, while providing high repeatability, only provides high recall at IoU $= 0.5$ and drops dramatically when requiring higher overlap (the reason for this is identified in Zhao *et al.* (2014)).

Baselines. When inspecting figure 6.6 from left to right, one notices that with few proposals the baselines provide relatively low recall (figure 6.6a). However as the number of proposals increases, `Gaussian` and `Uniform` become more competitive (figure 6.6b). In relative gain, detection proposal methods have most to offer for low numbers of windows.

Average Recall. Rather than reporting recall at particular IoU thresholds, we also report the average recall (AR) between IoU 0.5 to 1 (which is related to the ABO metric, see section 6.A), and plot AR for varying number of proposals in figure 6.7c. Much like the average precision (AP) metric for (class specific) object detection, AR summarises proposal performance across IoU thresholds (for a given number of proposals). In fact, in section 6.5 we will show that AR correlates well with detection performance. As can be seen in figure 6.7c, `MCG` performs well across the entire range of number of proposals. `Endres` and `EdgeBoxes` work well for a low number of proposals while for a higher number of proposals `Rigor` and `SelectiveSearch` perform best.

ImageNet. As discussed, compared to PASCAL, ImageNet includes 10× ground truth classes and 4× images. Somewhat surprisingly the ImageNet results in figure 6.8



(a) AR with 1000 proposals

(b) Ground truth size

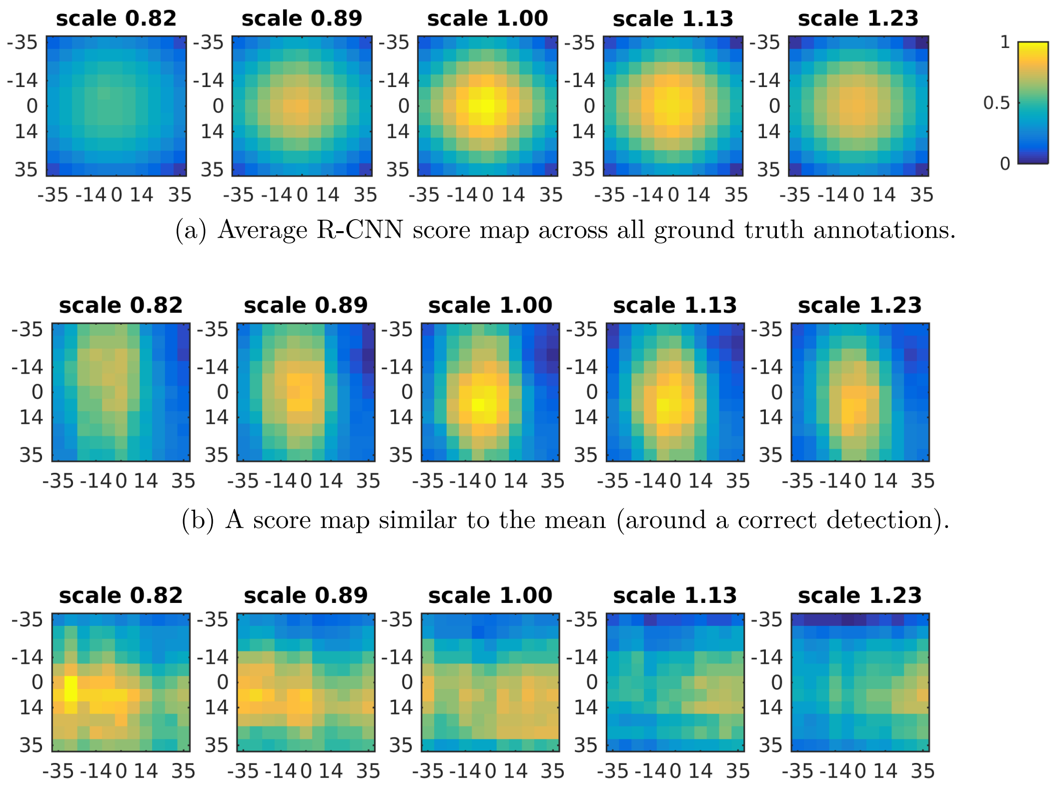
Figure 6.10: Comparison between all considered datasets: PASCAL VOC 2007 test set, ImageNet 2013 validation set, MS COCO 2014 validation set (see methods legend fig. 6.7c).

are almost identical to the ones in figures 6.6b, 6.7b, and 6.7c. To understand this phenomenon, we note that the statistics of ImageNet match the ones of PASCAL. In particular the typical image size and the mean number of object annotation per image (three) is similar in both datasets. This helps explain why the recall behaviour is similar, and why methods tuned on PASCAL still perform well on ImageNet.

MS COCO. We present the same results for MS COCO in figure 6.9. We see different absolute numbers, yet similar trends with some notable exceptions as can be seen in figure 6.10a. `EdgeBoxes` no longer ranks significantly better than `SelectiveSearch`, `Geodesic` and `Rigor` for few proposals. `MCG` and `Endres` improve relative to the other methods, in particular for a higher number of proposals. We attribute this difference to different statistics of the dataset, particularly the different size distribution, see figure 6.10b.

Overall, `MCG` is the top performing method across all datasets in terms of both recall and AR at all settings. This is readily apparent in figure 6.10a.

Generalisation. We emphasise that although the results on PASCAL, ImageNet, and MS COCO are quite similar (see figure 6.10a), ImageNet covers 200 object categories, many of them unrelated to the 20 PASCAL categories and COCO has significantly different statistics. In other words, there is no measurable over-fitting of the detection proposal methods towards the PASCAL categories. This suggests that proposal methods transfer adequately amongst object classes, and can thus be considered true “objectness” measures.



(a) Average R-CNN score map across all ground truth annotations.

(b) A score map similar to the mean (around a correct detection).

(c) A score map different than the mean (around a missed detection).

Figure 6.11: Normalised score maps of the R-CNN around ground truth annotations on the PASCAL 2007 test set. One grid cell in each map has width and height of ∼7px after the object height has been resized to the detector window of 227×227 px (3% of the object height).

6.5 Using the detection proposals

In this section we analyse detection proposals for use with object detectors. We consider two well known and quite distinct approaches to object detection. First we use a variant of the popular DPM part-based sliding window detector (Felzenszwalb *et al.*, 2010), specifically the LM-LLDA detector (Girshick and Malik, 2013). We also test the state of the art R-CNN (Girshick *et al.*, 2014) and Fast R-CNN (Girshick, 2015) detectors which couple object proposals with a convolutional neural network classification stage. Our goals are twofold. First, we aim to measure the performance of different proposal methods for object detection. Second, we are interested in evaluating how well the proposal metrics reported in the previous sections can serve as a proxy for predicting final detection performance. All following experiments involving proposals use 1 000 proposals.

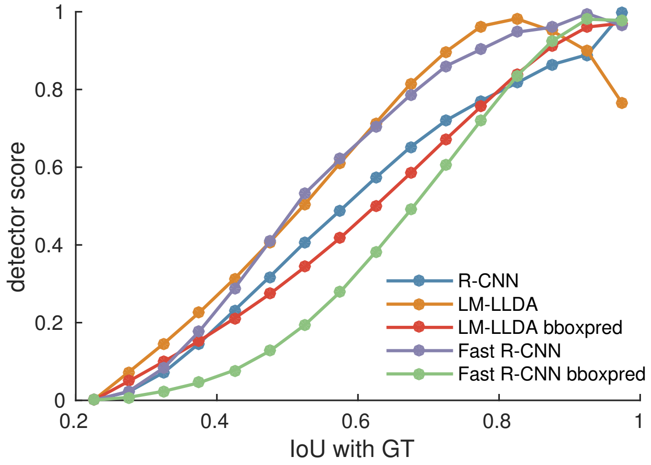


Figure 6.12: Normalised detector scores as a function of the overlap between the detector window and the ground truth.

6.5.1 Detector responses around objects

As a preliminary experiment, we aim to quantify the importance of having well localised proposals for object detection. We begin by measuring how detection scores are affected by the overlap between the detector window and the ground truth annotation on the PASCAL 2007 test set (Everingham *et al.*, 2014). When considering the detectors' bounding box prediction, we use the refined position to compute the overlap.

Score map. Figure 6.11a shows the average R-CNN detection score around the ground truth annotations. We notice that the score map is symmetric and attains a maximum at the ground truth object location. In other words, the detector has no systematic spatial or scale bias. However, averaging the score maps removes small details and imperfections of individual score maps. When considering individual activations instead of the average, we observe a high variance in the quality of the score maps, see figures 6.11b and 6.11c.

Score vs IoU. In figure 6.12 we show average detection scores for proposals with varying IoU overlap with the ground truth. The scores have been scaled between zero and one per class before averaging across classes. The drop of the LM-LLDA scores at high overlaps is due to a bias introduced during training by the latent location estimation on positive samples; this bias is compensated for by the subsequent bounding box prediction stage of LM-LLDA. For Fast R-CNN, the bounding box prediction effectively improves proposal IoU with the ground truth and results in a substantial shift of the curve to the right.

Localisation. We observe from figure 6.12 that both LM-LLDA and R-CNN exhibit an almost linear increase in detection score as IoU increases (especially between 0.4 and 0.8 IoU). From this we conclude that there is no IoU threshold that is “sufficiently good”

Proposals	LM-LLDA	R-CNN	Fast R-CNN	ΔTrain
`Dense`	33.5/34.4	–	–	–
`Bing`	21.8/22.4	36.7	37.3/49.0	+6.3
`CPMC`	30.0/30.7	51.7	53.7/57.1	-1.3
`EdgeBoxes`	31.8/32.2	53.0	55.4/**60.4**	+3.3
`Endres`	31.2/31.7	52.8	54.2/57.4	-0.2
`Geodesic`	31.8/32.2	53.8	53.6/57.5	-0.4
`MCG`	**32.5/33.0**	**56.5**	**58.1**/60.3	+1.8
`Objectness`	25.0/25.4	39.7	41.5/51.4	+9.1
`Rahtu`	29.6/30.4	46.1	48.9/53.6	+0.7
`RandomizedPrims`	30.5/30.9	51.6	53.2/57.6	-0.6
`Rantalankila`	30.9/31.4	53.1	55.0/57.9	-0.5
`Rigor`	31.5/32.1	54.1	55.4/58.4	-0.2
`SelectiveSearch`	31.7/32.3	54.6	56.3/59.5	+0.0
`Gaussian`	*27.3/28.0*	*40.6*	*44.6/50.8*	+0.8
`Sliding window`	20.7/21.5	32.7	32.7/44.8	+3.3
`Superpixels`	11.2/11.3	17.6	15.4/20.3	-2.0
`Uniform`	26.0/26.6	37.3	39.5/46.9	-0.1

Table 6.2: Mean average precision (mAP) on PASCAL 2007 for multiple detectors and proposal methods (using 1 000 proposals). LM-LLDA and Fast R-CNN results shown before/after bounding box regression. The final column shows the change in mAP obtained from re-training Fast R-CNN (with box regression) for the specific proposal method.

for obtaining top detection quality. We thus consider that improving localisation of proposals is as important as increasing ground truth recall, and the linear relation helps motivate us to linearly reward localisation in the average recall metric (see section 6.4.2). For Fast R-CNN there is also an almost linear relation, but performance saturates earlier. Thus, Fast R-CNN is likely to also benefit from better localisation, but up to a point.

6.5.2 LM-LLDA detection performance

We use pre-trained LM-LLDA (Girshick and Malik, 2013) models to generate dense detections using the standard sliding window setup and subsequently apply different proposals to filter these detections at test time. This does not speed-up detection, but enables evaluating the effect of proposals on detection quality. A priori we may expect that detection results will deteriorate due to lost recall, but conversely, they may improve if the proposals filter out windows that would otherwise be false positives.

Implementation. We take the raw detections of LM-LLDA before non-maximum suppression (NMS) and filter them with the detection proposals of each method. We keep all detections that overlap more than 0.8 IoU with a candidate proposal and subsequently apply NMS to the surviving detections. As a final step we do bounding box regression, as is common for DPM models (Felzenszwalb *et al.*, 2010). This procedure returns predictions near to, but distinct from, each proposal.

Note that this experimental setup "goes against the original spirit" of detection proposals in the sense that it evaluates several locations around each proposal. Successful application of proposals requires the detector to be robust to poor localisation of proposals. The LM-LLDA detector exhibits spatially highly varying scores, which causes no issues with higly redundant proposals in the setting of the sliding window approach, but is problematic on sparse proposals. For the application of proposals in their original itention see section 6.5.3.

Results. Table 6.2, LM-LLDA columns, show that using 1 000 proposals decreases detection quality compared with the original sliding window setup[13] by about 1-2 mAP for the best performing methods, see top row (`Dense`) versus the rows below. The five top performing methods all have mAP between 32.0 and 33.0 and are marked in green: `MCG`, `SelectiveSearch`, `EdgeBoxes`, `Geodesic`, and `Rigor`. Note that the difference between these methods and the `Gaussian` baseline is fairly small (33.0 versus 28.0 mAP).

When we compare these results with figure 6.7c at 1 000 proposals, we see that methods are ranked similarly. Methods with high average recall (AR) also have high mAP, and methods with lower AR also have lower mAP. We analyse the correlation between AR and mAP more closely in section 6.5.4.

From table 6.3 we see that the per-class performance can be grouped into three cases: classes on which the best proposals (1) clearly hurt performance (bicycle, boat, bottle, car, chair, horse, mbike, person), (2) improve performance (cat, table, dog), (3) do not show significant change (all remaining classes). In the case of (1) we observe both reduced recall and reduced precision in the detection curves, probably because bad localisation decreases the scores of strong detections.

6.5.3 R-CNN detection performance

The highly successful and widely used R-CNN detector (Girshick *et al.*, 2014) couples detection proposals with a convolutional neural network classification stage. It was designed from the ground up to rely on proposals, making it a perfect candidate for our case study. We report results for both the original R-CNN detector and also the improved Fast R-CNN (Girshick, 2015). We focus primarily on Fast R-CNN due to its efficiency and higher detection accuracy.

Implementation. For each proposal method we re-train and test Fast R-CNN (using the medium model M for efficiency). Unlike Fast R-CNN, the original R-CNN is fairly slow to train; we therefore experiment with the R-CNN model that is published with the code and which has been trained on 2 000 `SelectiveSearch` proposals.

Results. Although the absolute mAP numbers are considerably higher for Fast R-CNN (nearly double mAP), the results (Fast R-CNN and R-CNN) in table 6.2 show a

[13] Not to be confused with the `SlidingWindow` proposals baseline.

	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
LM-LLDA Dense	33.7	61.3	12.4	18.5	26.7	53.0	57.2	22.4	22.7	25.6	25.1	14.0	59.2	51.0	39.1	13.6	21.7	38.0	48.8	44.0	34.4
Bing	-7.5	-23.2	-6.2	-8.1	-10.6	-13.3	-17.5	-6.8	-9.8	-15.4	-7.5	-1.4	-19.6	-19.0	-16.1	-3.4	-6.6	-18.1	-18.8	-10.0	-11.9
CPMC	-1.0	-15.0	-0.2	-4.4	-13.5	-1.8	-9.2	3.2	-9.1	-2.6	5.1	2.2	-4.2	-4.8	-7.0	-2.0	-2.6	1.2	-4.1	-4.9	-3.7
EdgeBoxes	-2.0	-6.1	-0.7	-3.8	**-6.7**	**0.6**	**-5.8**	-1.1	**-2.0**	-1.8	-4.6	0.4	-1.3	**-1.3**	-3.0	-1.7	**-0.1**	-0.9	-0.2	-1.1	**-2.2**
Endres	-1.5	**-5.8**	-0.6	-4.8	-12.7	-1.1	-7.1	3.4	-6.9	-3.2	4.7	1.9	-2.4	-2.4	-7.7	-2.8	-1.9	1.5	0.4	-4.2	-2.7
Geodesic	-1.9	-8.1	-0.2	-4.6	-14.4	0.6	-6.5	2.6	-7.3	**-1.3**	4.7	2.4	-2.5	-2.7	-4.7	-1.2	-0.7	-0.1	**1.9**	0.2	**-2.2**
MCG	-0.7	-7.2	0.1	-3.6	**-6.7**	-1.2	-7.0	3.4	-3.2	-2.3	5.0	1.9	-3.5	**-1.3**	**-1.5**	**-1.1**	-1.3	2.2	0.3	**0.5**	**-1.4**
Objectness	-10.3	-15.1	-2.0	-6.2	-11.0	-9.5	-13.0	-3.6	-10.0	-6.4	-7.8	-1.0	-11.6	-15.9	-13.0	-2.7	-5.8	-11.2	-10.9	-12.9	-9.0
Rahtu	-0.3	-13.2	-0.3	**-1.2**	-13.0	-0.6	-12.0	3.3	-10.5	-4.3	2.0	2.1	-3.2	-4.9	-7.9	-2.8	-4.9	-5.0	0.0	-3.7	-4.0
Rand.Prim	**2.1**	-10.4	-0.5	-4.5	-13.2	-1.9	-10.1	5.0	-6.7	-3.5	2.0	2.4	-4.4	-5.1	-10.0	-2.3	-1.8	1.2	-3.8	-4.4	-3.5
Rantalankila	0.5	-13.6	0.3	-3.0	-12.9	-3.6	-9.0	4.4	-5.6	-3.7	4.1	2.5	-2.2	-4.0	-7.8	-2.5	-3.8	**2.1**	-1.5	-0.7	-3.0
Rigor	1.7	-7.9	0.5	-4.1	-12.4	-0.8	-9.0	**6.3**	-6.9	-1.7	1.8	**2.9**	**-0.9**	-3.3	-7.7	-1.8	-1.3	1.6	-1.2	-1.7	**-2.3**
SelectiveSearch	1.3	-7.7	**1.0**	-4.3	-11.1	-1.7	-7.8	3.9	-4.8	-1.5	**5.4**	2.2	-1.4	-3.8	-6.0	-1.5	-0.8	0.6	-2.4	-2.1	**-2.1**
Gaussian	-6.6	-13.4	-0.7	-4.4	-15.0	-6.1	-16.0	0.9	-9.1	-8.0	0.3	1.2	-4.2	-6.9	-10.3	-2.3	-6.5	-4.5	-3.6	-12.1	*-6.4*
SlidingWindow	-21.8	-20.7	-3.2	-8.1	-16.6	-14.7	-22.1	-0.7	-9.8	-11.7	-10.2	-1.4	-14.7	-20.1	-14.8	-3.8	-7.7	-21.0	-20.8	-14.8	-12.9
Superpixels	-23.9	-52.2	-3.1	-9.4	-17.4	-43.9	-42.3	-10.2	-11.3	-12.6	-15.8	-8.5	-50.1	-41.7	-30.9	-4.4	-10.6	-25.2	-39.7	-8.2	-23.1
Uniform	-3.2	-18.8	-4.0	-4.8	-15.2	-8.6	-16.6	0.2	-10.4	-8.8	3.7	1.3	-6.6	-11.3	-10.2	-3.6	-8.9	-5.8	-5.1	-20.2	-7.8
Top methods avg.	-0.3	-7.4	0.1	-4.1	-10.2	-0.5	-7.2	3.0	-4.8	-1.7	2.5	2.0	-1.9	-2.5	-4.6	-1.5	-0.8	0.7	-0.3	-0.8	-2.0

Table 6.3: LM-LLDA detection results on PASCAL 2007 (with bounding box regression). The top row indicates the average precision (AP) of LM-LLDA alone, while the other rows show the difference in AP when adding proposal methods. Green indicates improvement of at least 2 AP, blue indicates minor change ($-2 \leq \text{AP} < 2$), and white indicates a decrease by more than 2 AP. `EdgeBoxes` achieves top results on 6 of the 20 categories; `MCG` performs best overall with -1.4 mAP loss.

	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
Bing	56.6	54.9	45.0	28.6	24.6	53.9	63.5	72.5	15.6	59.4	49.0	59.7	68.5	60.3	50.7	16.5	49.0	42.8	64.8	44.9	49.0
CPMC	65.2	61.8	58.2	37.2	17.9	71.0	67.3	76.7	22.9	61.2	64.6	70.1	77.0	69.2	54.8	18.5	52.6	63.4	71.7	61.5	57.1
EdgeBoxes	67.0	69.9	59.8	**46.1**	28.3	72.9	**72.3**	73.8	28.8	**68.1**	62.4	67.6	**79.2**	**73.6**	**62.4**	**28.2**	55.8	61.2	70.4	59.7	**60.4**
Endres	61.5	**70.8**	57.1	33.5	18.0	72.5	68.8	77.3	21.7	61.8	64.5	68.2	78.0	69.9	56.2	21.4	54.5	63.2	72.4	56.9	57.4
Geodesic	63.2	68.0	55.9	39.2	19.8	71.1	70.4	74.4	24.8	65.0	63.5	65.6	78.7	69.2	58.0	20.4	54.5	57.8	70.2	60.9	57.5
MCG	66.6	69.1	60.1	42.0	**28.5**	71.9	**72.3**	77.3	**30.2**	61.3	62.4	69.8	77.4	68.2	62.2	27.5	**57.6**	**66.0**	75.8	59.4	60.3
Objectness	62.4	61.5	51.0	32.0	19.3	65.8	64.3	69.5	18.0	55.4	51.4	60.1	74.1	64.7	50.9	17.3	41.9	50.9	67.8	49.0	51.4
Rahtu	62.8	60.9	53.3	35.1	15.3	72.6	60.5	75.1	15.4	56.9	61.6	66.3	76.3	65.2	51.2	14.1	44.6	58.1	72.0	54.3	53.6
RandomizedPrims	70.2	68.2	55.5	39.5	18.5	72.3	63.7	76.8	25.7	62.4	64.2	68.7	76.6	68.5	51.0	22.4	53.1	62.9	72.4	59.7	57.6
Rantalankila	64.7	66.1	57.2	37.8	19.7	**74.2**	67.5	**78.2**	23.0	63.6	63.4	**70.3**	78.6	69.8	55.9	21.4	50.8	64.3	74.1	58.3	57.9
Rigor	62.6	70.5	57.5	40.1	15.9	72.9	65.7	77.9	28.6	65.1	63.7	68.6	77.9	68.9	54.8	23.3	56.3	63.8	73.7	60.3	58.4
SelectiveSearch	**70.3**	66.9	**61.5**	42.2	21.7	68.3	68.7	76.3	27.5	65.9	**67.0**	69.8	75.5	68.9	57.9	24.6	53.6	63.7	**76.0**	**62.4**	59.5
Gaussian	53.9	66.1	46.6	24.6	10.0	66.6	52.2	77.1	20.6	48.7	64.1	65.5	75.6	64.2	47.0	14.2	38.1	58.2	70.5	53.0	50.8
SlidingWindow	42.0	57.7	40.1	23.7	9.3	60.8	47.8	72.8	12.5	42.1	44.7	63.7	72.8	62.5	44.5	8.5	34.3	47.7	62.3	46.6	44.8
Superpixels	29.7	5.5	19.8	10.4	9.0	7.4	24.4	42.0	15.1	39.9	6.6	30.3	10.7	13.7	12.8	8.9	40.7	18.1	4.9	55.6	20.3
Uniform	51.0	58.0	38.6	24.6	11.7	64.3	50.9	72.3	14.8	43.4	62.6	63.4	73.9	59.3	43.4	10.8	27.5	60.4	69.0	38.3	46.9
best per class	70.3	70.8	61.5	46.1	28.5	74.2	72.3	78.2	30.2	68.1	67.0	70.3	79.2	73.6	62.4	28.2	57.6	66.0	76.0	62.4	62.1

Table 6.4: Fast R-CNN (model M) detection results (AP) on PASCAL VOC 2007. Bold numbers indicate the best proposal method per class, green numbers are within 2 AP of the best result. The "best per class" row shows the best performance when choosing the optimal proposals per class, improving from 60.4 mAP (EdgeBoxes) to 62.1 mAP.

similar trend than the LM-LLDA results. As expected, `SelectiveSearch`, with which Fast R-CNN was developed, performs well, but multiple other proposal methods get similar results. The five top performing methods are similar to the top methods for LM-LLDA: `Rantalankila` edges out `EdgeBoxes` for R-CNN and `Geodesic` for Fast R-CNN. `EdgeBoxes` and `MCG` provide the best results. The gap between `Gaussian` and the top result is more pronounced (60.4 versus 50.8 mAP), but this baseline still performs surprisingly well considering it disregards the image content. We show per-class Fast R-CNN results in table 6.4.

Retraining. To provide a fair comparison amongst proposal methods, the "Fast R-CNN" column in table 6.2 reports results after re-training for each method. The rightmost column of table 6.2 shows the change in mAP when comparing Fast R-CNN (with bounding box regression) trained with 1 000 `SelectiveSearch` proposals and applied at test time with a given proposal method, versus Fast R-CNN trained for the test time proposal method.

Most methods improve from re-training, although the performance of a few degrades. While in most cases the change in mAP is within 1-2 points, re-training provided substantial benefits for `Bing`, `EdgeBoxes`, `Objectness`, and `SlidingWindow`. These methods all have poor localisation at high IoU (see figure 6.6); re-training likely allows Fast R-CNN to compensate for their inferior localisation.

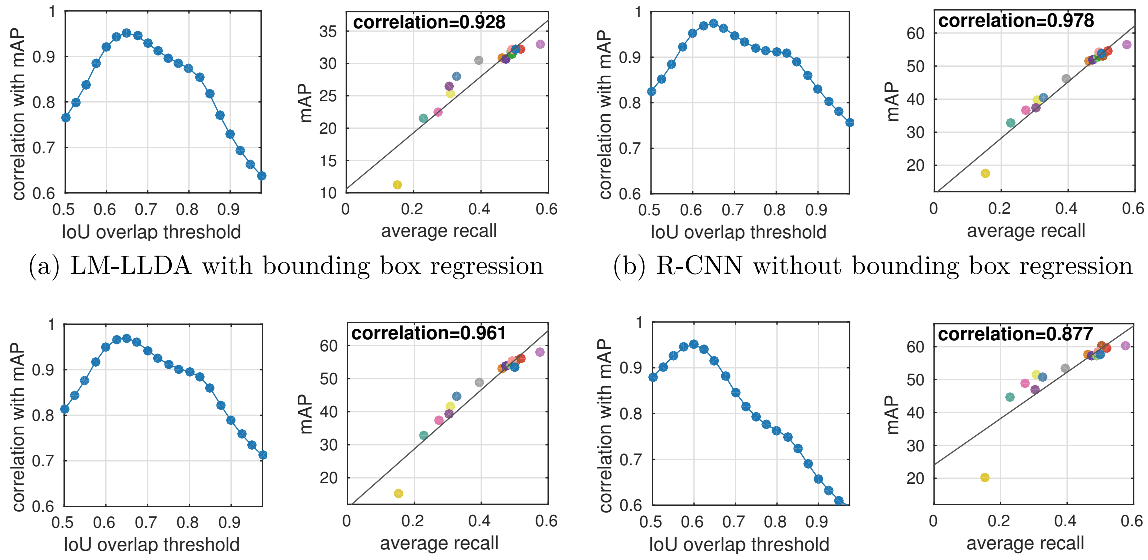
Summary. We emphasise that the various proposal methods exhibit similar ordering with all tested detectors (LM-LLDA, R-CNN, and Fast R-CNN). Our experiments did not reveal any proposal methods as being particularly well-adapted for certain detectors; rather, for object detection some proposals methods are strictly better than others.

6.5.4 Predicting detection performance

We aim to determine which recall metrics from section 6.4 (figures 6.6 and 6.7) serve as the best predictor for detector performance. In figure 6.13 we show the Pearson correlation coefficient between detector performance and two recall metrics: recall at different overlap thresholds (left columns) and the *average recall* (AR) between IoU of 0.5 to 1.0 (right columns)[14]. As before, we use 1 000 proposals per method.

We begin by examining correlation between detection performance and recall at various IoU thresholds (figure 6.13, left columns). All detectors show a strong correlation (> 0.9) at an IoU range of roughly 0.6 to 0.8, with the exception of Fast R-CNN with bounding box prediction, which correlates better for lower overlap. Note that recall at IoU of 0.5 is actually only weakly correlated with detection performance, and methods that optimise for IoU of 0.5, such as `Bing`, are not well suited for use with object detectors (see table 6.2). Thus, although recall at IoU of 0.5 has been traditionally

[14] We compute the average between 0.5 and 1 IoU (and not between 0 and 1 as in section 6.3), because we are interested in recall above the PASCAL evaluation criterion of 0.5 IoU. Proposals with worse overlap than 0.5 are not only harder to classify correctly, but require a potentially large subsequent location refinement to become a successful detection.



(a) LM-LLDA with bounding box regression (b) R-CNN without bounding box regression

(c) Fast R-CNN without bounding box regression (d) Fast R-CNN with bounding box regression

Figure 6.13: Correlation between detector performance on PASCAL 07 and different proposal metrics. Left columns: correlation between mAP and recall at different IoU thresholds. Right columns: correlation between mAP and AR.

used to evaluate object proposals, our analysis shows that it is *not* a good metric for predicting detection performance.

The correlation between detection performance and AR is quite strong, see figure 6.13, right columns. Computing the AR over a partial IoU range (e.g. 0.6 to 0.8) can further increase the correlation; however, since the effect is generally minor, we opted to use AR over the entire range from 0.5 to 1.0 for simplicity. While the strong correlation does not imply that the AR can perfectly predict detection performance, as figure 6.13 shows, the relationship is surprisingly linear. AR over the full range of 0 to 1 IoU (which is similar to ABO, see appendix 6.A) has weaker correlation with mAP, since proposals with low overlap are not sufficient for a successful detection under the PASCAL criterion and are also harder to classify.

For detectors with bounding box regression, the AR computation can be restricted to a tighter IoU range. In figure 6.12, we can observe that detection score of Fast R-CNN saturates earlier. Thus there is little benefit in proposals that are perfectly localised as the bounding box refinement improves the localisation of those proposals. If we restrict the AR to IoU from 0.5 to 0.7, we obtain a higher correlation of 0.949 for Fast R-CNN with bounding box regression (compared to 0.877 in figure 6.13d).

For a more detailed analysis of the correlation between mAP and AR we show the correlation for each class for different detectors in figure 6.14. The per-class correlation is highest for R-CNN and Fast R-CNN without regression.

We conclude that AR allows us to identify good proposal methods for object detection. The AR metric is simple, easy to justify, and is strongly correlated with detection performance. Note that our analysis only covers the case in which all methods produce

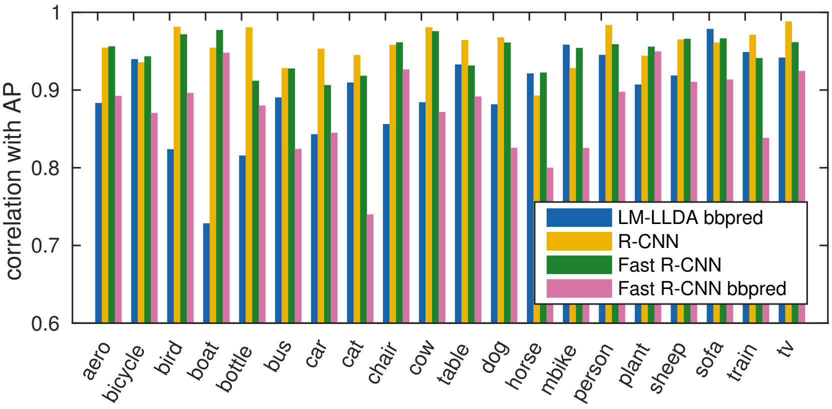
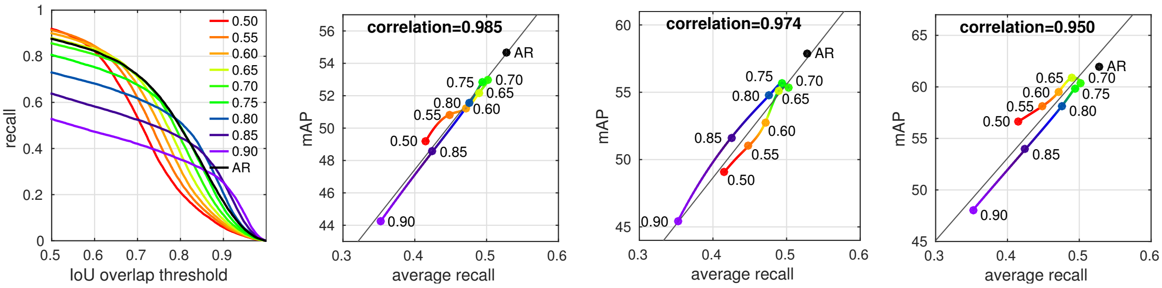


Figure 6.14: Correlation between AR and AP for each PASCAL VOC class and detector across all proposal methods.



(a) Recall versus IoU (b) R-CNN (c) Fast R-CNN without regression (d) Fast R-CNN with regression

Figure 6.15: Finetuning `EdgeBoxes` to optimise AR results in top detector performance. These results further support the conclusion that AR is a good predictor for mAP and suggest that it can be used for fine-tuning proposal methods.

the same number of proposals. As Girshick (2015) points out, as the number of proposals increases, AR will necessarily increase but resulting detector performance saturates and may even degrade. For a fixed number of proposals, however, AR is a good predictor of detection performance. We suggest that future proposal methods should aim to optimise this metric.

6.5.5 Tuning proposal methods

All previous experiments evaluate proposal methods using original parameter settings. However many methods have free parameters that allow for fine-tuning. For example, when adjusting window sampling density and the non-maximum suppression (NMS) in `EdgeBoxes` (Zitnick and Dollár, 2014), it is possible to trade-off low recall with good localisation for higher recall with worse localisation (a similar observation was made in Blaschko *et al.* (2013)). Figure 6.15 compares different versions of `EdgeBoxes` tuned to

	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
EdgeBoxesAR	69.6	78.3	66.2	58.6	42.5	82.1	78.1	83.0	42.7	74.6	66.4	81.1	82.0	74.5	68.3	35.1	66.1	68.7	75.2	62.6	67.8
+ gt oracle	75.5	79.4	70.6	63.1	55.0	82.6	84.3	83.9	46.6	75.1	68.1	82.5	83.3	75.9	76.8	41.2	67.6	70.1	77.3	65.7	71.2
+ NMS oracle	77.2	87.1	76.6	67.6	48.2	84.8	85.2	87.1	52.1	83.9	72.7	86.7	87.2	84.2	77.6	44.2	75.1	73.5	83.5	65.4	75.0
+ both oracles	83.7	87.8	79.6	72.6	61.6	85.1	88.2	87.8	55.7	84.0	74.8	87.3	87.5	84.9	86.8	50.3	76.2	74.6	85.3	67.6	78.1

Table 6.5: Fast R-CNN (model L) detection results on PASCAL 2007 test using EdgeBoxesAR and given access to "oracles" that provide additional information to the detector. Given access to both oracles, the only way to further improve detector performance would be to avoid proposals on background or to learn a more discriminative classifier. See text for details.

maximise recall at different IoU points Δ (we set $\alpha = \max(0.65, \Delta - 0.05)$, $\beta = \Delta + 0.05$, see Zitnick and Dollár (2014) for details). `EdgeBoxes` tuned for $\Delta = 0.70$ or 0.75 maximises AR and also results in the best detection results.

While originally `EdgeBoxes` allowed for optimising recall for a particular IoU threshold, we consider a new variant that directly maximises AR (marked 'AR' in figure 6.15) to further explore the link between AR and detection quality. To do so, we alter its greedy NMS procedure to make it *adaptive*. We start with a large NMS threshold β_0 to encourage dense sampling around the top scoring candidates (a window is suppressed if its IoU with a higher scoring window exceeds this threshold). After selecting each proposal, β_{k+1} is decreased slightly via $\beta_{k+1} = \beta_k \cdot \eta$ to encourage greater proposal diversity. Setting $\beta_0 = 0.90$ and $\eta = 0.9996$ gave best AR at 1 000 proposals on the PASCAL validation set (we kept $\alpha = 0.65$ fixed). This new adaptive `EdgeBoxes` variant is not optimal at any particular IoU threshold, but has best overall AR and improves Fast R-CNN mAP by 1.6 over the best previous variant (reaching 62.0 mAP).

The results in figure 6.15 further support our conclusion that AR is a good predictor for mAP and suggest that it can be used for fine-tuning proposal methods. We expect other methods to improve as well if optimised for AR instead of a particular IoU threshold.

6.5.6 Detection with oracles

We finish by exploring the limits of proposal methods when coupled with Fast R-CNN and given access to "oracles" that provide additional information to the detector. For these experiments we use the `EdgeBoxesAR` proposals described in section 6.5.5 which gave the best results of all evaluated methods when coupled with the Fast R-CNN model M. Re-training the larger model L with `EdgeBoxesAR` proposals improves mAP to 67.8 (compared to 66.7 using `SelectiveSearch` proposals as in Girshick (2015)).

We test two oracles. First, we augment the set of proposals with all ground truth annotations (*gt* oracle), which results in AR of 1 (but contains many false positives). Second, we perform optimal, per-class non-maximum suppression (*NMS* oracle) that suppresses all false positives that overlap any true positives (without suppressing any true positives, and keeping false positives in the background). Results for the gt and NMS oracles are shown in table 6.5.

The gt oracle improves mAP by about 3%. The NMS oracle has the overall stronger effect with about 7% mAP improvement. Combining both oracles improves mAP by about 10%, indicating that their effect is largely orthogonal. All remaining mistakes that prevent perfect detection are confusions on the background or misclassifications. Therefore, the only way to further improve detector performance would be to avoid proposals on background or to learn a more discriminative classifier.

6.6 Discussion

In this work we have revisited the majority of existing detection proposal methods, proposed new evaluation metrics, and performed an extensive and direct comparison of existing methods. Our primary goal has been to enable practitioners to make more informed decisions when considering use of detection proposals and selecting the optimal proposal method for a given scenario. Additionally, our open source benchmark will enable more complete and informative evaluations in future research on detection proposals. We conclude by summarising our key findings and suggesting avenues for future work.

Repeatability. We found that the *repeatability* of virtually all proposal methods is limited: imperceptibly small changes to an image cause a noticeable change in the set of produced proposals. Even changing a single image pixel already exhibits measurable differences in repeatability. We foresee room for improvement by using more robust superpixel (or boundary estimation) methods. However, while better repeatability for object detection would be desirable, it is not the most important property of proposals. Image independent methods such as `SlidingWindow` and `CrackingBing` have perfect repeatability but are inadequate for detection. Methods such as `SelectiveSearch` and `EdgeBoxes` seem to strike a better balance between recall and repeatability. We suspect that high quality proposal methods that are also more repeatable would yield improved detection accuracy, however this has yet to be verified experimentally.

Localisation Accuracy. Our analysis showed that for object detection improving proposal *localisation accuracy* (improved IoU) is as important as improving recall. Indeed, we demonstrated that the popular metric of recall at IoU of 0.5 is not predictive of detection accuracy. As far as we know, our experiments are the first to demonstrate this. Proposals with high recall but at low overlap are not effective for detection.

Average Recall. To simultaneously measure both proposal recall and localisation accuracy, we report *average recall* (AR), which summarises the distribution of recall across a range of overlap thresholds. For a fixed number of proposals, AR correlates surprisingly well with detector performance (for LM-LLDA, R-CNN, and Fast R-CNN). AR proves to be an excellent predictor of detection performance both for comparing competing methods as well as tuning a specific method's parameters. We encourage future work to report average recall (as shown in figures 6.7c/6.8c) as the primary metric for evaluating proposals for object detection. For detectors more robust to localisation errors (e.g. Fast R-CNN), the IoU range of the AR metric can be modified to better predict detector performance.

Top Methods. Amongst the evaluated methods, `SelectiveSearch`, `Rigor`, `MCG`, and `EdgeBoxes` consistently achieved top object detection performance when coupled with diverse object detectors. If fast proposals are required, `EdgeBoxes` provides a good compromise between speed and quality. Surprisingly, these top methods all achieve fairly similar detection performance even though they employ very different mechanisms for generating proposals. SelectiveSearch merges superpixels, Rigor computes multiple graph cut segmentations, MCG generates hierarchical segmentations, and EdgeBoxes scores windows based on edge content.

Generalisation. Critically, we measured no significant drop in recall when going from the 20 PASCAL categories to the 200 ImageNet categories. Moreover, while MS COCO is substantially harder and has very different statistics (more and smaller objects), relative method ordering remains mostly unchanged. These are encouraging result indicating that *current methods do indeed generalise to different unseen categories*, and as such can be considered true "objectness" methods.

Oracle Experiments. The best Fast R-CNN results reported in this chapter used the large model L and EdgeBoxesAR proposals, achieving mAP of 67.8 on PASCAL 2007 test. Using an oracle to rectify all localisation and recall errors improved performance to 71.2 mAP, and adding an oracle for perfect non-maximum suppression further improved mAP to 78.1 (see section 6.5.6 for details). The remaining gap of 21.9 mAP to reach perfect detection is caused by high scoring detections on the background and object misclassifications. This best case analysis for proposals that are perfectly localised shows that further improvement can only be gained by removing false positives in the proposal stage (producing fewer proposals while maintaining high AR) or training a more discriminative classifier.

Discussion. Do object proposals improve detection quality or are they just a transition technology until we have sufficient computing power? On the one hand, simply increasing the number of proposals, or using additional random proposals, may actually harm detection performance as shown in Girshick (2015). On the other hand, there is no fundamental difference between the pipeline of object proposals with a detector and a cascaded detector with two stages. Conceptually, a sliding window detector with access to the features of the proposal method may be able to perform at least as well as the cascade and as such detection proposals independent of the final classifier may eventually become unnecessary. Given enough computing power and an adequate training procedure, one might expect that a dense evaluation of CNNs could further improve performance over R-CNNs.

While in this work we have focused on object detection, object proposals have other uses. For example, they can be used to handle unknown categories at test time, or to enable weakly supervised learning (Vicente *et al.*, 2011; Guillaumin *et al.*, 2014; Tang *et al.*, 2014).

Finally, we observe that current proposal methods reach high recall while using features that are not utilised by detectors such as LM-LLDA, R-CNN, and Fast R-CNN (e.g. object boundaries and superpixels). Conversely, with the exception of Multibox (Erhan *et al.*, 2014), none of the proposal methods use CNN features. We expect some cross-pollination will occur in this space. Indeed, there has been some recent work in this space (Kuo *et al.*, 2015; Ren *et al.*, 2015; Pinheiro *et al.*, 2015) that shows promising results.

In the future, detection proposals will surely improve in repeatability, recall, localisation accuracy, and speed. Top-down reasoning will likely play a more central role as purely bottom-up processes have difficulty generating perfect object proposals. We may also see a tighter integration between proposals and the detector, and the segmentation mask generated by many proposal methods may play a more important role during detection. One thing is clear: progress has been rapid in this young field and we expect proposal methods to evolve quickly over the coming years.

6.A Analysis of Metrics

Average recall (AR) between 0.5 and 1 can also be computed by averaging over the overlaps of each annotation gt_i with the closest matched proposal, that is integrating over the y axis of the plot instead of the x axis. Let o be the IoU overlap and $\mathrm{recall}(o)$ the function shown for example in figure 6.6b. Let $\mathrm{IoU}(\mathrm{gt}_i)$ denote the IoU between the annotation gt_i and the closest detection proposal. We can then write:

$$\mathrm{AR} = 2\int\limits_{0.5}^{1} \mathrm{recall}(o)\,\mathrm{d}o = \frac{2}{n}\sum_{i=1}^{n} \max\left(\mathrm{IoU}(\mathrm{gt}_i) - 0.5, 0\right)$$

which is the same as the *average best overlap* (ABO) (Carreira and Sminchisescu, 2012) or the average *best spatial support* (BSS) (Malisiewicz and Efros, 2007) truncated at 0.5 IoU.

The ABO and BSS are typically computed by assigning the closest proposal to each annotation, i.e. a proposal can match more than one annotation. In contrast, for all our experiments we compute a bipartite matching to assign proposals to annotations (using a greedy algorithm for efficiency instead of the optimal Hungarian algorithm).

The *volume-under-surface* metric (VUS) (Manén *et al.*, 2013) plots recall as a function of both overlap and proposal count and computes the volume under that surface. Since in practice detectors utilize a fixed number of proposals, the VUS of a proposal method is only an indirect predictor of detection accuracy.

7 A convnet for improving non-maximum suppression

Non-maximum suppression (NMS) is used in virtually all state-of-the-art object detection pipelines. While essential object detection ingredients such as features, classifiers, and proposal methods have been extensively researched surprisingly little work has aimed to systematically address NMS. The de-facto standard for NMS is based on greedy clustering with a fixed distance threshold, which forces to trade-off recall versus precision. In chapter 6, we estimated that 25% of the remaining improvement on Pascal VOC could be gained with perfect NMS (see section 6.5.6).

In this chapter, we propose a convnet designed to perform NMS of a given set of detections. We report experiments on a synthetic setup, crowded pedestrian scenes, and for general person detection. Our approach overcomes the intrinsic limitations of greedy NMS, obtaining better recall and precision. This work has been published at GCPR (Hosang *et al.*, 2016b) with an oral presentation. Jan Hosang was the lead author and contributed all experiments.

7.1 Introduction

The bulk of object detection pipelines are based on three steps: 1) propose a set of windows (via sliding window or object proposals), 2) score each window via a trained classifier, 3) remove overlapping detections (non-maximum suppression). DPM (Felzenszwalb *et al.*, 2010) and R-CNN (Girshick *et al.*, 2014; Girshick, 2015; Ren *et al.*, 2015) follow this approach. Both object proposals (see chapter 6) and detection classifiers (Russakovsky *et al.*, 2015) have received enormous attention, while non-maximum suppression (NMS) has been seldom addressed. The de-facto standard for NMS consists of greedily merging the higher scoring windows with lower scoring ones if they overlap enough (e.g. intersection-over-union $\text{IoU} > 0.5$), which we call GreedyNMS in the following.

GreedyNMS is popular because it is conceptually simple, fast, and for most tasks results in satisfactory detection quality. Despite its popularity, it has important shortcomings. As illustrated in figure 7.2, GreedyNMS trades off precision versus recall. If the IoU threshold is too large (too strict) then not enough surrounding detections are suppressed, high scoring false positives are introduced and precision suffers. If the IoU threshold is too low (too loose) then multiple true positives are merged together and the recall suffers. For any IoU threshold, GreedyNMS is sacrificing precision or recall (as shown experimentally in section 7.4). One can do better than this by leveraging the

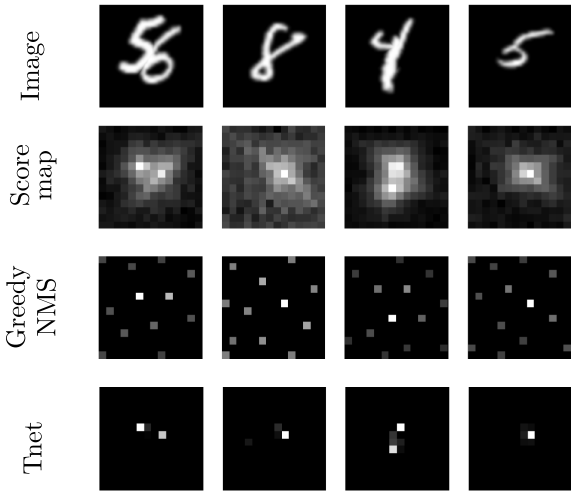


Figure 7.1: GreedyNMS produces false positives and prunes true positives, while our proposed Tnet correctly localize even very close digits. First to last row: oMNIST image, input score map, GreedyNMS IoU > 0.3, and Tnet IoU & S(1, 0 → 0.6).

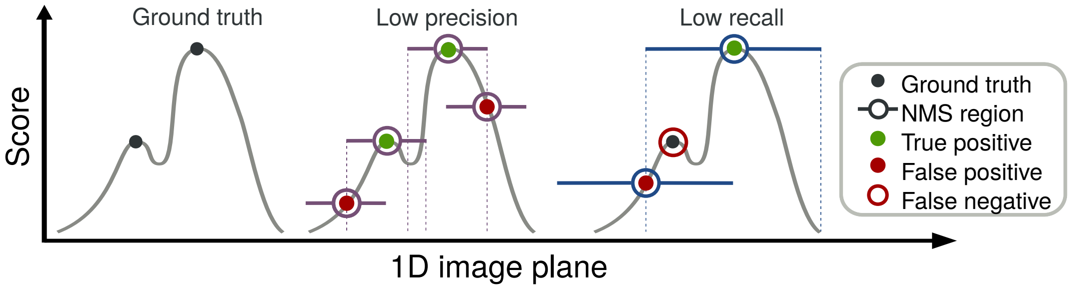


Figure 7.2: 1D illustration of the GreedyNMS shortcomings. Black dots indicate true objects, grey curve is the detector response, green dots are true positives, red dots/circles are false positives/negatives.

full signal of the score map (statistics of the surrounding detections) rather than blindly applying a fixed policy everywhere in the image.

Current object detectors are becoming surprisingly effective on both general (e.g. Pascal VOC, COCO) and specific object detection (e.g. pedestrians, faces). The oracle analyses for “perfect NMS” from table 6.5 and Parikh and Zitnick (2011, figure 12) both indicate that NMS accounts for almost a quarter of the remaining mistakes.

Instead of doing hard pruning decisions as GreedyNMS, we design our network to make soft decisions by re-scoring (re-ranking) the input detection windows. Our re-scoring is final, and no post-processing is done afterwards, thus the resulting score maps must be very “peaky”. We call our proposed network “Tyrolean network”, abbreviated Tnet. (Tyrolean because “it likes to see peaks”.)

Contribution. We are the first to show that a convnet can be trained and used to overcome the limitations of GreedyNMS. Our experiments demonstrate that, across different occlusion levels, the Tyrolean network (Tnet) performs strictly better than GreedyNMS at *any* IoU threshold.

As an interesting scenario for NMS, we report results for crowded pedestrian scenes and general person detection. Our Tnet can operate solely over detection boxes (like GreedyNMS), and does not use external training data. Furthermore, Tnet provides better results than auto-context (Tu and Bai, 2010). We consider our results a proof of concept, opening the door for further exploration.

7.2 Base Tyrolean network

The main intuition behind our proposed Tyrolean network (Tnet) is that the score map of a detector together with a map that represents the overlap between neighbouring detections contains valuable information to perform better NMS than GreedyNMS (see figure 7.1, second row). Our network is a traditional convnet but with access to two slightly unusual inputs (described below), namely score map information and IoU maps. Figure 7.3 shows the overall network. In our base Tnet the first stage applies 512 11×11 filters over each input layer, and 512 1×1 filters are applied on layers 2 and 3. ReLU non-linearities are used after each layer but the last one. Neither max-pooling nor local normalization is used.

The base network is trained and tested in a fully convolutional fashion. It uses the same information as GreedyNMS, and does not access the image pixels directly. The required training data are only a set of object detections (before NMS), and the ground truth bounding boxes of the dataset. We focus on the single class case and consider exploiting multi-class information future work.

Input grid. As preprocessing all detections in an image are mapped into a 2d grid (based on their centre location). If more than one detection falls into the same cell, we keep only the highest scoring detection. Each cell in the grid is associated with a detection bounding box and score. We use cells of 4×4 pixels, thus an input image of size $\mathrm{W} \times \mathrm{H}$ will be mapped to input layers of size $\mathrm{w} \times \mathrm{h} = {}^{\mathrm{W}}/_{4} \times {}^{\mathrm{H}}/_{4}$. Since the

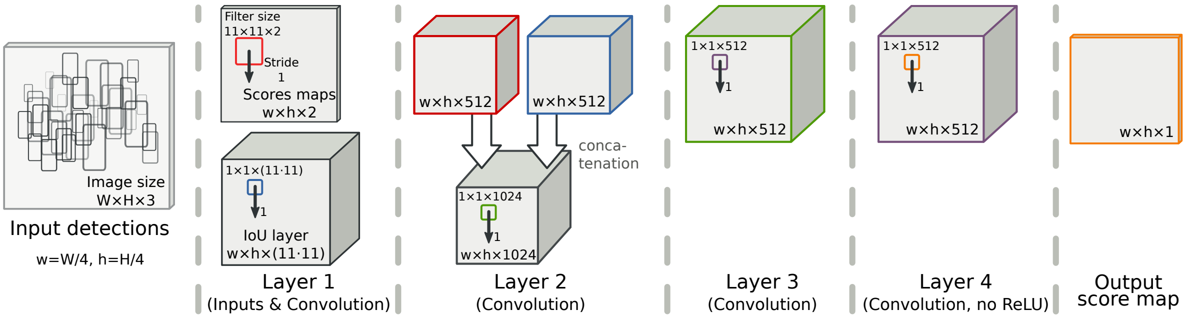


Figure 7.3: Base architecture of our Tyrolean network (Tnet). Each box is a feature map, its dimensions are indicated at its bottom, the coloured square indicates the convolutional filters size, the stride is marked next to the downward arrow.

cells are small, mapping detections to the input grid has minimal impact on the NMS procedure. In preliminary experiments we validated that: a) we can at least recover the performance of GreedyNMS (applying GreedyNMS over the input grid provides the same results as directly applying GreedyNMS), b) the detection recall stays the same (after mapping to the input grid the overall recall is essentially identical to the raw detections).
This incarnation of Tnet can handle mild changes in scale amongst neighbouring detections. Section 7.4 reports experiments with detections over a 3× scale range. In section 7.4 we also explain how to adapt our approach to general person detection (Pascal VOC (Everingham *et al.*, 2014)), with large scale and aspect ratio variance.

IoU layer. In order to reason about neighbouring detection boxes (or segments) we feed Tnet with IoU values. For each location we consider a $11 \times 11 = 121$ neighbourhood, thus the input IoU layer has $\mathrm{w} \times \mathrm{h} \times 121$ values. Together the cell size and neighbourhood size should provide the Tnet with sufficient information about surroundings of a detection, where this choice depends on the object sizes in the image and the expected object density and thus are application dependent.

Score maps layer. To reason about the detection confidence, we feed Tnet with the raw detection score map (once mapped to the input grid). The NMS task involves ranking operations which are not easily computed by linear and ReLU $(\max(\cdot, 0))$ operators. To ease the task we also feed the Tnet with score maps resulting from GreedyNMS at multiple IoU thresholds. All score maps are stacked as a multi-channel input image and feed into the network. $\mathrm{S}(\tau)$ denotes a score map resulting from applying GreedyNMS with IoU$\geq \tau$, $\mathrm{S}(\tau_1, \tau_2)$ denotes a two channels map ($\mathrm{S}(\tau_1)$ and $\mathrm{S}(\tau_2)$ stacked). Note that $\mathrm{S}(1)$ returns the raw detection score map. Our base Tnet uses $\mathrm{S}(1, 0.3)$ which has dimensionality $\mathrm{w} \times \mathrm{h} \times 2$ (see figure 7.3). The convolutional filters applied over the score maps input have the same size as the IoU layer neighbourhood (11×11 cells).

Tnet is then responsible for interpreting the multiple score maps and the IoU layer, and make the best local decision. Our Tnet operates in a fully feed-forward convolutional manner. Each location is visited only once, and the decision is final. In other words, for each location the Tnet has to decide if a particular detection score corresponds to a correct detection or will be suppressed by a neighbouring detection in a single feed-forward path.

Parameter rules of thumb. Figure 7.3 indicates the base parameters used. Preliminary experiments indicated that removing top layers has a clear negative impact on the network performance, while the width of these layers is rather insensitive. Having a high enough resolution in the input grid is critical, while keeping a small enough number of convolutions over the inputs allows to keep the number of model parameters under control. During training data augmentation is necessary to avoid overfitting. The training procedure is discussed in section 7.2.2, while experimental results for some parameters variants are reported in section 7.4.

Input variants. Experiments in the next sections consider multiple input variants. The IoU layer values can be computed over bounding boxes (regressed by the sliding window detector) or over estimated instance segments (Pinheiro *et al.*, 2015). Similarly, for the score maps we consider different numbers of GreedyNMS thresholds, which changes the dimensionality of the input score map layer.

In all cases we expect the Tnet to improve over a fixed threshold GreedyNMS by discovering patterns in the detector score maps and IoU arrangements that enable to do adaptive NMS decisions.

7.2.1 Handling scale and aspect ratio

Big scale differences, such as on Pascal VOC, are a problem with the fully convolutional architecture introduced above, because we have a fixed convolutional filter size that effectively gives the system a fixed size context to take into account. How big should this context be? In the case of relatively small pedestrians in PETS and ParkingLot it turns out that a context of 44×44 pixels is sufficient. However in Pascal two people can be almost as big as the entire image, the centre points of their bounding boxes can be several hundred pixels apart, so the context needs to be much bigger in that case.

Input grid. The idea to remedy this issue is to adapt the neighbourhood size to the size and aspect ratio of the detection that is to be rescored, so big objects have a larger neighbourhood than small objects. Since we want to use the same model for big and small objects, the representation has to have the fixed size, so we use an 11×11 grid to represent the neighbourhood (defined to be twice the size of the object) just like in the ordinary Tnet. In general detections in the image have different sizes, requiring input grids of different resolutions. We decide to switch to a detection-centric representation and generate an input grid for each detection individually, which is feasible because the Faster R-CNN outputs relatively few detections that were already processed individually (as opposed to fully convolutionally). The assignment of detections to the input grid is done as usual by picking that maximum scoring detection for which the centre falls into each grid cell.

7.2.2 Training procedure

Typically detectors are trained as classifiers on a set of positive and negative windows, determined by the IoU between detection and object annotation. When doing so the spatial relation between detector outputs and the annotations is neglected. We adopt the idea from (Stewart and Andriluka, 2016) of computing the loss by matching detections to annotations, and train the network to predict new detection scores that are high for matched detections and low everywhere else. In contrast to the conventional wisdom of training the detector to have a smooth score decrease around positive instances, we declare a detection right next to a true positive to be a negative training sample. Processing detections *independently* would hurt generalisation, but Tnet has access to neighbouring detections circumventing this problem. This is necessary because our network must itself perform NMS.

Training loss. Our goal is to reduce the score of all detections that belong to the same person, except exactly one of them. To that end, we match every annotation to the highest scoring detection that overlaps at least 0.5 IoU. This determines the set of positives, while all other detections are negative training examples. This yields a label y_p for every location p in the input grid (see previous section). Since background detections are much more frequent than true positives, it is necessary to weight the loss terms to balance the two. We use the weighted logistic loss and choose the weights so that both classes have the same weight per frame. We also consider setting weights to balance classes across the full dataset and giving lower weights for highly occluded samples, see section 7.4.1.

The model is trained from scratch, randomly initialized with MSRA (He *et al.*, 2015), and optimized via Adam (Kingma and Ba, 2015). All experiments are implemented with Caffe (Jia *et al.*, 2014).

As pointed out in (Mathias *et al.*, 2014) the threshold for GreedyNMS requires to be carefully selected on the validation set of each task, the commonly used default IoU > 0.5 can severely underperform. Other NMS approaches such as (Tang *et al.*, 2015b; Rothe *et al.*, 2014) also require training data to be adjusted. When maximizing performance in cluttered scenes is important, training a Tnet is thus not a particularly heavy burden. Training our base Tnet on un-optimized CPU and GPU code takes a day.

7.3 Controlled setup experiments

NMS is usually the last stage of an entire detection pipeline. Therefore, in an initial set of experiments, we want to understand the problem independent of a specific detector and abstract away the particularities of a given dataset.

If objects appeared alone in the images, NMS would be trivial. The core issue for NMS is deciding if two local maxima in the detection score map correspond to one or multiple objects. To investigate this core aspect we create the oMNIST ("overlapping

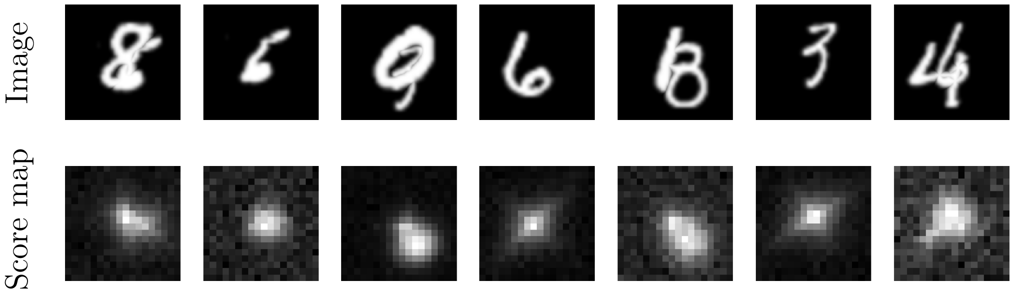


Figure 7.4: Example data from our controlled experiments setup. The convnet must decide if one or two digits are present (and predict is their exact location) while using only a local view of score and IoU maps (no access to the input image).

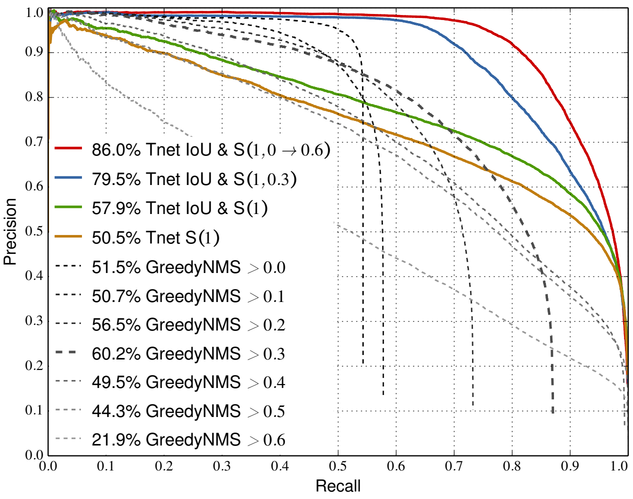


Figure 7.5: oMNIST test set detection results.

Method	AR
GreedyNMS	
bboxes IoU > 0.3	54.3%
DeepMask segments	52.0%
Tnet variants	
IoU & S(1, 0→0.6)	*59.6%*
IoU & S(1, 0.3)	57.9%
IoU & S(1)	36.5%
S(1)	33.9%

Table 7.1: PETS validation set results. Base Tnet is underlined.

MNIST") toy dataset. This data does not aim at being particularly realistic, but rather to enable a detailed analysis of the NMS problem.

Each image is composed of one or two off-centre MNIST digits with IoU $\in [0.2, 0.6]$. We mimic a detector by generating synthetic perturbed score maps. Albeit noisy, the detector is "ideal" because its detection score remains high despite strong occlusions. Figure 7.1 and 7.4 show examples of the generated score maps and corresponding images. By design GreedyNMS will have difficulties handling such cases (at any IoU threshold). We generate a training/test split of 100k/10k images (fix across experiments).

Other than score maps our convnet uses IoU information between neighbouring detections (like GreedyNMS). Our experiments cover using the perfect segmentation masks for IoU (ideal case), noisy segmentation masks, and the sliding window bounding boxes.

7.3.1 Results

Results are summarised in figure 7.5. Curves are scored via AR; the average recall on the precision range [0.5, 1.0]. The evaluation is done using the standard Pascal VOC protocol, with IoU > 0.5 (Everingham *et al.*, 2014).

GreedyNMS. As can be seen in figure 7.5 varying the IoU thresholds for GreedyNMS trades off precision and recall. The best AR that can be obtained with GreedyNMS is 60.2% for IoU > 0.3. Example score maps for this method can be found in figure 7.1, third row.

Upper bound. As an upper bound for any method relying on score map information we calculate the overlap between neighbouring hypotheses based on perfect segmentation masks (available in this toy scenario). With perfect overlaps and perfect scores GreedyNMS returns perfect results. Based on our idealized but noisy detection score maps the upper bound reaches 90.0% AR. In section 7.4 we report experiments using segmentation masks estimated from the image, which results in inferior performance.

Base Tnet. Using the same information as GreedyNMS with bounding boxes, our base Tnet reaches better performance for the entire recall range (see figure 7.5), S(1, 0.3) indicates the score maps from GreedyNMS with IoU > 0.3 and ≥ 1, i.e. the raw score map. In this configuration Tnet obtains 79.5% AR, clearly superior to GreedyNMS. This shows that, at least in a controlled setup, a convnet can indeed exploit the available information to overcome the limitations of the popular GreedyNMS method.

Instead of picking a specific IoU threshold to feed Tnet, we consider IoU & S($1, 0 \rightarrow 0.6$), which includes S(1, 0.6, 0.4, 0.3, 0.2, 0.0). As seen in figure 7.5, not selecting a specific threshold results in the best performance; 86.0% AR. If we remove GreedyNMS score maps and only provide the raw score map (IoU & S(1)) performance decreases significantly. As soon as some ranking signal is provided (via GreedyNMS score maps), our Tnet is able to learn how to exploit best the information available. Qualitative results are presented in figure 7.1, bottom row.

Auto-context. Importantly we show that IoU & S(1) improves over S(1) only. (S(1) is the information exploited by auto-context methods, mentioned in section 2.4). This shows that the convnet is learning to do more than simple auto-context. The detection improves not only by noticing patterns on the score map, but also on how the detection boxes overlap.

7.4 Person detection experiments

After the proof of concept in a controlled setup, we move to a realistic pedestrian detection setup. We are particularly interested in datasets that show diverse occlusion where NMS is non-trivial. We decided for the PETS dataset (Ferryman and Ellis, 2010), which exhibits diverse levels of occlusion and provides a reasonable volume of training and test data. We use 5 sequences for training, one sequence for validation and testing (23k, 4k, 10k annotations respectively). PETS has been previously used to study person detection (Tang *et al.*, 2013), tracking (Milan *et al.*, 2014), and crowd density estimation (Subburaman *et al.*, 2012). Additionally we test the generalization of the trained model on the ParkingLot dataset (Shu *et al.*, 2012), and the applicability to general person detections on Pascal VOC (Everingham *et al.*, 2014). Figure 7.10 shows example frames.

Standard pedestrian datasets such as Caltech (Dollár *et al.*, 2012b) or KITTI (Geiger *et al.*, 2012) average less than two pedestrians per frame, making close-by detections a rare occurrence. In PETS and ParkingLot $> 50\%$ of pedestrians have some occlusion, and about $\sim 20\%$ have significant occlusion (IoU > 0.4). Pascal presents fewer occlusion cases, people being the class where it is most frequent.

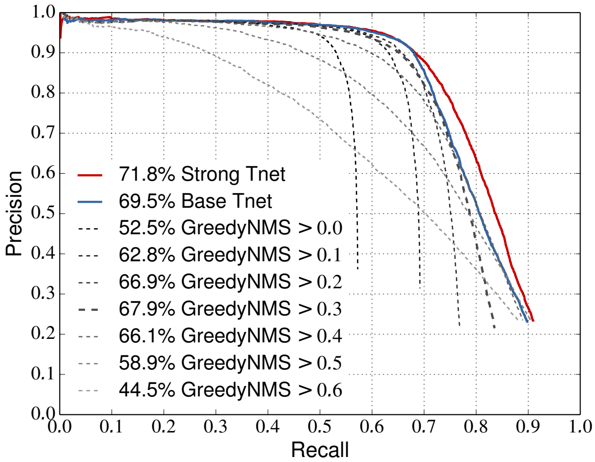


Figure 7.6: Detection results on PETS test set. Our approach is better than any GreedyNMS threshold and better than the upper envelope of all GreedyNMS curves.

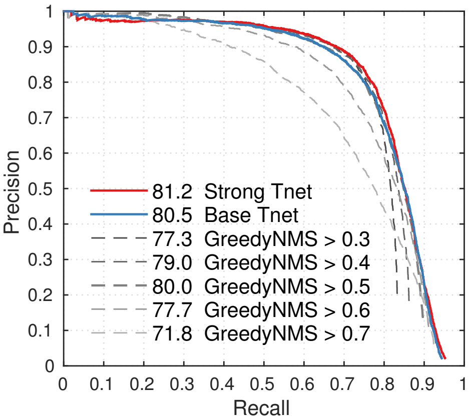


Figure 7.7: Person detection results on Pascal'12 test set. Tnet improves over all GreedyNMS thresholds.

Person detector. In this work we take the detector as a given. For the PETS experiments we use the baseline DPM detector from Tang *et al.* (2013). We are not aware of a detector (convnet or not) providing better results on PETS-like sequences (we considered some of the top detectors in Dollár *et al.* (2012b)). Importantly, for our exploration the detector quality is not important per-se. As discussed in section 7.3 GreedyNMS suffers from intrinsic issues, even when providing an idealized detector. In fact Tnet benefits from better detectors, since there will be more signal in the score maps. We thus consider our DPM detector a fair detection input. We use the DPM detections after bounding box regression, but before any NMS processing.

Person segments. In section 7.4.1 we report results using segments estimated from the image content. We use our re-implementation of DeepMask (Pinheiro *et al.*, 2015), trained on the Coco dataset (Lin *et al.*, 2014). We use DeepMask as a realistic example of what can be expected from modern techniques for instance segmentation.

7.4.1 Results

Our PETS results are presented in table 7.1 (validation set) and figure 7.6 (test set). Qualitative results are shown in figure 7.10.

Boxes. Just like in the oMNIST case, the GreedyNMS curves in figure 7.6 have a recall versus precision trade-off. We pick IoU > 0.3 as a reference threshold.

Segments. GreedyNMS should behave best when the detection overlap is based on the visible area of the object. We compute DeepMask segments over DPM detection, feed these in GreedyNMS, and select the best IoU threshold for the validation set. Table 7.1 shows results slightly below the bounding boxes case. Although many segments are

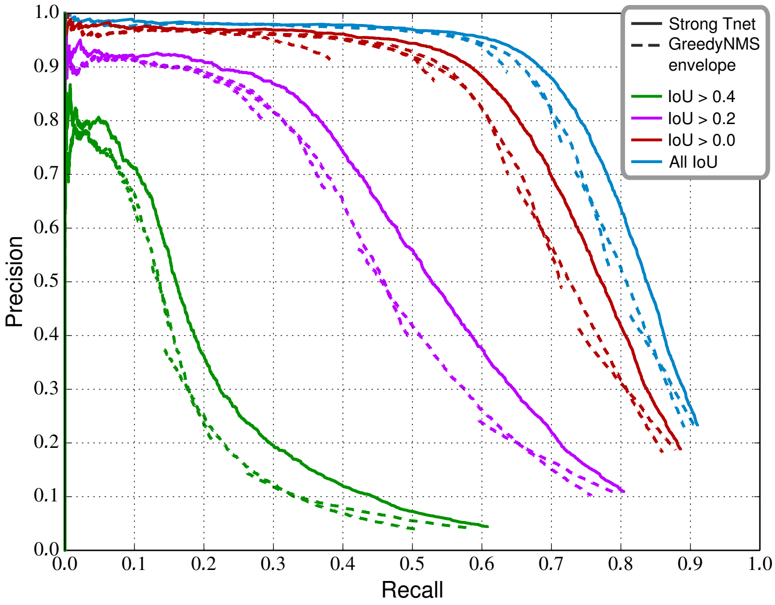


Figure 7.8: GreedyNMS versus Strong Tnet when evaluated over different subsets of PETS test data, based on level of occlusion. In each subset our Tnet improves over the upper envelope of all GreedyNMS threshold curves.

rather accurate, they drop in quality when heavier occlusion is present. In theory using segments should improve GreedyNMS, in practice they hurt more than they help.

Auto-context. For the S (1) entry in table 7.1 only the raw detection score map is feed to Tnet (same nomenclature as section 7.3.1). Since performance is lower than other variants (e.g. IoU & S (1)), this shows that our approach is exploiting available information better than just doing auto-context over DPM detections.

Tnet. Both in validation and test set our trained network with IoU & S (1, 0.3) input provides a clear improvement over vanilla GreedyNMS. Just like in the oMNIST case, the network is able to leverage patterns in the detector output to do better NMS than the de-facto standard GreedyNMS method.

Table 7.1 reports the results for a few additional variants. IoU & S(1, 0 → 0.6) shows that it is not necessary to select a specific IoU threshold for the input score map layer. Given an assortment (S(1, 0.6, 0.4, 0.3, 0.2, 0.0)) the network will learn to leverage the information available.

Using a relaxed loss that decreases weight of hard examples (peaks on the background and strong occlusions) helps further improve the results, moving from 57.9% to 58.9% AR. Weighting classes equally over the full dataset (global weighting) instead of frame-by-frame gives a mild improvement from 57.9% to 58.0% AR.

Strong Tnet. We combine the best ingredients identified on the validation set into one strong model. We use IoU & S(1, 0→0.6), relaxed loss, and global weighting. Figure

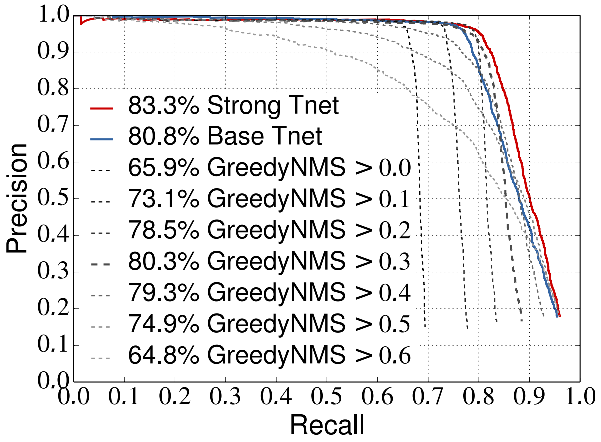


Figure 7.9: Detection results on the ParkingLot dataset. Tnet is better than any GreedyNMS threshold, even though it has been trained using PETS data only.

7.6 shows that we further improve over the base Tnet from 59.5% to 71.8% AR on the PETS test set. The gap between base Tnet and GreedyNMS is smaller on the test set than on validation, because test set has lighter occlusions. Still our strong Tnet provides a consistent improvement over GreedyNMS.

Figure 7.8 provides a more detailed view of the results from figure 7.6. It compares our strong Tnet result versus the upper envelope of GreedyNMS over all thresholds ([0, 1]), when evaluated over different subsets of the test set. Each subset corresponds to ground truth bounding boxes with other boxes overlapping more than a given IoU level. For all ranges, our strong Tnet improves over GreedyNMS. This shows that our network does not fit to a particular range of occlusions, but learns to handle all of them with comparable effectiveness.

At test time Tnet takes ∼200 milliseconds per frame (all included).

ParkingLot results. To verify that our Tnet can generalize beyond PETS, we run the same DPM detector as on the PETS experiment over the ParkingLot sequence and do NMS using the networks trained on PETS training set only. Results in figure 7.9 show that Tnet improves from 80.3% to 83.3% AR over the best GreedyNMS threshold of IoU > 0.3. Even though Tnet was not trained on this sequence we see a similar result as on the PETS dataset. Not only does our Strong Tnet improve over the best GreedyNMS result, but it improves over the upper envelope of all GreedyNMS thresholds (similar trend as figure 7.6).

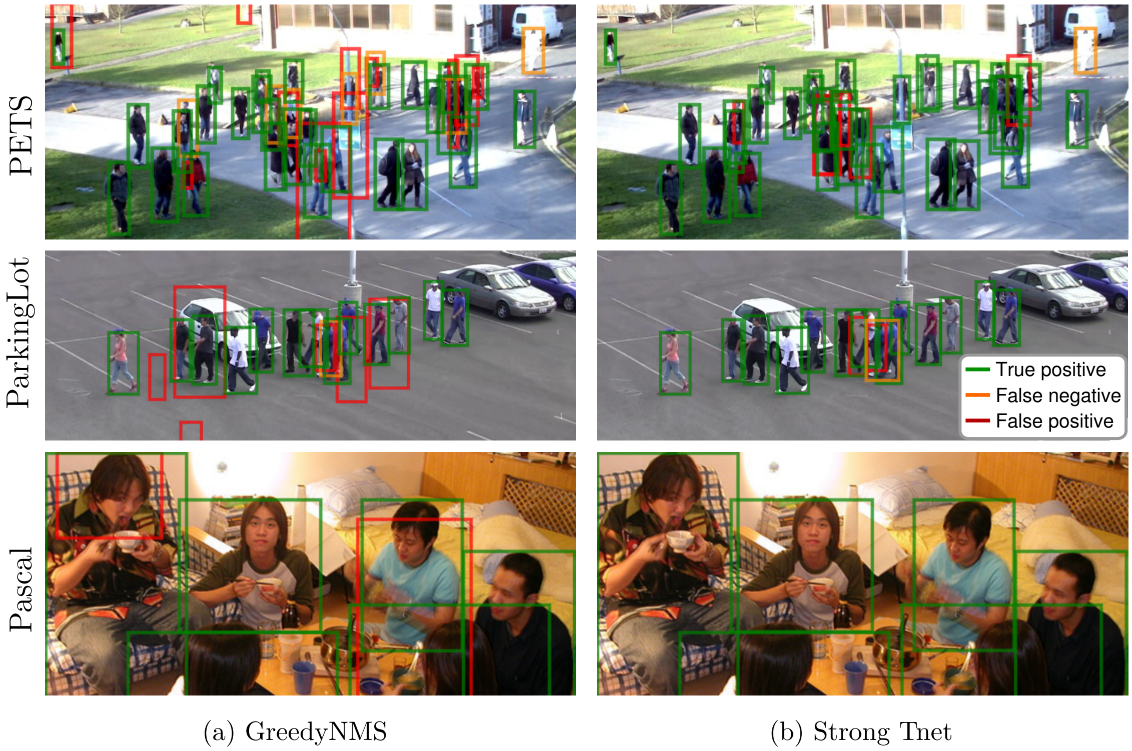
Pascal results. Pascal VOC (Everingham *et al.*, 2014) contains less occlusion but is more challenging with respect to appearance, scale, and aspect ratio variance. We focus on the "people" class which offers the highest diversity in occlusion. As a base detector we use the publicly available Faster R-CNN (Ren *et al.*, 2015). The small variance in performance of the GreedyNMS swipe in figure 7.7 shows that this data contains fewer occlusions than PETS.

Tnet is trained on Pascal '07 trainval and tested on the test set. To adapt the Tnet to multiple scales and aspect ratio we switch from the fully convolutional approach to a detection-centric representation. Instead of a fixed-size neighbourhood grid we adapt its scale and aspect ratio to each detection box being re-scored. We also use the image features. After tuning the training parameters, Base Tnet matches the best GreedyNMS with 80.5% AP (figure 7.7). Strong Tnet matches the upper envelope of all GreedyNMS thresholds, improving the results to 81.2% AP.

7.5 Conclusion

We have discussed the limitations of GreedyNMS in detail and presented experiments showing its recall versus precision trade-off. For the sake of speed and simplicity GreedyNMS disregards most of the information available in the detector response. Our proposed Tyrolean network (Tnet) mines the patterns in the score map values and bounding box arrangements to surpass the performance of GreedyNMS. On the person detection task, our final results show that our approach provides, compared to any GreedyNMS threshold, both high recall and improved precision. These results confirm that Tnet can overcome the intrinsic limitations of GreedyNMS, while keeping practical test time speeds. We consider the reported results a proof of concept, opening the door for further extensions.

Current detection pipelines consist of a convnet and a hard-coded NMS procedure. Replacing the NMS with a Tnet opens the possibility of true end-to-end training of object detectors and we reckon that significant improvements can be obtained by replacing NMS with a Tnet.



(a) GreedyNMS

(b) Strong Tnet

Figure 7.10: Qualitative detection results of GreedyNMS and Strong Tnet (both operating at same recall). Tnet is able to suppress false positives as well as recover recall that is lost with GreedyNMS.

8 Learning non-maximum suppression

Object detectors have hugely profited from moving towards an end-to-end learning paradigm: proposals, features, and the classifier becoming one neural network improved results two-fold on general object detection. One indispensable component is non-maximum suppression (NMS), a post-processing algorithm responsible for merging all detections that belong to the same object. The de facto standard NMS algorithm is still fully hand-crafted, suspiciously simple, and — being based on greedy clustering with a fixed distance threshold — forces a trade-off between recall and precision. We propose a new network architecture designed to perform NMS (using only boxes and their score). We report experiments for person detection on PETS and COCO datasets. Our approach shows promise providing improved localization and occlusion handling.

This work will be published at CVPR 2017 and presented as a spotlight. Jan Hosang was the lead author and contributed all experiments.

8.1 Introduction

All modern object detectors follow a three step recipe: (1) proposing a search space of windows (exhaustive by sliding window or sparser using proposals), (2) scoring/refining the window with a classifier/regressor, and (3) merging windows that might belong to the same object. This last stage is commonly referred to as "non-maximum suppression" (NMS) (Girshick *et al.*, 2014; Girshick, 2015; Ren *et al.*, 2015; Felzenszwalb *et al.*, 2010; Redmon *et al.*, 2016; Liu *et al.*, 2016).

The de facto standard for NMS is a simple hand-crafted test time post-processing, which we call GreedyNMS. The algorithm greedily selects high scoring detections and deletes close-by, less confident neighbours as they are likely to cover the same object. This algorithm is simple, fast, and surprisingly competitive compared to proposed alternatives.

The most notable recent performance breakthrough in general object detection was marked by R-CNN (Girshick *et al.*, 2014), which effectively replaced features extraction and classifiers by a neural network, almost doubling performance on Pascal VOC. Another significant improvement was to absorb the object proposal generation into the network (Ren *et al.*, 2015), while other works avoid proposals altogether (Liu *et al.*, 2016; Redmon *et al.*, 2016), leading to both speed and quality improvements. We can see a general trend towards end-to-end learning and it seems reasonable to expect further improvements by doing complete end-to-end training of detectors. NMS is one step in the pipeline that, for the most part, has evaded the end-to-end learning paradigm. All of the above detectors train the classifier in a procedure that ignores the fact that the NMS problem exists and then runs GreedyNMS as a disconnected post-processing.

There is however a need to overcome GreedyNMS due to its significant conceptual shortcomings. GreedyNMS makes hard decision by deleting detections and bases this decision on one fixed parameter that controls how wide the suppression is. A wide suppression would remove close-by high scoring detections that are likely to be false positives that hurt precision. On the other hand, if objects are close (e.g. in crowded scenes), close-by detections can be true positives, in which case suppression should be narrow to improve recall. When objects are close-by, GreedyNMS is doomed to sacrifice precision or recall independent of its parameter.

It is desirable to learn NMS to overcome its limitations. An NMS approach based on neural network, could learn to adapt to the data distribution, overcome the trade-off of GreedyNMS, and importantly could be incorporated *into* a detector. In this chapter we propose the first "pure NMS network" which is able to do the task of non-maximum suppression without image content or access to decisions of another algorithm. This network renders the need for final GreedyNMS post-processing superfluous.

In section 8.2 we start by discussing with the underlying issue: why is needed NMS at all? We discuss the task of detection and how it relates to the specifics of detectors and NMS. We identify two necessary ingredients that current detectors are lacking and design an NMS network that contains these ingredients (section 8.3). The result is conceptually different than both NMS and current detectors. In section 8.4, we report promising results that show that this network is indeed capable of replacing GreedyNMS. We believe that this work opens the door to true end-to-end detectors.

8.2 Detection and non-maximum Suppression

In this section we review non-maximum suppression (NMS) and why it is necessary. In particular, we point out why current detectors are conceptually incapable of producing exactly one detection per object and propose two necessary ingredients for a detector to do so.

In section 8.1, we noted that virtually all detectors do not return all detections that have been scored, but instead use NMS as a post-processing step to remove redundant detections. In order to have true end-to-end learned detectors, we are interested in detectors without any post-processing. To understand why NMS is necessary, it is useful to look at the task of detection and how it is evaluated.

Object detection. The task of object detection is to map an image to a set of boxes: one box per object of interest in the image, each box tightly enclosing an object. This means detectors ought to return exactly one detection per object. Since uncertainty is an inherent part of the detection process, evaluations allow detections to be associated to a confidence. Confident erroneous detections are penalized more than less confident ones. In particular mistakes that are less confident than the least confident correct detection are not penalized at all.

Detectors do not output what we want. So the detection problem can be interpreted as a classification problem, that estimates probabilities of object classes being present for every possible detection in an image. This viewpoint gives rise to "hypothesize and score" detectors that build a search space of detections (e.g. sliding window, proposals) and estimate class probabilities *independently* for each detection. As a result, two very similar windows that cover the same object both predict a very similar score, since they look at almost identical image content, and both predict a high score since they both cover the object. In general one object triggers several detections of varying confidence, depending on how well the detection window covers the object.

GreedyNMS. Since the actual goal is to generate *exactly one* detection per object (or exactly one high confidence detection), a common practice (since at least 1994 (Burel and Carel, 1994)) is to assume that highly overlapping detections belong to the same object and collapse them into one detection. The predominant algorithm (GreedyNMS) accepts the highest scoring detection, then rejects all detections that overlap more than some threshold ϑ and repeats the procedure with the remaining detections, i.e. greedily accepting local maxima and discarding their neighbours, hence the name. This algorithm eventually also accepts wrong detections, which is no problem if their confidence is lower than the confidence of correct detections.

GreedyNMS is not good enough. This algorithm works well if (1) the suppression is wide enough to always suppress high scoring detections triggered by same object and (2) the suppression is narrow enough to never suppress high scoring detections of the next closest object. If objects are far apart condition (2) is easy to satisfy and a wide suppression works well. Only in crowded scenes with high occlusion between objects exists a tension between wide and narrow suppression. That means with one object per image NMS becomes trivial, while highly occluded objects require a better NMS algorithm.

8.2.1 A future without NMS

Striving for true end-to-end systems without hand crafted algorithms we shall ask: **Why do we need a hand crafted post processing step? Why does the detector not directly output one detection per object?**

Independent processing or estimating class probabilities given only image content leads to overlapping detection giving similar scores, which is also the requirement of robust functions: similar input lead to similar outputs. A detector that outputs only one high scoring detection per object thus has to be also conditioned on other detections: multiple detections on the same object should be processed jointly, so the detector tell there are multiple detections and only one of them should receive a high score.

Typical inference of detectors consist of a classifier that discriminates between image content that contains an object and image content that does not. The positive and negative training examples for this detector are usually defined by some measure of overlap between objects and bounding boxes. Since similar boxes will produce similar confidences anyway, small perturbation of object locations can be considered positive examples, too. This technique augments the training data and leads to detectors with more robustness. Using this type of classifier training does not reward one high scoring detection per object, but instead deliberately encourages multiple high scoring detection per object.

From this analysis we can see that two key ingredients are necessary in order for a detector to generate exactly one detection per object:

1. A *loss* that penalises double detections to teach the detector we want *precisely one* detection per object.

2. *Joint processing* of neighbouring detections so the detector has the necessary information to tell whether an object was detected multiple times.

In this chapter, we explore a network design that accommodates both. To validate the claim that these are key ingredients and our the proposed network is capable of performing NMS, we study this network in isolation without end-to-end learning with the detector. That means network operates solely on detections without image features and as such can be considered a “pure NMS network”.

8.3 Doing NMS with a convnet

After establishing the two necessary requirements for a convnet to perform NMS in section 8.2, this section presents our network that addresses both (penalizing double detections in §8.3.1, joint processing of detections in §8.3.2).

Our Gnet design avoids hard decisions and does not discard detections to produce a smaller set of detections. Instead, we reformulate NMS as a rescoring task that seeks to decrease the score of detections that cover objects that already have been detected, as in chapter 7. After rescoring, simple thresholding is sufficient to reduce the set of detections. In fact for evaluation we pass the full set of detections to the evaluation script without any post processing.

8.3.1 Loss

A detector is supposed to output exactly one high scoring detection per object. The loss for such a detector must inhibit multiple detections of the same object, irrespective of how close these detections are. Stewart and Andriluka (2016) use a Hungarian matching loss to accomplish that: successfully matched detections are positives and unmatched detections are negatives. That matching ensures that each object can only be detected once and any further detection counts as a mistake. Henderson and Ferrari (2016) present an average precision (AP) loss that is also based on matching.

Ultimately a detector is judged by the evaluation criterion of a benchmark, which in turn defines a matching strategy to decide which detections are correct or wrong. This is the matching that should be used at training time. Typically benchmarks sort detections in descending order by their confidence and match detections in this order

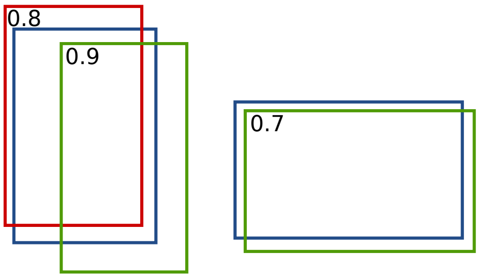


Figure 8.1: Blue boxes are object annotations, green are correct detections, red are wrong detections, numbers indicate detection confidence. The left annotation is matched to the green detection because it has a higher score, even though it is localized worse than the red detection.

to objects, preferring most overlapping objects (see figure 8.1). Since already matched objects cannot be matched again surplus detections are counted as false positives that decrease the precision of the detector.

We use the result of the matching as labels for the classifier: successfully matched detections are positive training examples, while unmatched detections are negative training examples for a standard binary loss. Typically all detections that are used for training of a classifier have a label associated as they are fed into the network. In this case the network has access to detections and object annotations and the matching layer generates labels, that depend on the predictions of the network. Note how this class assignment directly encourages the rescoring behaviour that we wish to achieve.

Let d_i denote a detection, $y_i \in \{-1, 1\}$ indicate whether or not d_i was successfully matched to an object, and let f denote the scoring function that jointly scores all detections on an image $f\left([d_i]_{i=1}^n\right) = [s_i]_{i=1}^n$. We train with the weighted logistic loss

$$L(s_i, y_i) = \sum_{i=1}^{N} w_{y_i} \log\left(1 + \exp\left(-s_i y_i\right)\right),$$

which at first glance may appear to be separable, when in fact the losses of two detections are actually coupled through the matching that produces y_i.

Weighting counteracts the extreme class imbalance of detection. We choose the weights so the expected class conditional weight of an example equals a parameter $\mathbf{E}\left(w_1 I\left(y_i = 1\right)\right) = \gamma$.

8.3.2 "Chatty" windows

In order to effectively minimize the aforementioned loss, we need our network to jointly process detections. To this end we design a network with a repeating structure, which we call blocks (sketched in figure 8.2). One block gives each detection access to the representation of its neighbours and subsequently updates its own representation. Stacking multiple blocks means the network alternates between allowing every detection "talk" to its neighbours and updating its own representation. We call this the **GossipNet** (Gnet), because detections talk to their neighbours to update their representation.

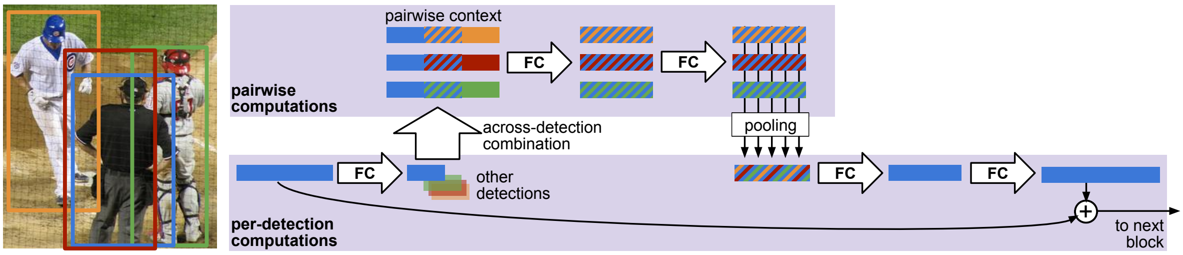


Figure 8.2: One block of our Gnet visualised for *one* detection. The representation of each detection is reduced and then combined into neighbouring detection pairs and concatenated with detection pair features (hatched boxes, corresponding features and detections have the same colour). Features of detection pairs are mapped independently through fully connected layers. The variable number of pairs is reduced to a fixed-size representation by max-pooling. Pairwise computations are done for each detection independently.

There are two non-standard operations here that are key. The first is a layer, that builds representations for pairs of detections. This leads to the key problem: an irregular number of neighbours for each detection. Since we want to avoid the discretisation scheme used in chapter 7, we will solve this issue with pooling.

Pairwise detection context. Each mini-batch consists of all n detections on an image, each represented by a c dimensional feature vector, so the data has the dimension $n \times c$ and access to another detection's representations means operating across batch elements. We use a detection context layer, that, for every detection d_i, generates all pairs of detections (d_i, d_j) for which d_j sufficiently overlaps with d_i (IoU > 0.2). The representation of a pair of detections consists of the concatenation of both detection representations and g dimensional detection pair features (see below), which yields an $l = 2c+g$ dimensional feature. Each detection pair's representation is arranged along the channel dimension and the different pairs are arranged along a spatial axis, $n \times l \times k_i \times 1$ (where k_i is the number of neighbours of detection d_i), so that a fully connected layer can process each detection pair independently (this is implemented as a convolution). Note that the number of neighbours k_i and so the number of pairs is different for every detection even within one mini-batch; we fill the unused space with zeros. To reduce the variable sized neighbourhood into a fixed size representation we use global pooling ($n \times l \times k_i \times 1 \rightarrow n \times l$), after which we can use normal fully connected layers to update the detection representation.

Detection pair features. The features for each detection pair used in the detection context consists of the detection scores of both detections and several properties of a detection pair: (1) the intersection over union (IoU), (2-4) the normalised distance in x and y direction and the normalised l2 distance (normalized by the average of width and height of the detection), (4-5) scale difference of width and height (e.g. $\log(w_i/w_j)$), and (6) aspect ratio difference $\log(a_i/a_j)$.

Block. A block does one iteration of allowing detections to look at their respective neighbours and updating the detection representation (sketched in figure 8.2). It consists of a dimensionality reduction, a pairwise detection context layer, 2 fully connected layers applied to each pair independently, global pooling, and two fully connected layers, where the last one increases dimensionality again. The input and output of a block are added as in the Resnet architecture (He *et al.*, 2016b). The first block receives zero features as inputs, so all information that is used to make the decision is bootstrapped from the detection pair features. The output of the last block is used by three fully connected layers to predict a new score for each detection independently.

Parameters. Unless specified otherwise our networks have 16 blocks. The feature dimension for the detection features is 128 and is reduced to 32 before building the pairwise detection context. The fully connected layers after the last block output 128 dimensional features. When we change the feature dimension, we always keep the ratio between the number of features constant, so indicating the detection feature dimension is sufficient.

We initialise the network with MSRA (He *et al.*, 2015). Due to the large number of layers, activations tend to explode and it proves useful to scale weights to approximately output unit variance (Mishkin and Matas, 2016).

Message passing. The process of several stacked blocks can be interpreted as message passing. Every detection sends messages to all of its neighbours in order to negotiate which detection is assigned an object and which detections should decrease their scores. Instead of hand-crafting the message passing algorithm and its rules, we deliberately let the network latently learn the messages that are being passed.

8.3.3 Remarks

The Gnet is fundamentally different than GreedyNMS in the sense that all features are updated concurrently, while GreedyNMS operates sequentially. Since Gnet does not have access to GreedyNMS decisions, it is surprising how close performance of the two algorithms turns out to be in section 8.4. Since we build a potentially big network by stacking many blocks, the Gnet might require large amounts of training data. In the experiments we deliberately choose a setting with many training examples.

The Gnet is a pure NMS network in the sense that is it has no access to image features and operates solely on detections (i.e. two coordinates and a confidence). This means the Gnet cannot be interpreted as extra layers to the detector. The fact that it is a neural network and that it is possible to feed in a feature vector (from the image or the detector) into the first block makes it particularly suitable for combining it with a detector which we leave for future work.

The goal is to *jointly* rescore all detections on an image. By allowing detections to look at their neighbours and update their own representation, we bootstrap conditional dependence between detections. Together with the loss that encourages exactly one detection per object, we have satisfied both conditions from section 8.2. We will see in section 8.4 that the performance is relatively robust to parameter changes and works increasingly well for increasing depth.

8.4 Experiments

In this section we experimentally evaluate the proposed architecture on the PETS and COCO dataset for people detection. In this work we focus on the people class, as it is by far the largest class on COCO and we want to avoid starving for training data. Other than overall results, we also report separately high and low occlusion cases. We are interested in performance on highly occluded people, since this is the case in which non-maximum suppression (NMS) is hard. We show good performance for high occlusions and improved detection localisation.

All results are measured in average precision (AP), which is the area under the recall-precision curve. The overlap criterion (for matching detections to objects) is traditionally 0.5 IoU (as for Pascal VOC, noted as $\text{AP}_{0.5}$), but COCO also uses stricter criteria to encourage better localisation quality. In particular one metric averages AP evaluated over several overlap criteria in the range $[0.5, 0.95]$ in 0.05 increments, which we denote by $\text{AP}_{0.5}^{0.95}$.

8.4.1 PETS: Pedestrian detection in crowds

Dataset. PETS (Ferryman and Ellis, 2010) is a dataset consisting of several very crowded sequences. We used it in chapter B as a roughly single scale pedestrian detection dataset with diverse levels of occlusion. Even though we aim for a larger and more challenging dataset we first analyse our method in the setup of chapter B. We use the same training and test set as well as the same detections from (Tang *et al.*, 2013) that has been trained for high occlusion. We reduce the number of detections with an initial GreedyNMS of 0.8 so we can fit the joint rescoring of all detections into one GPU. (Note that these detections alone lead to bad results, worse than “GreedyNMS > 0.6” in 8.3, and this is very different to having input of GreedyNMS of 0.5 as an input like in chapter B).

Training. We train a model with 8 blocks and a 128 dimensional detection representation for 30k iterations, starting with a learning rate of 10^{-3} and decrease it by 0.1 every 10k iterations.

Baselines. We compare to the classic GreedyNMS algorithm, which is typically used, with several different overlap thresholds and the Strong Tnet from chapter B. Since all methods operate on the same detections, the results are fully comparable.

Analysis. Figure 8.3 compares our method with the GreedyNMS baseline and the Tnet on the PETS test set. Starting from a wide GreedyNMS suppression with the threshold $\vartheta = 0$ shows almost a step function, since high scoring true positives suppress

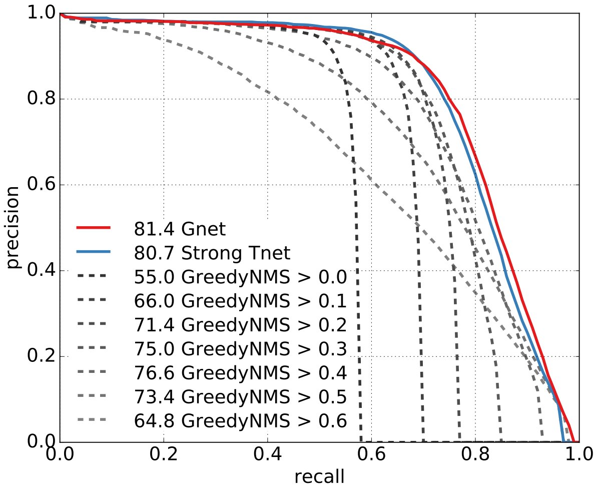


Figure 8.3: Performance on the PETS test set.

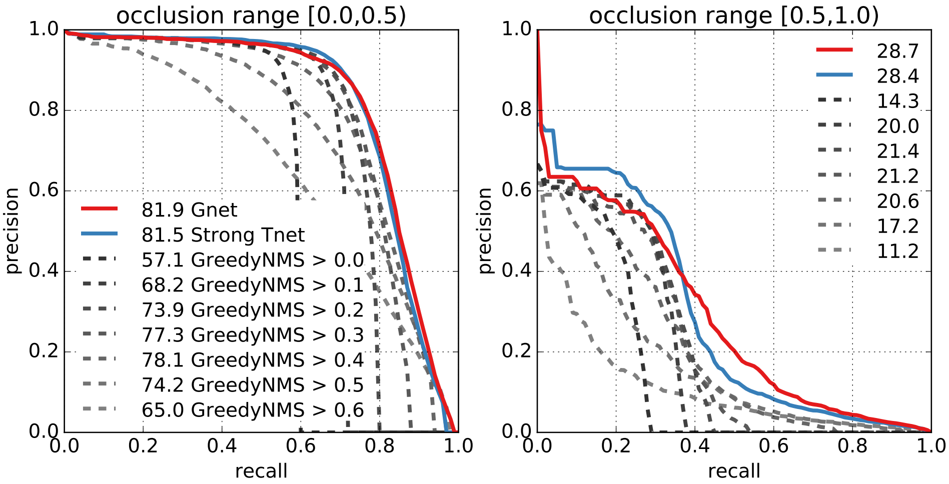


Figure 8.4: Performance on the PETS test set for different occlusion ranges.

all touching detections at the cost of also suppressing other true positives. Gradually increasing ϑ improves the maximum recall but also introduces more high scoring false positives, so precision is decreasing. This shows nicely the unavoidable trade-off due to having a fixed threshold ϑ mentioned in section 8.2. The reason for the clear trade-off is the diverse occlusion statistics as argued in section 8.2.

Tnet performs better than the upper envelope of the GreedyNMS, as it essentially recombines output of GreedyNMS at a range of different thresholds. In comparison our Gnet performs slightly better, despite not having access to GreedyNMS decisions at all. Compared to the best GreedyNMS performance, Gnet is able to improve by 4.8 AP.

Figure 8.4 shows performance separated into high and low occlusion cases. Again, the Gnet performs only slightly better than Tnet. Performance in the occlusion range $[0, 0.5)$ looks very similar to the performance overall, but GreedyNMS performs roughly 2 AP better. For the highly occluded cases, the performance improvement of Gnet compared

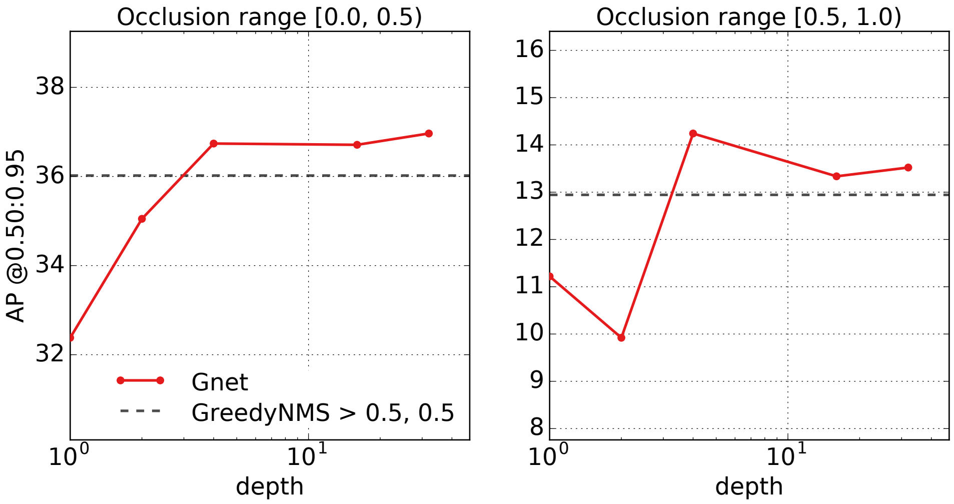


Figure 8.5: $AP_{0.5}^{0.95}$ versus number of blocks for low and high occlusion respectively on a subset of the validation set.

	All		Occlusion [0, 0.5)		Occlusion [0.5, 1]	
Method	$AP_{0.5}$	$AP_{0.5}^{0.95}$	$AP_{0.5}$	$AP_{0.5}^{0.95}$	$AP_{0.5}$	$AP_{0.5}^{0.95}$
GreedyNMS > 0.5	66.0	36.2	65.7	36.0	35.1	12.9
Gnet, 32 blocks	**66.8**	**37.2**	**66.4**	**37.0**	**36.7**	**13.5**

Table 8.1: Comparison between Gnet and GreedyNMS on minival. Results for the full set and results separated into occlusion levels.

to the best GreedyNMS is bigger with 7.3 AP. This shows that the improvement for both Gnet and Tnet is mainly due to improvements on highly occluded cases as argued in section 8.2.

8.4.2 COCO: Person detection

Dataset. The COCO datasets consists of 80k training and 40k evaluation images. It contains 80 different categories in unconstrained environments.

We train our network on the full training set and evaluate two different subsets of the validation set, each consisting of 5k images. One subset is used to explore architectural choices for our network in section 8.4.2.1 (minival) and the most promising model is evaluated on another subset in section 8.4.2.2 (minitest).

	All		Occlusion [0, 0.5)		Occlusion [0.5, 1]	
Method	$AP_{0.5}$	$AP_{0.5}^{0.95}$	$AP_{0.5}$	$AP_{0.5}^{0.95}$	$AP_{0.5}$	$AP_{0.5}^{0.95}$
GreedyNMS > 0.5	66.4	37.2	66.2	37.1	31.2	11.8
Gnet, 32 blocks	**67.4**	**38.4**	**67.2**	**38.2**	**34.9**	**13.6**

Table 8.2: COCO results on minitest, full results and separated into occlusion levels.

We use the Python implementation of Faster RCNN (Ren *et al.*, 2015)[15] for generating detections. We run the detector with default parameters, but use detection before the typical non-maximum suppression step.

Training. We train the Gnet for 300k iterations, starting with a learning rate of 10^{-4} and decreasing it to 10^{-5} after 160k iterations. The detection feature dimension is 128, the number of blocks is specified for each experiment.

Baselines. We use GreedyNMS as a baseline. Since the Gnet is trained, we use the optimal overlap threshold for GreedyNMS on the test set for each experiment.

8.4.2.1 *Network analysis*

Figure 8.5 shows performance on the y axis in the standard evaluation $\mathrm{AP}_{0.5}^{0.95}$, that rewards better localisation. Interestingly, the best GreedyNMS threshold is 0.5 for both low and high occlusion. Even though a narrower suppression does increase recall, the number of introduced false positives decreases the precision too much to obtain a better AP.

Figure 8.5 shows the Gnet performance as a function of its number of blocks. As the number of blocks increases, the performance on low occlusion cases increases steadily. With four blocks Gnet outperforms GreedyNMS, 32 blocks again slightly improve better to 37.0 AP compared to GreedyNMS with 36.0 AP. The Trend on highly occluded cases is similar, although performance starts to decrease after a depth of 4 blocks. This could be an indication that there are too few highly occluded cases in the training set to properly train the model of increasing capacity.

Out of those Gnet models with varying depth, we select the deepest model with 32 blocks to show further evaluations. Table 8.1 shows overall and per-occlusion performance for both $\mathrm{AP}_{0.5}$ and $\mathrm{AP}_{0.5}^{0.95}$. For the traditional $\mathrm{AP}_{0.5}$ metric, Gnet improves over GreedyNMS by 0.8 to 66.8 AP. For the stricter evaluation, the improvement of 1.0 AP is slightly bigger, demonstrating larger improvement for stricter overlap criteria, i.e. Gnet leads to better localized detections. The picture for the low occlusion range $[0, 0.5)$ is similar, since 99% of all detections fall into this occlusion range. For the high occlusion case Gnet improves by 1.6 AP with the overlap criterion 0.5 and a 0.6 improvement for $\mathrm{AP}_{0.5}^{0.95}$, so localisation accuracy does seem to be significantly improved. This shows how hard localisation and NMS is in highly occluded cases.

8.4.2.2 *Holdout set: test results*

Since annotations on the test set are not available and we want to explicitly show statistics per occlusion, we use a holdout subset of the validation set (minitest) that has not been used in experiments above. Table 8.2 shows performance of the same model as in table 8.1 (32 blocks). The trends that we observed for minival repeat: small consistent improvements of around 1 AP overall and low occlusion for both $\mathrm{AP}_{0.5}$ and

[15] `https://github.com/rbgirshick/py-faster-rcnn`

$\text{AP}_{0.5}^{0.95}$. Improvements for high occlusions are greater, but again the improvement for the overlap criterion 0.5 is greater than for $\text{AP}_{0.5}^{0.95}$ (3.7 and 1.8 AP respectively).

Both the small consistent improvements overall and more significant improvements for high occlusion ranges make for encouraging results. The Gnet is capable of performing NMS without access to image features or GreedyNMS decisions.

8.5 Conclusion

In this work we have opened the door for training detectors that no longer need a non-maximum suppression (NMS) post-processing step. We have shown that NMS is usually needed as post-processing because the detectors commonly process neighbouring detections independently. As a result we have identified two key ingredients missing in detectors that are necessary to be able to discard NMS: (1) a loss that penalises double detections and (2) joint processing of detections.

We have introduced the Gnet, the first "pure NMS network" that is fully capable of performing the NMS task without having access to image content or help from another algorithm. Being a neural network, it lends itself to being incorporated into detectors and having access to image features in order to build detectors that can be trained truly end-to-end. These end-to-end detectors will not require any post-processing.

The experimental results indicate that, with enough training data, the proposed Gnet is a suitable replacement for GreedyNMS. The network surpasses GreedyNMS in particular for occluded cases and provides improved localization.

In its current form the Gnet requires large amounts of training data and it would benefit from future work on data augmentation or better initialisation by pre-training on synthetic data. Incorporating image features could can have a big impact, as they have the potential of informing the network about about the number of objects in the image. A multi-class generalisation will also be an important step towards end-to-end detectors.

We believe the ideas and results discussed in this work point to a future where the distinction between detector and NMS will disappear.

Conclusions, insights, and future perspectives 9

OBJECT detection has seen significant improvements during recent years, in particular by incorporating deep learning and constantly reconsidering how to better utilise deep learning in object detectors. In this thesis we have worked on analysing and understanding current detection pipelines and challenging common assumptions that are made when designing detectors and training them. In this chapter, we conclude the thesis by recapitulating our key findings and pointing out future research directions that arise from this work.

9.1 Conclusions and insights

9.1.1 Pedestrian detection

In chapters 3–5, we reviewed a decade of research, explored deep learning with large and small vanilla networks, and analysed failure modes of recent pedestrian detectors. This work has led to the following insights:

Families of approaches. Historically, the most prominent families of approaches were performing similarly well. Since 2015, neural networks have been driving the field forward. While boosted decision forests were keeping up for a while, they do not seem to keep up with the most recent improvements.

Temporal and spacial context. We found the consistent, small improvements from temporal context and pedestrian co-occurrence to be mostly orthogonal.

Deep learning. Deep learning is an effective means of learning features directly from pixels using neural networks, without explicitly modelling problem specific aspects like occlusion or parts. Using additional data helps, both outside data for pretraining the network and sampling more frames from the training video stream. Although more densely sampled frames become more similar, they still show significant benefit. Pretraining networks on ImageNet classification also shows performance improvements, but even without pretraining, large networks perform significantly better than for previous neural network detectors.

Features are driving progress. It is possible to emulate the progress from 2004 (Viola *et al.*, 2003) to 2014 (`SquaresChnFtrs` in chapter 3) by changing the features in the `SquaresChnFtrs` detector. Our deep learning experiments confirm this insight and since then the field has mostly worked on learning better features for small pedestrians.

Well localised annotations are important. Poorly localised annotations on the Caltech pedestrian detection benchmark are problematic. Detectors trained on the INRIA benchmark show overall worse performance, but better localisation since INRIA annotations are better localised. Caltech trained models suffer from localisation issues and the testset annotations are incapable of measuring better localisation. Detector performance on Caltech has become so good that performance metrics saturate because of poorly localised annotations. Our new annotations and their model aided transfer to intermediate frames in the training video will help future progress and accurately measure that progress.

Failure modes. Main failure modes of top detectors are double detections and vertical structures for false positives and small pedestrians and side views of pedestrians for missing recall. These insights suggest specific future work.

9.1.2 Detection proposals

In chapter 6, we examined detection proposals, which are often used to define the search space of object detectors.

Limited repeatability. Virtually all examined proposal methods show limited repeatability; even changing a single pixel has a noticeable effect on the set of generated detection proposals. `SelectiveSearch` and `EdgeBoxes` strike a better balance between recall and repeatability than other methods. Overall it is not clear that repeatability is a crucial property of detection proposals, but we expect better proposal methods to be more repeatable.

Localisation matters. Detector responses scale linearly with higher overlap between proposal and object. As a result, improving recall is as important as improving localisation or put another way, high recall at low overlap is not effective. We proposed a robust measure, *average recall*, that correlates well with detector performance given a fixed number of proposals. This is a better proxy measure to optimise for detector performance than recall at one specific overlap threshold.

Failure analysis of Fast R-CNN with `EdgeBoxesAR`. Fast R-CNN with `EdgeBoxesAR` proposals reaches 67.8% mAP on Pascal VOC. Of the missing 32.2% mAP of performance, 10% can be obtained by better localised proposals. 21% can be rectified by perfect non-maximum suppression (i.e. true positives suppress only false positives). The remaining 68% are caused by high scoring false positives on background, which are only avoidable by either removing them from the set of proposals or with a better classifiers, which does not make these classification mistakes.

Broad comparison. We compared proposal methods with respect to a wide range of aspects, which allows practitioners to pick a method that suits their needs and gives guidance on how to evaluate new work on proposal methods. Prime candidates are `SelectiveSearch`, `Rigor`, `MCG`, and `EdgeBoxes`, all of which approach the problem very differently.

Class generalisation. We found that the proposal methods we examined generalise beyond the 20 PASCAL classes, they were designed for. They performed similarly well for 200 ImageNet classes and slightly worse for 80 classes on COCO, which contains smaller objects and more objects per image. This enables other uses other than traditional object detection, e.g. handle unknown categories or provide weak supervision.

9.1.3 Non-maximum suppression

NMS is necessary because detectors process neighbouring detections independently and are trained to trigger also close to objects. In chapter 7 and 8, we have explored NMS as a rescoring problem, means to overcome limitations of the classic GreedyNMS and to learn NMS. This paradigm leads the way to detectors that no longer need NMS or any other form of post-processing.

GreedyNMS limintations. GreedyNMS uses one global overlap threshold to decide whether two detections cover the same object or not. A too wide suppression rejects true positives in dense crowds, while too narrow suppression accepts high scoring false positives close to objects. No setting of the overlap parameter mitigates both mistakes for practical scenarios with crowds.

Convnets can overcome limitations of GreedyNMS. It is possible to learn a convnet that combines decisions of GreedyNMS at different overlap thresholds, while only using the information that is also available to GreedyNMS (detection scores and overlap between detections). This convnet is better than GreedyNMS at any overlap threshold and even better than the upper envelope of all GreedyNMS average-precision curves.

Convnets can perform NMS. Using only detection scores and geometric information we presented a neural network architecture that can learn to perform NMS without using GreedyNMS or image content. The architecture allows joint processing of neighbouring detections by allowing message passing between detections.

Necessary ingredients for detector to learn NMS. To learn NMS or, more generally, learn a detector that produces exactly one high scoring detection per object, we identified two necessary ingredients: (1) a loss that penalises double detections and (2) joint processing of neighbouring detections. We studied this new paradigm of detector training in NMS, i.e. a function that maps a set of detections to a different set of detections without access to the image.

9.2 Limitations and short-term improvements

9.2.1 Pedestrian detection

We identified concrete failure modes of current pedestrian detectors, which leads to suggestions for future work.

Better features for small pedestrians. Small pedestrians are problematic because they consist of few pixels and earlier convnet detectors use feature maps with a 16 pixel stride. Zhang *et al.* (2016a) and Cai *et al.* (2016) both introduce explicit scale handling to better detect small pedestrians.

Vertical extension of receptive field. Confusions between pedestrians and vertical structures like light poles could be addressed by extending the receptive field of the detector vertically. This gives the detector the possibility take image content above the potential pedestrian into account.

More training data. Since the detectors miss many side view pedestrians, it would be helpful including more of those into the training set. In general more pedestrian variety as well as more background pedestrians will be beneficial.

Double detections: we need better NMS. Double detections are caused by a structural problem in the detection pipeline. GreedyNMS forces a trade-off between precision and recall and needs to be replaced by something more adaptable. In chapter 7 and 8 we address NMS to become learnable and adaptive.

9.2.2 Non-maximum suppression

Our work on retiring non-maximum suppression is a first step in the process. The following points sketch a rough roadmap towards the goal learning non-maximum suppression implicitly in detectors in an end-to-end fashion.

Replacement for global max pooling. We suspect that the global max pooling in the Gnet might be discarding a lot of information. Vinyals *et al.* (2016) describe a way to process set inputs, which might lead to more discriminative representation of pairs of detections.

Multiclass extension of matching loss. The current version of the matching loss was only used in a single class setting. It can be extended to the multi-class setting by matching detections to annotations only if their classes match. Typically, minibatch training of convnets weights images or detections equally, while mAP weights classes equally. We can use weighted logistic regression to optimize for class-weighted evaluation.

Preliminary experiments with this multiclass extension show small improvements over GreedyNMS on COCO.

Add image features to Gnet. Image features can help disambiguating hard cases with a high level of occlusion. The Gnet was explicitly designed to take detection features as an input and update them to use them for non-maximum suppression. It is straight forward to use detection features, for example as generated in Faster R-CNN to obtain a stronger NMS network.

Add Gnet into detector. We identified two necessary ingredients for a network to perform NMS and experimentally verified the Gnet is capable of performing NMS. We can incorporate these two ingredients into a detector to train a detector that will not require any post-processing at test time. This involves switching to the matching loss and adding joint processing of neighbouring detections as we have done in the Gnet.

Different losses. In chapter 8 we argue that, in order to retire NMS, we need a loss that penalises double detections. Defining correct an incorrect detections via matching appears to be a promising direction since the matching is also used in the evaluation. We use the logistic loss, because perfect separation between correct and incorrect detections leads to a perfect AP curve. It is, however, unclear that the trade-offs of a classification loss (e.g. number of misclassifications vs. severity of misclassification) are beneficial. A more promising approach would be to directly, or at least approximately, optimize the area under the AP curve. Henderson and Ferrari (2016) and Eban *et al.* (2016) explore ideas in that direction.

9.3 Future perspectives

9.3.1 Detection proposals and scalable object detection

As we argued in section 2.3, detection proposals are only useful in the context of a specific application and optimising the right metrics is a key aspect of this (see chapter 6). For some applications the proposal methods seem to be absorbed into the approaches and the distinction between proposal and detector or between proposal and video instance segmenter disappears.

In section 2.3, we pointed out some work on weakly supervised object detection and instance segmentation that utilises proposals. These systems have the ability to learn fine grained output from cheaper and coarser annotations, e.g. instance segmentations from bounding box annotations. They enable scaling the set of recognisable object classes beyond the currently typical 80–200 classes at small costs.

The ability of some proposal methods to generalise beyond the set of training classes is key to these approaches. We expect weakly supervised learning to be the main application for proposals, since problems with stronger supervision allow the training of more problem specific models.

Scalable and cost effective detection should become more autonomous and utilise all available supervision currently available. While a lot of work is focusing on extreme settings with very little supervision overall, we should not discard the large amount of annotations on ImageNet or COCO. The available supervision carries crucial information about objects in general and should help to discover new object classes. New annotations by humans should be concentrated on maximising the performance improvement of specific models. Instead of collecting and annotating 2 000 instances of a new object class, it is much more helpful to annotate instances where a model is uncertain or annotate a cluster of newly discovered objects. This leads to a combination of class generalisation, weakly supervised learning, and active learning.

9.3.2 Small scale objects

Many object detection benchmarks are biased towards one prominent object occupying the center foreground of images. In these cases with high resolution objects, object detectors show strong performance and downsampling in convnets does not pose problems. However, small scale objects remain a prominent source of concern in real-world scenarios. The problem becomes apparent in less biased benchmarks with many small scale objects such as COCO (Lin *et al.*, 2014) or data collected with a driving car (Cordts *et al.*, 2016; Geiger *et al.*, 2012; Dollár *et al.*, 2009b).

The current approaches to addressing this problem use lower layers of the use convnet representation, as those layers have a higher resolution. However, features in lower layers have a smaller receptive fields and do not benefit from the high-level reasoning that is happening in the upper layers of deep networks. Inspired by the SharpMask approach (Pinheiro *et al.*, 2016), Lin *et al.* (2016c) construct a feature pyramid, in which the high resolution features are predicted using also the upper layers of the network.

This approach allows more reasoning on the global scale of the image, yet it is unclear if the level of supervision is sufficient to enable effective reasoning. Detectors are typically only supervised with object classes and locations and we hope that the concepts that are useful for scene level reasoning (other than objects) emerge due to the hierarchical architecture of convnets.

A multitask setting that provides supervision for multiple, related tasks may lead to richer representations that enable stronger scene level reasoning. In the case of diverse viewpoints and scenes (like on COCO) useful tasks may be scene classification, semantic labelling, and action recognition. In scenes with very strong geometric priors and constraints (like on Citiscapes, KITTI, and Caltech) detectors may profit from representations that better capture 3d geometry are able to predict ground planes or distances between camera and objects.

9.3.3 Generation of a sparse set of detections

Virtually all current detectors are trained to generate multiple detections per object and then reduce the set of detections post-hoc. In chapter 7 and 8 we approach the problem with rescoring, never changing the number of detections, but (in the best case) ranking double detections so low, that they do not have negative impact on the AP curve.

It would be computationally more efficient to not generate several detections of the same object in the first place. One approach to accomplish this could be a sequential process that alternates between generating detections and updating the image representation. Stewart and Andriluka (2016) propose an LSTM to do this kind of sequential processing, but they process patches of the image independently, which causes double detections for objects that lie on the border between to patches. The LSTM should operate on a image wide representation to avoid double detections between two neighbouring patches. That does not mean that the LSTM needs to use an image wide representation as its internal state. It can operate on one patch of the image representation at a time and, for example, use an attention mechanism to select regions to generate detections. However, after having generated a detection, the process needs to update the global image representation, so subsequent detection predictions can be conditioned on previous predictions.

9.3.4 Linking detections to image content

Bounding boxes are a useful simplification of object locations in an image. They are compact and fast to annotate, but they make the task of non-maximum suppression hard: Two highly overlapping bounding boxes can belong to the same object or to two distinct objects, one heavily occluding the other. This could be resolved, if the detector has a finer representation of the image area that objects occupy.

Considering that each pixel should belong to at most one object, it should be possible to avoid re-using image content when doing non-maximum suppression or when generating detections. Intuitively, if we had perfect instance segments, standard non-maximum suppression using overlaps between the segments would solve the problem. In the case of a double detection, the two segmentation masks would largely overlap, while in the case of high occlusion the overlap should be very small.

Explicit representation of instance segments for all detections is impracticable, because of their size. It is necessary for a detector to reason about segments implicitly and with a compact representation. This requires an image representation that is shared between neighbouring detections, in which detections can “claim” image evidence for themselves in order to affect other detections. It could be approached with a sequential process as discussed before or with “communicating detections” as in chapter 8.

Instance segmentation should become more important not just for the sake of more fine-grained output. Detectors should be able learn about the concept of linking detections to pixels in order to avoid over-counting. It may also be useful to extend this idea to reasoning about occlusion.

A Weakly Supervised Instance and Semantic Segmentation

SEMANTIC labelling and instance segmentation are two tasks that require particularly costly annotations. Starting from weak supervision in the form of bounding box annotations, we propose a new approach that does not require modification of the segmentation training procedure. We show that when carefully designing the input labels from given bounding boxes, even a single round of training is enough to improve over previously reported weakly supervised results. Overall, our weak supervision approach reaches $\sim$95% of the quality of the fully supervised model, both for semantic labelling and instance segmentation.

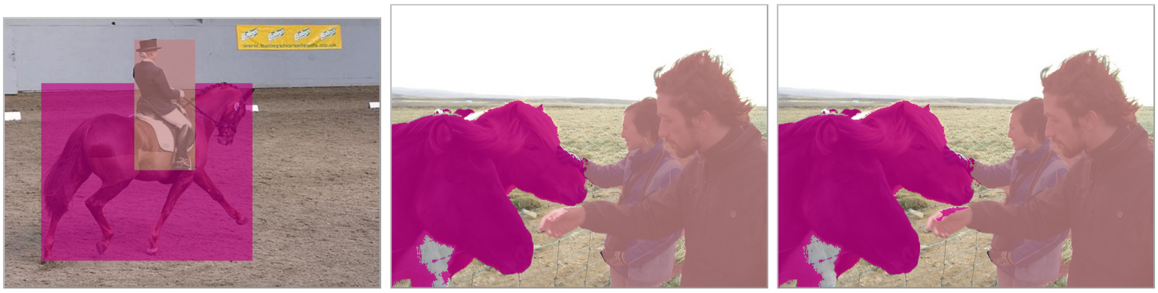
This work will be published at CVPR 2017. Anna Khoreva was the lead author on this paper, providing most experiments. Jan Hosang contributed all work on instance segmentation experiments in section A.6 except for the DeepLab experiments.

A.1 Introduction

Convolutional networks (convnets) have become the de facto technique for pattern recognition problems in computer vision. One of their main strengths is the ability to profit from extensive amounts of training data to reach top quality. However, one of their main weaknesses is that they need a large number of training samples for high quality results. This is usually mitigated by using pre-trained models (e.g. with $\sim 10^6$ training samples for ImageNet classification (Russakovsky *et al.*, 2015)), but still thousands of samples are needed to shift from the pre-training domain to the application domain. Applications such as semantic labelling (associating each image pixel to a given class) or instance segmentation (grouping all pixels belonging to the same object instance) are expensive to annotate, and thus significant cost is involved in creating large enough training sets.

Compared to object bounding box annotations, pixel-wise mask annotations are far more expensive, requiring $\sim$15× more time (Lin *et al.*, 2014). Cheaper and easier to define, box annotations are more pervasive than pixel-wise annotations. On principle, a large number of box annotations (and images representing the background class) should convey enough information to understand which part of the box content is foreground and which is background. In this appendix we explore how much one can close the gap between training a convnet using full supervision for semantic labelling or instance segmentation versus using only bounding box annotations.

Our experiments focus on the 20 Pascal classes (Everingham *et al.*, 2014) and show that using only bounding box annotations over the same training set we can reach $\sim$95%



Training sample, with box annotations

Test image, fully supervised result

Test image, weakly supervised result

Figure A.1: We propose a technique to train semantic labelling from bounding boxes, and reach 95% of the quality obtained when training from pixel-wise annotations.

of the accuracy achievable with full supervision. We show top results for (bounding box) weakly supervised semantic labelling and, to the best of our knowledge, for the first time report results for weakly supervised instance segmentation.

We view the problem of weak supervision as an issue of input label noise. We explore recursive training as a de-noising strategy, where convnet predictions of the previous training round are used as supervision for the next round. We also show that, when properly used, "classic computer vision" techniques for box-guided instance segmentation are a source of surprisingly effective supervision for convnet training.

In summary, our main contributions in this appendix are:

- We explore recursive training of convnets for weakly supervised semantic labelling, discuss how to reach good quality results, and what the limitations of the approach are (section A.3.1).
- We show that state of the art quality can be reached when properly employing GrabCut-like algorithms to generate training labels from given bounding boxes, instead of modifying the segmentation convnet training procedure (section A.3.2).
- We report the best known results when training using bounding boxes only, both using Pascal VOC12 and VOC12+COCO training data, reaching comparable quality with the fully supervised regime (section A.4.2).
- We are the first to show that similar results can be achieved for the weakly supervised instance segmentation task (section A.6).

A.2 Related work

Semantic labelling. Semantic labelling may be tackled via decision forests (Shotton *et al.*, 2009) or classifiers over hand-crafted superpixel features (Gould *et al.*, 2009). However, convnets have proven particularly effective for semantic labelling. A flurry of variants have been proposed recently (Pinheiro and Collobert, 2014; Long *et al.*, 2015;

Chen *et al.*, 2015; Lin *et al.*, 2016b; Zheng *et al.*, 2015; Kokkinos, 2016; Yu and Koltun, 2016). In this work we use DeepLab (Chen *et al.*, 2015) as our reference implementation. This network achieves state-of-the-art performance on the Pascal VOC12 semantic segmentation benchmark and the source code is available online.

Almost all these methods include a post-processing step to enforce a spatial continuity prior in the predicted segments, which provides a non-negligible improvement on the results ($2 \sim 5$ points). The most popular technique is DenseCRF (Krähenbühl and Koltun, 2011), but other variants are also considered (Kolmogorov and Zabih, 2004; Barron and Poole, 2015).

Weakly supervised semantic labelling. In order to keep annotation cost low, recent work has explored different forms of supervision for semantic labelling: image labels (Pathak *et al.*, 2015b,a; Papandreou *et al.*, 2015; Pinheiro and Collobert, 2015; Wei *et al.*, 2015), points (Bearman *et al.*, 2016), scribbles (Xu *et al.*, 2015; Lin *et al.*, 2016a), and bounding boxes (Dai *et al.*, 2015a; Papandreou *et al.*, 2015). Dai *et al.* (2015a), Papandreou *et al.* (2015), and Hong *et al.* (2015) also consider the case where a fraction of images are fully supervised. Xu *et al.* (2015) proposes a framework to handle all these types of annotations.

In this work we focus on box level annotations for semantic labelling of objects. The closest related work are thus Dai *et al.* (2015a) and Papandreou *et al.* (2015). BoxSup (Dai *et al.*, 2015a) proposes a recursive training procedure, where the convnet is trained under supervision of segment object proposals and the updated network in turn improves the segments used for training. WSSL (Papandreou *et al.*, 2015) proposes an expectation-maximisation algorithm with a bias to enable the network to estimate the foreground regions. We compare with these works in the result sections. Since all implementations use slightly different networks and training procedures, care should be taken during comparison. Both Dai *et al.* (2015a) and Papandreou *et al.* (2015) propose new ways to train convnets under weak supervision. In contrast, in this work we show that one can reach better results without modifying the training procedure (compared to the fully supervised case) by instead carefully generating input labels for training from the bounding box annotations (section A.3).

Instance segmentation. In contrast to instance agnostic semantic labelling that groups pixels by object class, instance segmentation groups pixels by object instance and ignores classes.

Object proposals (chapter 6, Pont-Tuset and Gool, 2015) that generate segments, such as Pont-Tuset *et al.* (2017) and Krähenbühl and Koltun (2015), can be used for instance segmentation. Similarly, given a bounding box (e.g. selected by a detector), GrabCut (Rother *et al.*, 2004) variants can be used to obtain an instance segmentation, e.g. Lempitsky *et al.* (2009), Cheng *et al.* (2015), Taniai *et al.* (2015), Tang *et al.* (2015a), and Yu *et al.* (2015).

To enable end-to-end training of detection and segmentation systems, it has recently been proposed to train convnets for the task of instance segmentation (Hariharan *et al.*, 2015; Pinheiro *et al.*, 2015). In this work we explore weakly supervised training of an instance segmentation convnet. We use DeepMask (Pinheiro *et al.*, 2015) as a reference implementation for this task. In addition we re-purpose DeepLabv2 network (Chen *et al.*, 2016), originally designed for semantic segmentation, for the instance segmentation task.

A.3 From boxes to semantic labels

The goal of this work is to provide high quality semantic labelling starting from object bounding box annotations. We design our approach aiming to exploit the available information at its best. There are two sources of information: the annotated boxes and priors about the objects. We integrate these in the following cues:

C1 Background. Since the bounding boxes are expected to be exhaustive, any pixel not covered by a box is labelled as background.

C2 Object extend. The box annotations bound the extent of each instance. Assuming a prior on the objects shapes (e.g. oval-shaped objects are more likely than thin bar or full rectangular objects), the box also gives information on the expected object area. We employ this size information during training.

C3 Objectness. Other than extent and area, there are additional object priors at hand. Two priors typically used are spatial continuity and having a contrasting boundary with the background. In general we can harness priors about object shape by using segment proposal techniques (Pont-Tuset and Gool, 2015), which are designed to enumerate and rank plausible object shapes in an area of the image.

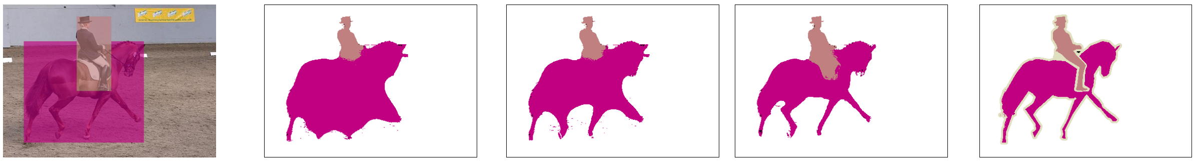
A.3.1 Box baselines

We first describe a naive baseline that serves as starting point for our exploration. Given an annotated bounding box and its class label, we label all pixels inside the box with the given class. If two boxes overlap, we assume the smaller one is in front. Any pixel not covered by boxes is labelled as background.

Figure A.2 left side and figure A.3c show such example annotations. We use these labels to train a segmentation network with the standard training procedure. We employ the DeepLabv1 approach from Chen *et al.* (2015) (details in section A.4.1).

Recursive training. We observe that when applying the resulting model over the training set, the network outputs capture the object shape significantly better than just boxes (see figure A.2). This inspires us to follow a recursive training procedure, where these new labels are fed in as ground truth for a second training round. We name this recursive training approach `Naive`.

The recursive training is enhanced by de-noising the convnet outputs using extra information from the annotated boxes and object priors. Between each round we improve the labels with three post-processing stages:



Example input rectangles

Output after 1 training round

After 5 rounds

After 10 rounds

Ground truth

Figure A.2: Example results of using only rectangle segments and recursive training (using convnet predictions as supervision for the next round), see section A.3.1.

1. Any pixel outside the box annotations is reset to background label (cue C1).
2. If the area of a segment is too small compared to its corresponding bounding box (e.g. IoU$<$ 50%), the box area is reset to its initial label (the same that is used in the first round). This enforces a minimal area (cue C2).
3. As it is common practice among semantic labelling methods, we filter the output of the network to better respect the image boundaries. (We use DenseCRF (Krähenbühl and Koltun, 2011) with the DeepLabv1 parameters (Chen *et al.*, 2015).) In our weakly supervised scenario, boundary-aware filtering is particularly useful to improve object delineation (cue C3).

The recursion and these three post-processing stages are crucial to reach good performance. We name this recursive training approach `Box`, and show an example result in figure A.2.

Ignore regions. We also consider a second variant `Box`$^{\mathtt{i}}$ that, instead of using filled rectangles as initial labels, we fill in the 20% inner region, and leave the remaining inner area of the bounding box as ignore regions. See figure A.3d. Following cues C2 and C3 (shape and spatial continuity priors), the 20% inner box region should have higher chances of overlapping with the corresponding object, reducing the noise in the generated input labels. The intuition is that the convnet training might benefit from trading-off lower recall (more ignore pixels) for higher precision (more pixels are correctly labelled). Starting from this initial input, we use the same recursive training procedure as for `Box`.

Despite the simplicity of the approach, as we will see in the experimental section A.4, `Box` / `Box`$^{\mathtt{i}}$ is already competitive with the current state of the art.

However, using rectangular shapes as training labels is clearly suboptimal. Therefore, in the next section, we propose an approach that obtains better results while avoiding multiple recursive training rounds.

A.3.2 Box-driven segments

The box baselines are purposely simple. A next step in complexity consists in utilising the box annotations to generate an initial guess of the object segments. We think of this as “old school meets new school”: we use the noisy outputs of classic computer vision methods, box-driven figure-ground segmentation (Rother *et al.*, 2004) and object proposal techniques (Pont-Tuset and Gool, 2015) to feed the training of a convnet. Although the output object segments are noisy, they are more precise than simple rectangles, and thus should provide improved results. A single training round will be enough to reach good quality.

A.3.2.1 GrabCut baselines

GrabCut (Rother *et al.*, 2004) is the established technique to estimate an object segment from its bounding box. We propose to use a modified version of GrabCut, which we call `GrabCut+`, where HED boundaries (Xie and Tu, 2015) are used as pairwise term instead of the typical RGB colour difference. (The HED boundary detector is trained on the generic boundaries of BSDS500 (Arbeláez *et al.*, 2011).) We considered other GrabCut variants, such as Cheng *et al.* (2015) and Tang *et al.* (2015a); however, the proposed `GrabCut+` gives higher quality segments.

Similar to $\texttt{Box}^{\texttt{i}}$, we also consider a $\texttt{GrabCut+}^{\texttt{i}}$ variant, which trades off recall for higher precision. For each annotated box we generate multiple (~ 150) perturbed `GrabCut+` outputs. If 70% of the segments mark the pixel as foreground, the pixel is set to the box object class. If less than 20% of the segments mark the pixels as foreground, the pixel is set as background, otherwise it is marked as ignore. The perturbed outputs are generated by jittering the box coordinates (±5%) as well as the size of the outer background region considered by GrabCut (from 10% to 60%). An example result of $\texttt{GrabCut+}^{\texttt{i}}$ can be seen in figure A.3g.

A.3.2.2 Adding objectness

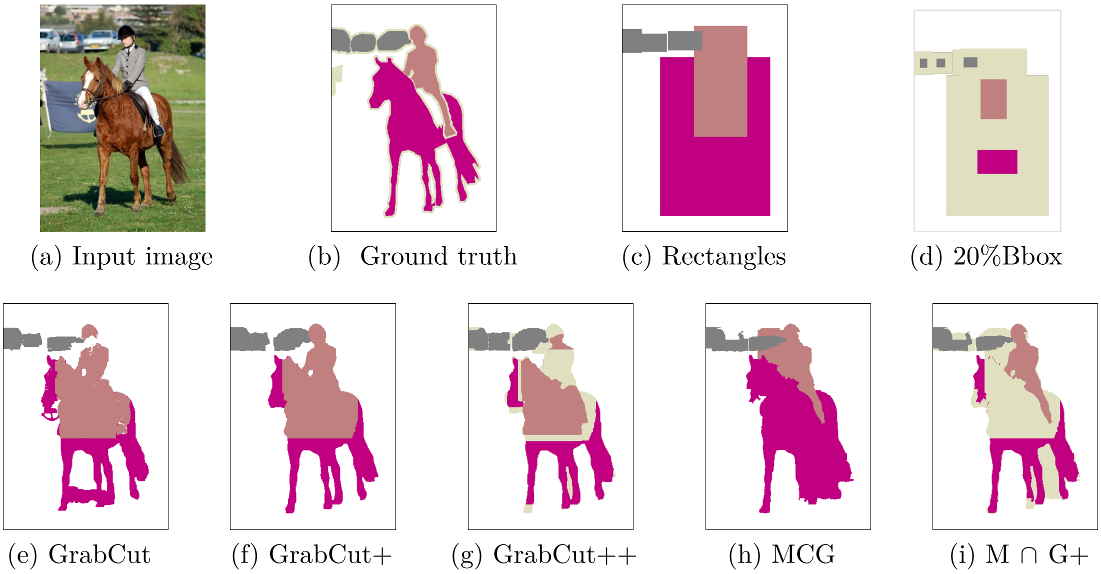
With our final approach we attempt to better incorporate the object shape priors by using segment proposals (Pont-Tuset and Gool, 2015). Segment proposals techniques are designed to generate a soup of likely object segmentations, incorporating as many “objectness” priors as useful (cue C3).

We use the state of the art proposals from MCG (Pont-Tuset *et al.*, 2017). As final stage the MCG algorithm includes a ranking based on a decision forest trained over the Pascal VOC 2012 dataset. We do *not* use this last ranking stage, but instead use all the (unranked) generated segments. Given a box annotation, we pick the highest overlapping proposal as a corresponding segment.

Building upon the insights from the baselines in section A.3.1 and A.3.2, we use the MCG segment proposals to supplement `GrabCut+`. Inside the annotated boxes, we mark as foreground pixels where both MCG and `GrabCut+` agree; the remaining ones are marked as ignore. We denote this approach as $\texttt{MCG} \cap \texttt{GrabCut+}$ or $\texttt{M} \cap \texttt{G+}$ for short.

Because MCG and `GrabCut+` provide complementary information, we can think of $\texttt{M} \cap \texttt{G+}$ as an improved version of $\texttt{GrabCut+}^{\texttt{i}}$ providing a different trade-off between precision and recall on the generated labels (see figure A.3i).

The BoxSup method (Dai *et al.*, 2015a) also uses MCG object proposals during training; however, there are important differences. They modify the training procedure so as to denoise intermediate outputs by randomly selecting high overlap proposals. In


(a) Input image (b) Ground truth (c) Rectangles (d) 20%Bbox

(e) GrabCut (f) GrabCut+ (g) GrabCut++ (h) MCG (i) M ∩ G+

Figure A.3: Example of the different segmentations obtained starting from a bounding box annotation. Grey/pink/magenta indicate different object classes, white is background, and ignore regions are beige. `M ∩ G+` denotes `MCG ∩ GrabCut+`.

comparison, our approach keeps the training procedure unmodified and simply generates input labels. Our approach also uses ignore regions, while BoxSup does not explore this dimension. Finally, BoxSup uses a longer training than our approach.

Section A.4 shows results for the semantic labelling task, compares different methods and different supervision regimes. In section A.5 we show that the proposed approach is also suitable for the instance segmentation task.

A.4 Semantic labelling results

Our approach is equally suitable (and effective) for weakly supervised instance segmentation as well as for semantic labelling. However, only the latter has directly comparable related work. We thus focus our experimental comparison efforts on the semantic labelling task. Results for instance segmentation are presented in section A.6.

Section A.4.1 discusses the experimental setup, evaluation, and implementation details for semantic labelling. Section A.4.2 presents our main results, contrasting the methods from section A.3 with the current state of the art. Section A.4.3 further expands these results with a more detailed analysis, and presents results when using more supervision (semi-supervised case).

A.4.1 Experimental setup

Datasets. We evaluate the proposed methods on the Pascal VOC12 segmentation benchmark (Everingham *et al.*, 2014). The dataset consists of 20 foreground object classes and one background class. The segmentation part of the VOC12 dataset contains 1 464 training, 1 449 validation, and 1 456 test images. Following previous work (Chen *et al.*, 2015; Dai *et al.*, 2015a), we extend the training set with the annotations provided by Hariharan *et al.* (2011), resulting in an augmented set of 10 582 training images.

In some of our experiments, we use additional training images from the COCO (Lin *et al.*, 2014) dataset. We only consider images that contain any of the 20 Pascal classes and (following Zheng *et al.* (2015)) only objects with a bounding box area larger than 200 pixels. After this filtering, 99 310 images remain (from training and validation sets), which are added to our training set. When using COCO data, we first pre-train on COCO and then fine-tune over the Pascal VOC12 training set.

All of the COCO and Pascal training images come with semantic labelling annotations (for fully supervised case) and bounding box annotations (for weakly supervised case).

Evaluation. We use the "comp6" evaluation protocol. The performance is measured in terms of pixel intersection-over-union averaged across 21 classes (mIoU). Most of our results are shown on the validation set, which we use to guide our design choices. Final results are reported on the test set (via the evaluation server) and compared with other state-of-the-art methods.

Implementation details. For all our experiments we use the DeepLab-LargeFOV network, using the same train and test parameters as Chen *et al.* (2015). The model is initialized from a VGG16 network pre-trained on ImageNet (Simonyan and Zisserman, 2015). We use a mini-batch of 30 images for SGD and initial learning rate of 0.001, which is divided by 10 after a 2k/20k iterations (for Pascal/COCO). At test time, we apply DenseCRF (Krähenbühl and Koltun, 2011). Our network and post-processing are comparable to the ones used in Dai *et al.* (2015a) and Papandreou *et al.* (2015).

Note that multiple strategies have been considered to boost test time results, such as multi-resolution or model ensembles (Chen *et al.*, 2015; Kokkinos, 2016). Here we keep the approach simple and fixed. In all our experiments we use a fixed training and test time procedure. Across experiments we only change the input training data that the networks gets to see.

A.4.2 Main results

Box results. Figure A.4 presents the results for the recursive training of the box baselines from section A.3.1. We see that the `Naive` scheme, a recursive training from rectangles disregarding post-processing stages, leads to poor quality. However, by using the suggested three post-processing stages, the `Box` baseline obtains a significant gain, getting tantalisingly close to the best reported results on the task (Dai *et al.*, 2015a). Adding ignore regions inside the rectangles ($\texttt{Box} \rightarrow \texttt{Box}^{\texttt{i}}$) provides a clear gain and leads by itself to state of the art results.

Figure A.4 also shows the result of using longer training for fully supervised case. When using ground truth semantic segmentation annotations, one training round is enough to achieve good performance; longer training brings marginal improvement.

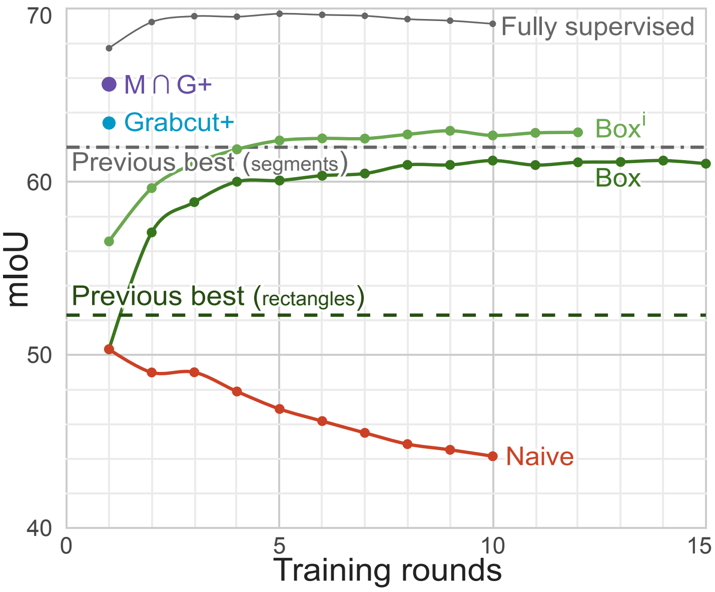


Figure A.4: Segmentation quality versus training round for different approaches, see also tables A.1 and A.2. Pascal VOC12 validation set results. "Previous best (rectangles/segments)" corresponds to WSSL_R/BoxSup_MCG in table A.2.

As discussed in section A.3.1, reaching good quality for `Box`/`Box`$^\text{i}$ requires multiple training rounds instead, and performance becomes stable from round 5 onwards. Instead, `GrabCut+`/`M ∩ G+` do not benefit from additional training rounds.

Box-driven segment results. Table A.1 evaluates results on the Pascal VOC12 validation set. It indicates the `Box`/`Box`$^\text{i}$ results after 10 rounds, and `MCG`/`GrabCut+`/`GrabCut+`$^\text{i}$/`M ∩ G+` results after one round. "Fast-RCNN" is the result using detections (Girshick, 2015) to generate semantic labels (lower-bound), "GT Boxes" considers the box annotations as labels, and $\text{DeepLab}_\text{ours}$ indicates our fully supervised segmentation network result obtained with a training length equivalent to three training rounds (upper-bound for our results). We see in the results that using ignore regions systematically helps (trading-off recall for precision), and that `M ∩ G+` provides better results than `MCG` and `GrabCut+` alone.

Table A.2 indicates the box-driven segment results after 1 training round and shows comparison with other state of the art methods, trained from boxes only using either Pascal VOC12, or VOC12+COCO data. BoxSup_R and WSSL_R both feed the network with rectangle segments (comparable to `Box`$^\text{i}$), while WSSL_S and BoxSup_MCG exploit arbitrary shaped segments (comparable to `M ∩ G+`). Although our network and post-processing is comparable to the ones in Dai *et al.* (2015a) and Papandreou *et al.* (2015), there are differences in the exact training procedure and parameters.

Overall, our results indicate that—without modifying the training procedure—`M ∩ G+` is able to improve over previously reported results and reach 95% of the fully-supervised training quality. By training with COCO data (Lin *et al.*, 2014) before fine-tuning for Pascal VOC12, we see that with enough additional bounding boxes we can match the

Method		val. mIoU
-	Fast-RCNN	44.3
	GT Boxes	62.2
Weakly supervised	Box	61.2
	Box$^{\text{i}}$	62.7
	MCG	62.6
	GrabCut+	63.4
	GrabCut+$^{\text{i}}$	64.3
	M ∩ G+	**65.7**
Fully supervised	DeepLab$_{\text{ours}}$	69.1

Table A.1: Weakly supervised semantic labelling results for our baselines. Trained using Pascal VOC12 bounding boxes alone, validation set results. DeepLab$_{\text{ours}}$ (Chen *et al.*, 2015) indicates our fully supervised result.

full supervision from Pascal VOC 12 (68.9 versus 69.1). This shows that the labelling effort could be significantly reduced by replacing segmentation masks with bounding box annotations.

A.4.3 Additional results

Semi-supervised case. Table A.2 compares results in the semi-supervised modes considered by Dai *et al.* (2015a) and Papandreou *et al.* (2015), where some of the images have full supervision, and some have only bounding box supervision. Training with 10% of Pascal VOC12 semantic labelling annotations does not bring much gain to the performance (65.7 versus 65.8), this hints at the high quality of the generated M ∩ G+ input data.

By using ground-truth annotations on Pascal plus bounding box annotations on COCO, we observe 2.5 points gain (69.1 → 71.6 , see table A.2). This suggests that the overall performance could be further improved by using extra training data with bounding box annotations.

Boundaries supervision. Our results from MCG, GrabCut+, and M ∩ G+ all indirectly include information from the BSDS500 dataset (Arbeláez *et al.*, 2011) via the HED boundary detector (Xie and Tu, 2015). These results are fully comparable to BoxSup-MCG (Dai *et al.*, 2015a), to which we see a clear improvement. Nonetheless one would like to know how much using dense boundary annotations from BSDS500 contributes to the results. We use the weakly supervised boundary detection technique from (Khoreva *et al.*, 2016) to learn boundaries directly from the Pascal VOC12 box annotations. Training M ∩ G+ using weakly supervised HED boundaries results in 1 point loss compared to using the BSDS500 (64.8 versus 65.7 mIoU on Pascal VOC12 validation set). We see then that although the additional supervision does bring some help, it has a minor

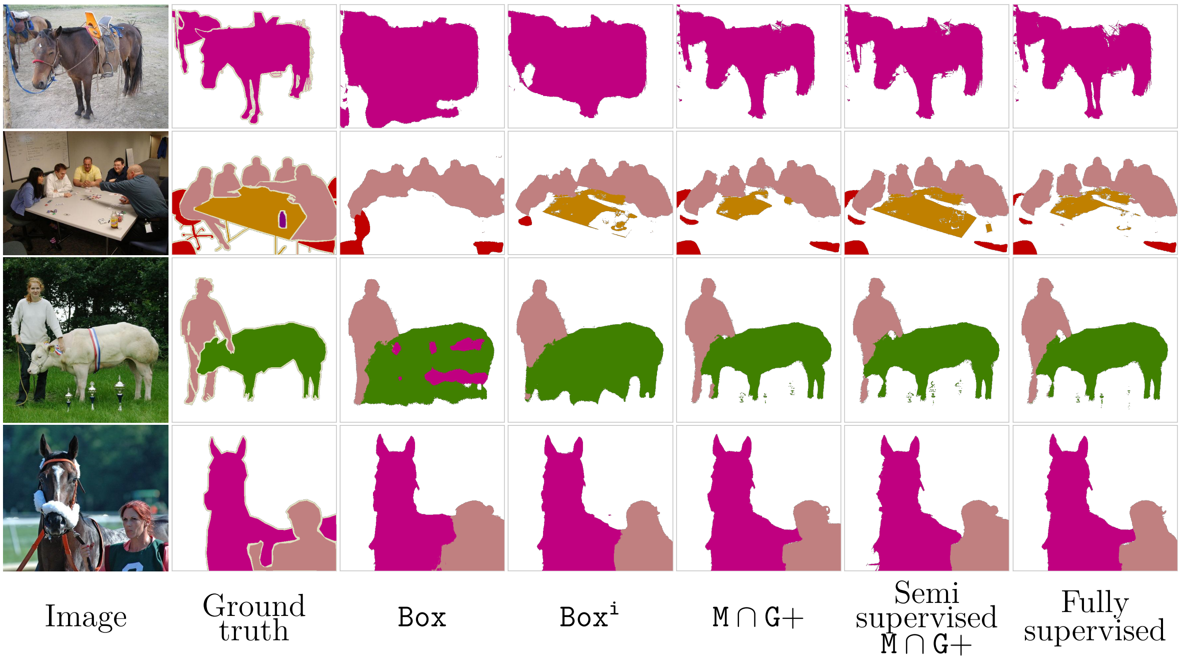


Figure A.5: Qualitative results on VOC12. Visually, the results from our weakly supervised method $\mathtt{M} \cap \mathtt{G+}$ are hardly distinguishable from the fully supervised ones.

effect and our results are still rank at the top even when we use only Pascal VOC12 + ImageNet pre-training.

Different convnet results. For comparison purposes with Dai *et al.* (2015a) and Papandreou *et al.* (2015) we used DeepLabv1 with a VGG-16 network in our experiments. To show that our approach also generalizes across different convnets, we also trained DeepLabv2 with a ResNet101 network (Chen *et al.*, 2016). Table A.3 presents the results.

Similar to the case with VGG-16, our weakly supervised approach $\mathtt{M} \cap \mathtt{G+}$ reaches 93%/95% of the fully supervised case when training with VOC12/VOC12+COCO, and the weakly supervised results with COCO data reach similar quality to full supervision with VOC12 only.

Super-vision	#GT images	#Weak images	Method	val. set mIoU	test set mIoU	test set FS%
VOC12 (V)						
Weak	-	V 10k	Bearman *et al.* (2016)	45.1	-	-
			BoxSup$_{\mathrm{R}}$ (Dai *et al.*, 2015a)	52.3	-	-
			WSSL$_{\mathrm{R}}$(Papandreou *et al.*, 2015)	52.5	54.2	76.9
			WSSL$_{\mathrm{S}}$(Papandreou *et al.*, 2015)	60.6	62.2	88.2
			BoxSup$_{\mathrm{MCG}}$(Dai *et al.*, 2015a)	62.0	64.6	91.6
			Box$^{\mathtt{i}}$	62.7	63.5	90.0
			M ∩ G+	**65.7**	**67.5**	**95.7**
Semi	V 1.4k	V 9k	WSSL$_{\mathrm{R}}$(Papandreou *et al.*, 2015)	62.1	-	-
			BoxSup$_{\mathrm{MCG}}$(Dai *et al.*, 2015a)	63.5	66.2	93.9
			WSSL$_{\mathrm{S}}$(Papandreou *et al.*, 2015)	65.1	66.6	94.5
			M ∩ G+	**65.8**	**66.9**	**94.9**
Full	V 10k	-	BoxSup (Dai *et al.*, 2015a)	63.8	-	-
			WSSL (Papandreou *et al.*, 2015)	67.6	70.3	99.7
			DeepLab$_{ours}$ (Chen *et al.*, 2015)	69.1	70.5	100
VOC12 + COCO (V+C)						
Weak	-	V+C 110k	Box$^{\mathtt{i}}$	65.3	66.7	91.1
			M ∩ G+	**68.9**	**69.9**	**95.5**
Semi	V 10k	C 123k	BoxSup$_{\mathrm{MCG}}$(Dai *et al.*, 2015a)	68.2	71.0	97.0
		C 100k	M ∩ G+	**71.6**	**72.8**	**99.5**
Full	V+C 133k	-	BoxSup (Dai *et al.*, 2015a)	68.1	-	-
			WSSL (Papandreou *et al.*, 2015)	71.7	73	99.7
	V+C 110k		DeepLab$_{ours}$ (Chen *et al.*, 2015)	72.3	73.2	100

Table A.2: Semantic labelling results for validation and test set; under different training regimes with VOC12 (V) and COCO data (C). Underline indicates full supervision baselines, and bold are our best weakly- and semi-supervised results. FS%: performance relative to the best fully supervised model (DeepLab$_{ours}$). Discussion in sections A.4.2 and A.4.3.

Supervision	Method	mIoU	FS%
VOC12			
Weak	M ∩ G+	69.4	93.2
Full	DeepLabv2-ResNet101 (Chen *et al.*, 2016)	74.5	100
VOC12 + COCO			
Weak	M ∩ G+	74.2	95.5
Full	DeepLabv2-ResNet101 (Chen *et al.*, 2016)	77.7	100

Table A.3: DeepLabv2-ResNet101 network semantic labelling results on VOC12 validation set, using VOC12 or VOC12+COCO training data. FS%: performance relative to the full supervision. Discussion in section A.4.3.

A.5 From boxes to instance segmentation

Complementing the experiments of the previous sections, we also explore a second task: weakly supervised instance segmentation. To the best of our knowledge, these are the first reported experiments on this task.

As object detection moves forward, there is a need to provide richer output than a simple bounding box around objects. Recently Hariharan *et al.* (2015), Pinheiro *et al.* (2015), and Pinheiro *et al.* (2016) explored training convnets to output a foreground versus background segmentation of an instance inside a given bounding box. Such networks are trained using pixel-wise annotations that distinguish between instances. These annotations are more detailed and expensive than semantic labelling, and thus there is interest in weakly supervised training.

The segments used for training, as discussed in section A.3.2, are generated starting from individual object bounding boxes. Each segment represents a different object instance and thus can be used directly to train an instance segmentation convnet. For each annotated bounding box, we generate a foreground versus background segmentation using the `GrabCut+` method (section A.3.2), and train a convnet to regress from the image and bounding box information to the instance segment.

A.6 Instance segmentation results

Experimental setup. We choose a purposely simple instance segmentation pipeline, based on the "hyper-columns system 2" architecture (Hariharan *et al.*, 2015). We use Fast-RCNN (Girshick, 2015) detections (post-NMS) with their class score, and for each detection estimate an associated foreground segment. We estimate the foreground using either some baseline method (e.g. GrabCut) or using convnets trained for the task (Pinheiro *et al.*, 2015; Chen *et al.*, 2016).

For our experiments we use a re-implementation of the DeepMask (Pinheiro *et al.*, 2015) architecture, and additionally we re-purpose a DeepLabv2 VGG-16 network (Chen *et al.*, 2016) for the instance segmentation task, which we name DeepLab$_{\text{BOX}}$.

Inspired by Xu *et al.* (2016) and Carreira *et al.* (2016), we modify DeepLab to accept four input channels: the input image RGB channels, plus a binary map with a bounding box of the object instance to segment. We train the network DeepLab$_{\text{BOX}}$ to output the segmentation mask of the object corresponding to the input bounding box. The additional input channel guides the network so as to segment only the instance of interest instead of all objects in the scene. The input box rectangle can also be seen as an initial guess of the desired output. We train using ground truth bounding boxes, and at test time Fast-RCNN detection boxes are used.

We train DeepMask and DeepLab$_{\text{BOX}}$ using `GrabCut+` results either over Pascal VOC12 or VOC12+COCO data (1 training round, no recursion like in section A.3.1), and test on the VOC12 validation set, the same set of images used in section A.4. The augmented annotation from Hariharan *et al.* (2011) provides per-instance segments for VOC12. We do not use CRF post-processing for neither of the networks.

Supervision	Method	$\text{mAP}^r_{0.5}$	$\text{mAP}^r_{0.75}$	ABO
-	Rectangle	21.6	1.8	38.5
	Ellipse	29.5	3.9	41.7
	MCG	28.3	5.9	44.7
	GrabCut	38.5	13.9	45.8
	GrabCut+	41.1	17.8	46.4
VOC12				
Weak	DeepMask	39.4	8.1	45.8
	$\text{DeepLab}_{\text{BOX}}$	44.8	16.3	**49.1**
Full	DeepMask	41.7	9.7	47.1
	$\text{DeepLab}_{\text{BOX}}$	47.5	20.2	51.1
VOC12 + COCO				
Weak	DeepMask	42.9	11.5	48.8
	$\text{DeepLab}_{\text{BOX}}$	46.4	18.5	**51.4**
Full	DeepMask	44.7	13.1	49.7
	$\text{DeepLab}_{\text{BOX}}$	49.4	23.7	53.1

Table A.4: Instance segmentation results on VOC12 validation set. Underline indicates the full supervision baseline, and bold are our best weak supervision results. Weakly supervised DeepMask and $\text{DeepLab}_{\text{BOX}}$ reach comparable results to full supervision. See section A.6 for details.

Following instance segmentation literature (Hariharan *et al.*, 2014a, 2015) we report in table A.4 mAP^r at IoU threshold 0.5 and 0.75. mAP^r is similar to the tradional VOC12 evaluation, but using IoU between segments instead of between boxes. Since we have a fixed set of windows, we can also report the average best overlap (ABO) (Pont-Tuset and Gool, 2015) metric to give a different perspective on the results.

Baselines. We consider five training-free baselines: simply filling in the detection rectangles (boxes) with foreground labels, fitting an ellipse inside the box, using the MCG proposal with best bounding box IoU, and using GrabCut and `GrabCut+` (see section A.3.2), initialized from the detection box.

Analysis. The results table A.4 follows the same trend as the semantic labelling results in section A.4. `GrabCut+` provides the best results among the baselines considered and shows comparable performance to DeepMask, while our proposed $\text{DeepLab}_{\text{BOX}}$ outperforms both techniques. We see that our weakly supervised approach reaches ∼95% of the quality of fully-supervised case (both on $\text{mAP}^r_{0.5}$ and ABO metrics) using two different convnets, DeepMask and $\text{DeepLab}_{\text{BOX}}$, both when training with VOC12 or VOC12+COCO.

Examples of the instance segmentation results from weakly supervised DeepMask (VOC12+COCO) are shown in figure A.6.

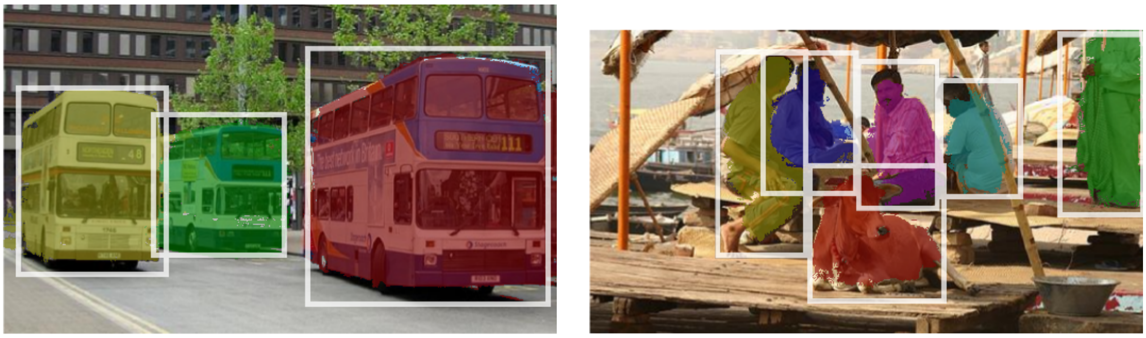


Figure A.6: Example result from our weakly supervised DeepMask (VOC12+COCO) model.

A.7 Conclusion

The series of experiments presented in this appendix provides new insights on how to train pixel-labelling convnets from bounding box annotations only. We showed that when carefully employing the available cues, recursive training using only rectangles as input can be surprisingly effective ($\texttt{Box}^{\texttt{i}}$). Even more, when using box-driven segmentation techniques and doing a good balance between accuracy and recall in the noisy training segments, we can reach state of the art performance without modifying the segmentation network training procedure ($\texttt{M}\cap\texttt{G}+$). Our results improve over previously reported ones on the semantic labelling task and reach $\sim$95% of the quality of the same network trained on the ground truth segmentation annotations (over the same data). By employing extra training data with bounding box annotations from COCO we are able to match the full supervision results. We also report the first results for weakly supervised instance segmentation, where we also reach $\sim$95% of the quality of the fully-supervised training.

Our current approach exploits existing box-driven segmentation techniques, treating each annotated box individually. In future work we would like to explore co-segmentation ideas (treating the set of annotations as a whole), and consider even weaker forms of supervision.

Exploring a Distributed Shape Representation for Object Recognition

It has long been argued that many object classes are best represented by their shape. Even though numerous shape representations have been proposed, state-of-the-art object models do not explicitly represent the object's shape. This appendix explores the complementarity between HOG based linear classifiers (DPM) and shape representations.

We propose a Distributed Shape (DiSh) representation capturing both global and local properties of the object shape. We analyse various aspects of this representation on the ETHZ Shape dataset and on the more challenging Pascal dataset. Our results indicate that shape information can be discriminative, but that it is often less important than one might expect, even for object classes that are often considered shape dominant.

This is previously unpublished work that has been conducted in 2013. Jan Hosang is the lead author.

B.1 Introduction

Shape is perceived as an important cue for object detection. Following this intuition a significant amount of work has been dedicated to the topic of shape representations (Ferrari *et al.*, 2006; Gavrila, 2007; Schindler and Suter, 2008; Maji and Malik, 2009; Schlecht and Ommer, 2011). Yet, at the time at which this research was conducted, standard object detectors (such as the Deformable Parts Model (DPM) (Felzenszwalb *et al.*, 2010)) do not explicitly model shape information.

In this appendix we revisit the importance of shape by proposing a new detector to capture local and global shape information (named DiSh) and studying its complementarity with the high quality DPM detector. First, we discuss shape representation and its use for detection in section B.2. We describe our shape aware detector in section B.3 and validate its performance for object categories where shape is considered the dominant cue (section B.5.1). Section B.4 describes how we integrate our shape features with the DPM detector. We discuss the evaluation results over Pascal VOC 2007 in section B.5.2.

The main contributions of this work are: 1) showing that the DPM is more competitive on the ETHZ Shape dataset than previously reported, 2) proposing the new DiSh detector that, despite its simplicity, obtains competitive performance on the ETHZ shape dataset, and 3) the study of its complementarity with the DPM. Extending DPM with DiSh does show improvement, however, our results indicate that these improvements are not directly related to shape information.

B.2 Encoding shape information

The shape of an object relates to the geometric properties of its boundaries, disregarding color, texture, or material information. Some works aim to explicitly encode and analyse the boundary geometry, while others do so indirectly by modeling statistics of image gradients.

Global methods describe the whole object boundary with a single geometric object—typically a closed curve—and aim at doing recognition by shape matching (Gavrila, 1998, 2007; Ravishankar *et al.*, 2008; Cootes *et al.*, 1995; Schindler and Suter, 2008). The global nature of the matching make these methods brittle to partial occlusion and noise.

To improve robustness, local shape methods encode the object's geometric properties with an ensemble of boundary fragments (Shotton *et al.*, 2008; Opelt *et al.*, 2008, 2006; Ferrari *et al.*, 2008, 2007; Leordeanu *et al.*, 2007; Danielsson and Carlsson, 2010). Most of these papers make hard decisions about which pixels belong to an edge or not, making them brittle to blur and image noise.

Template based object detectors do not aim to explicitly model the object contour, however they model the object class shape implicitly. Those methods typically build on local descriptors of image gradients that are more robust than hard edge decisions. Among them, Hough voting methods cast votes from detected (sparse) feature points (Schlecht and Ommer, 2011; Maji and Malik, 2009; Seemann *et al.*, 2005; Leibe *et al.*, 2004). More recently dense scoring is favoured via Hough Forests (Gall and Lempitsky, 2009), HOG + linear SVM (Dalal and Triggs, 2005) (used by the DPM by Felzenszwalb *et al.* (2010)), or boosted forests (Dollár *et al.*, 2009a; Zhang *et al.*, 2011; Nam *et al.*, 2011). Template based methods have difficulties handling intra-class variance and occlusions (Hoiem *et al.*, 2012b).

Our approach is inspired by local shape methods (by encoding spatial relations amongst edges), but does not restrict the relations to neighbouring edges (as explained in section B.3) and uses more robust features (HOG) than typical shape encoding methods. Closest to our work are Danielsson and Carlsson (2010) and Leordeanu *et al.* (2007), which use hard edge pixel decisions, and Dollár *et al.* (2009a), Zhang *et al.* (2011), Nam *et al.* (2011), which use techniques similar to ours, but limit themselves to pedestrian detection (while we consider diverse classes) and do not study the relation to shape representation.

B.3 Distributed shape representation (DiSh)

We want to build a model that is capable of extracting and detecting shape information in an image. To that end, we draw on four main ingredients (explained below): robust local features, a distributed representation, the use of higher-order features, and discriminative training. Our aim is not to have a model that *must* draw information from shape cues, but rather a model that *can* use shape information if it is most discriminative for the detection task.

Robust features. Edge information needs to be extracted in a manner robust to small deformations and image gradient noise. We extract local gradient statistics and encode it using Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005). To increase expressiveness we compute the HOG descriptor at multiple resolutions. The proposed method is agnostic of the low level features used.

Distributed representation. Instead of building our representation from neighbouring feature cells, we aim for a sparse and distributed representation of the object shape. On the one hand, a sparse representation forces the model to focus on the essentials of the class appearance while ignoring particularities of individual instances. On the other hand, using distributed features (i.e. located in different, disconnected areas of the image, instead of crammed together) allows to be robust to partial occlusion and noise.

Higher-order features. On top of our robust distributed features we add a non-linear stage. Higher-order features combine multiple features as input for a non-linear stage to increase the discriminative power of the learned model. Such higher-order features allow to emulate soft versions of and/or/not logic conditions on the edge features.

Discriminative training. In order to attain the sparsity constraints of the distributed representation, and to maximize classification performance the (higher-order) features are selected based on their discriminative power (via Adaboost), as detailed in section B.3.1.

B.3.1 DiSh representation learning

The DiSh representation learning consists of two steps. First, we learn a set of local shape features that represent the object shape in a distributed fashion. Second, these features are weighted to maximise object detection performance. In particular, we build the shape features by boosting HOG, using logistic regression as weak classifiers, and subsequently update the weights of the learned weak classifiers using an SVM. We call each learned weak classifier a DiSh feature.

B.3.1.1 Feature learning

In the first step Real Adaboost (Schapire and Singer, 1999) is used to learn a discriminative set of features. Each feature is a trained weak classifier h_t, greedily built to improve the performance of the strong classifier H.

$$H(p_d) = \sum_{t=1}^{T} \alpha_t \cdot h_t(I,\, p_d) \tag{B.1}$$

where I is the input image and $p_d = (x_d, y_d, l_d)$ is the position of a candidate detection to be scored. x_d and y_d specify the location in image space, while l_d specifies the location in scale space, i.e. the layer of the feature pyramid.

In the first-order version of DiSh ($k = 1$) a single low-level feature dimension is used to define a weak classifier. In the case of HOG, this corresponds to one entry of the descriptor vector, typically, the gradient magnitude of one particular gradient direction in one HOG cell. Higher-order features ($k > 1$) combine multiple entries of the low-level descriptor.

Each weak classifier h_t is a logistic regressor with k dimensional input,

$$h_t(p_d) = \frac{2}{1 - \exp(-\beta_t \cdot \mathbf{x})} - 1 \tag{B.2}$$

$$\mathbf{x} = \left[1,\, \psi_{\texttt{HOG}}(I,\, p_d)(\xi_t^0),\, \ldots,\, \psi_{\texttt{HOG}}(I,\, p_d)(\xi_t^k)\right] \tag{B.3}$$

where $\beta_t \in \mathbb{R}^{k+1}$ parametrizes the logistic regression on the feature vector $\mathbf{x} \in \mathbb{R}^{k+1}$. The feature function $\psi_{\texttt{HOG}}$ computes the HOG descriptor of image I at position p_d. For the multi-scale case, we compute an extended descriptor which concatenates HOG at multiple scales. ξ_t^i for $i \in \{1, \ldots, k\}$ indexes one dimension of the HOG (multi-scale) feature descriptor. Each of these indices correspond to a specific location, gradient orientation, and scale inside the detection window p_d.

The parameters that are optimized in each boosting iteration are β_t, p_t and θ_t. For $k = 1$, the minimization can be solved by training a logistic regressor for each possible position p_t (inside the detection window) and cell descriptor dimension θ_t (among 10^5 possibilities). The best logistic regressor is selected by evaluating them on the weighted training data and selecting the one that minimizes the exponential (Adaboost) loss. For $k \geq 2$ the search space becomes too large ($10^{5 \cdot 2}$) for an exhaustive search. We thus employ a greedy search scheme, which runs the minimization for $k = 1$, fixes $(p_1,\, \theta_1)$, and tries all possible second locations for $k = 2$. This strategy is repeated for each $k > 2$. This way, the training cost grows linearly instead of exponentially in the order k of the feature.

B.3.1.2 Feature weighting

Once the DiSh representation is learned, we can refine the weights α_t using standard linear Support Vector Machine (SVM) training. For this we interpret the weak classifier h_t as an element of a T-dimensional feature vector. We found that the SVM-trained weights perform slightly better than the ones provided by Adaboost, so we report those results only.

B.3.1.3 HOG implementation

We use the HOG version of (Felzenszwalb *et al.*, 2010) that builds a large descriptor including contrast sensitive and insensitive edges, normalized over different areas (same as (Dalal and Triggs, 2005)). This large descriptor is then linearly projected to a vector three times smaller. The final descriptor contains information about gradient magnitude and orientation specific gradient magnitude, which we distinguish in our visualizations (figures B.1 and B.3).

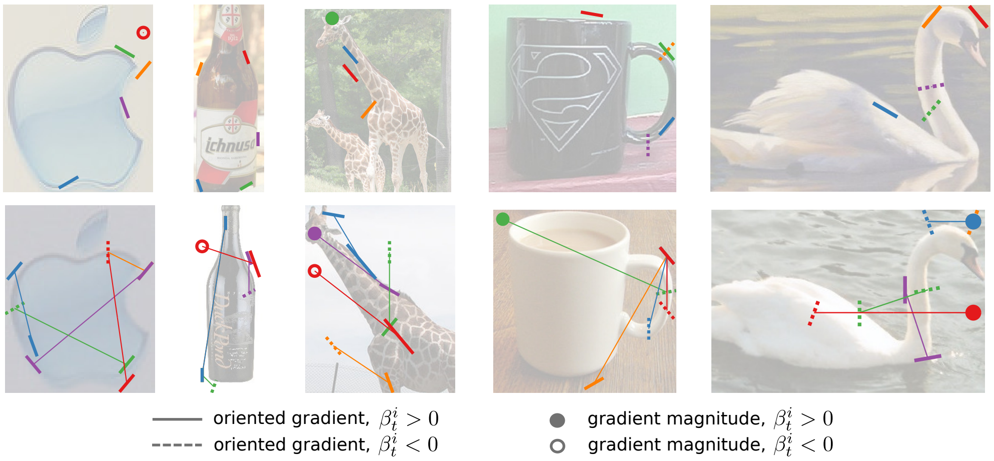

oriented gradient, $\beta_t^i > 0$
oriented gradient, $\beta_t^i < 0$
gradient magnitude, $\beta_t^i > 0$
gradient magnitude, $\beta_t^i < 0$

Figure B.1: Strongest `DiSh` activations for true positives. Each activated feature drawn with a unique colour. First row: order $k = 1$, single scale features; second row: order $k = 2$, multi-scale models. `DiSh` features activate on the object boundaries, thus encoding the class shape (see §B.3, for details).

B.3.1.4 *Learned `DiSh` representations*

To provide some intuition, we illustrate ten learned `DiSh` models in figure B.1. This figure shows the five features (weak classifiers) with strongest contribution to the correct detection score. Features are coded by their type: oriented gradient vs gradient magnitude, and positive vs negative influence on the input of the exponential in equation B.2. Each feature is drawn with a unique colour. Small miss-alignments of the features with respect to the image gradients are due to the spatial quantisation of the `HOG` descriptor.

From these examples we can see that the discriminative training of `DiSh` does indeed learn features related to the object boundary geometry, thus encoding the object class shape. It can also be seen that second order features do capture distant areas of the boundary, materialising the distributed nature of our shape representation.

B.4 DiSh representation integrated into DPM

In section B.5 we report results on the ETHZ Shape dataset, and on the Pascal VOC 2007 dataset. Since the Pascal VOC 2007 dataset contains multiple classes that are not shape dominant, we propose to extend the state-of-the-art `DPM` with `DiSh` in order to obtain competitive performance. Combining these two detectors also enables us to investigate their complementarity, as discussed in section B.5.2.

The Deformable Parts Model (`DPM`) (Felzenszwalb *et al.*, 2010) is a star-shaped constellation model, with a root filter connected to n part filters. At test time, the part filters are allowed to displace from their rest positions with some deformation cost. We connect `DiSh` to the `DPM` model by rigidly attaching a new part $n + 1$ to the root filter.

The detection score of a configuration $(p_0, \dots, p_n)$ is given by two terms: how well the image features match the model filters, and how well the position of the parts matches the model.

$$\mathrm{score}(p_0, \dots, p_n) = \sum_{i=0}^{n} \mathrm{score}_{\mathrm{app},i}(p_i) + \mathrm{score}_{\mathtt{DiSh}}(p_0) - \sum_{i=1}^{n} \mathrm{cost}_{\mathrm{def},i}(p_i) + b \tag{B.4}$$

The root part $(i = 0)$ and the `DiSh` part $n + 1$ are excluded from the sum for the deformation cost, because they are both rigidly attached to the sliding window. The definition of the root and part filters ($\mathrm{score}_{\mathrm{app},i}$), the deformation term ($\mathrm{cost}_{\mathrm{def},i}$), as well as the bias b follow the original formulation, see Felzenszwalb *et al.* (2010) for details.

Analogous to the definition of $\mathrm{score}_{\mathrm{app},0}(p_0)$, we define the score of the `DiSh` part as the product between a filter F_{n+1} and our `DiSh` representation $\phi_{\mathtt{DiSh}}$:

$$\mathrm{score}_{\mathtt{DiSh}}(p) = F_{n+1} \cdot \phi_{\mathtt{DiSh}}(I,\, p) \tag{B.5}$$

$$\text{where} \quad \phi_{\mathtt{DiSh}}(I,\, p) = [h_1(I,\, p),\, \dots,\, h_T(I,\, p)]\,. \tag{B.6}$$

$\phi_{\mathtt{DiSh}}$ is a vector of the learned weak classifiers as described in section B.3.1, and F_{n+1} is the filter learnt by the `DPM` that weights the weak classifiers and replaces α_t in equation B.1.

B.4.0.1 Training

The standard training loop of `DPM` starts by estimating the latent variables of the model, proceeds with hard negative mining, and then updates the model parameters. To learn complementary features to the `HOG` used by `DPM` we learn a `DiSh` representation after each round of negative mining, just before the model parameters update (i.e. `SVM` training). `DiSh` is trained on the same data the model is optimized on: the positive samples (including their estimated latent variables) and all mined negatives. By training `DiSh` in the inner loop of the `DPM` training, `DiSh` becomes tightly integrated and aids the model in its task to discriminate between positives and the hardest negatives.

When learning the weak classifiers we give Adaboost access to the score of the `DPM` in order to weight the training samples before learning the first new weak classifier. Each training round of the `DPM` adds 10 weak classifiers to the model. Since the number of rounds is defined by stopping criteria (Felzenszwalb *et al.*, 2010) that are training data dependent, the number of `DiSh` features vary between 170 and 210 for different Pascal classes.

B.5 Experiments

The following experiments serve two main purposes. First, we want to analyse how well shape representations perform on datasets that are not designed towards shape dominant objects. For this goal, we confirm that `DiSh` is able to encode shape information on the shape-oriented ETHZ Shape database and then explore how helpful `DiSh` is on a dataset

	Logo	Bottle	Giraffe	Mug	Swan	*average*
IoU 0.2, recall @0.4 FPPI						
Danielsson and Carlsson (2010)	95.5	92.6	93.3	97.0	**100.0**	*95.7*
Ferrari *et al.* (2007)	83.2	83.2	58.6	83.6	75.4	76.8
DPM (Girshick *et al.*, 2012)	**100.0**	**100.0**	*95.7*	**100.0**	82.4	95.6
DiSh, o2, ms	**100.0**	**100.0**	93.6	**100.0**	**100.0**	**98.7**
DPM + DiSh, o2, ms	**100.0**	**100.0**	**97.9**	93.5	**100.0**	98.3
IoU 0.5, recall @0.3 FPPI						
Ferrari *et al.* (2008)	50.0	92.9	49.0	67.8	47.1	61.4
Maji and Malik (2009)	95.0	92.9	89.6	93.6	88.2	91.9
Schlecht and Ommer (2011)[16]	81.4	93.4	70.0	74.6	90.2	81.9
Yarlagadda and Ommer (2012)	95.0	**100.0**	**91.3**	96.7	**100.0**	**96.5**
Guo *et al.* (2014)	95.7	96.3	86.7	94.7	**100.0**	94.7
DPM (Girshick *et al.*, 2012)	**100.0**	**100.0**	87.2	**96.8**	82.4	93.3
DiSh, o1, ss	**100.0**	96.4	85.1	*90.3*	94.1	93.2
DiSh, o1, ms	90.0	**100.0**	83.0	87.1	**100.0**	92.0
DiSh, o2, ms	**100.0**	**100.0**	85.1	83.9	**100.0**	*93.8*
DPM + DiSh, o2, ms	**100.0**	**100.0**	80.9	87.1	**100.0**	93.6

Table B.1: Results on ETHZ Shape. The top and bottom parts of the tables show results for two settings typically used on this dataset. oi: order i, ss/ms: single/multi scale.

which is not shape-oriented, the Pascal VOC 2007 dataset. Second, we want to do the converse and analyse how well a well established object detector that does not explicitly model shape performs on a shape-oriented dataset. To this end, we also evaluate the performance of the Deformable Parts Model (DPM) on the ETHZ Shape dataset.

B.5.1 Results on ETHZ Shape dataset

The ETHZ Shape dataset (Ferrari *et al.*, 2006) consists of 255 images containing one roughly centered object: apple logo, bottle, giraffe, mug, or swan. We follow the evaluation protocol of Ferrari *et al.* (2008) and Maji and Malik (2009).

For these experiments we train a linear SVM on top of ~ 600 DiSh features (defined by the number of DPM training rounds), as described in section B.3.1. The number of DiSh features used in DPM + DiSh is comparable to the number of features used when using DiSh alone (see section B.4). We provide results for different parameter settings for the DiSh representation: order $k = 1$, 2, and for both single scale (ss) and multi-scale (ms). In the multi-scale setup we use HOG cells at scale 1 and 2 (double size). The DPM (Girshick *et al.*, 2012) results are obtained by training one component with eight

[16] Schlecht and Ommer (2011) evaluates at 0.4 FPPI, which is easier

parts (default number of parts), and switching off the mirroring since the dataset only contains objects with similar orientation (e.g. right facing swans). All other parameters are left to default.

B.5.1.1 DiSh results

The experiments in table B.1 show that DiSh obtains competitive results, improving over five previously published methods, and getting close to the best known performance. Our best results are obtained with the order 2, multi scale DiSh representation.

Interestingly, order 1 single scale DiSh already reaches good performance, indicating that this dataset can be solved with simpler means than previously thought (i.e. boosting HOG features). We also note that the DiSh variants have different relative performance across classes; to maximize performance one would ideally want to perform per class model selection.

B.5.1.2 DPM results

Other than the novel DiSh we also evaluate the standard DPM on this dataset. Interestingly, DPM also outperforms related work, reaching results on par with DiSh order 1, single scale. Note that our DPM results are stronger than previously reported (Yarlagadda and Ommer, 2012); we assume this is due to using a single component per class, instead of the default of three components.

B.5.1.3 DPM + DiSh results

Extending the DPM by adding the order 2, multi scale DiSh features improves its performance, although it does not reach the performance of DiSh alone. On this small dataset, the DPM parts seem to provide little complementary information to the DiSh features.

From these experiments we conclude that the DiSh representation is suitable to capture the properties of the shape dominant classes of the ETHZ dataset. In section B.5.2 we explore their use in a more challenging setup.

B.5.2 Results on Pascal VOC 2007 dataset

The Pascal VOC 2007 data set (Everingham *et al.*, 2007a) is widely accepted as a challenging evaluation of general purpose object detection. Its test set contains annotations for 20 object classes in ~ 5000 images. Because of the increased difficulty, shape centric approaches are rarely evaluated on this dataset. We do all experiments using the standard evaluation protocol, using the updated average precision (AP) computation (Everingham *et al.*, 2007a).

We use as baseline the DPM from Girshick *et al.* (2012) with all default parameters and compare it to our joint DPM + DiSh training as explained in section B.4. We learn 10 weak classifiers before every SVM update, which results in a ~ 200 dimensional

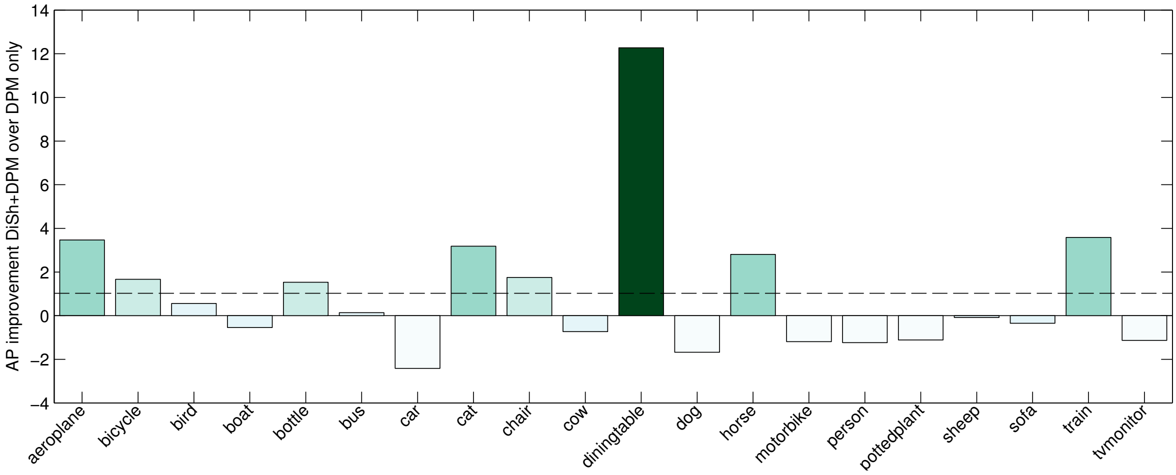


Figure B.2: AP improvement per Pascal VOC 2007 class, DPM+DiSh over DPM alone. DPM+DiSh with order 2 and multi scale features. Dashed horizontal line shows the mean AP improvement.

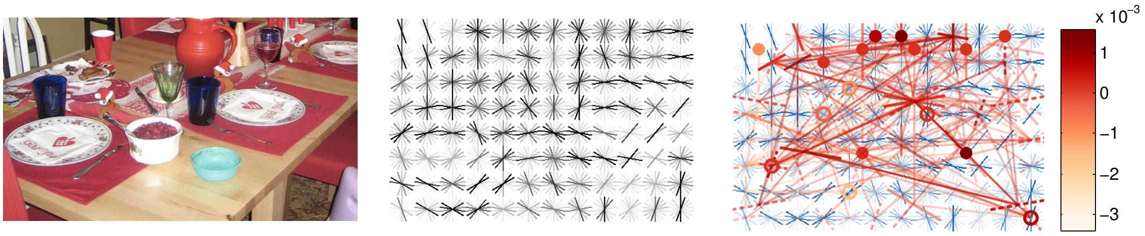


Figure B.3: Top scoring true positive for dining table class. Left: detection window content, middle: HOG representation, right: DiSh activations. HOG representation and DiSh activations color coded by their score. See figure B.1 for dots and lines labels.

DiSh feature vector per component of the final DPM + DiSh model, which has ∼ 12 000 dimensions total.

B.5.2.1 Relative improvement

Relative results for all classes are shown in figure B.2. The DPM baseline reaches 30.1 AP. On average across all classes, we improves performance 1 percent point of average precision (pp AP), which is an meaningful improvement on Pascal.

The dinning table category obtains a noticeable jump of 12pp AP, validating that our DiSh features can have a significant impact over the DPM performance. Even when ignoring this category, the mean AP still improve for DPM + DiSh versus DPM only.

Interestingly, we observe no clear trend amongst categories that one might consider “shape oriented” versus others. There is neither a clear trend regarding animate versus inanimate object categories, nor amongst categories with low or high absolute DPM performance.

B.5.2.2 Shape cues

In figure B.3 we present the DiSh feature activations for a top scoring true positive dining table. The figure shows all DiSh features associated with the active component, color coded by final score contribution. This figure shows three important aspects that we have seen reflected across the dataset. First, the visual aspect of the table category does not match the common mental model of a table. One would expect a surface supported by four legs, while the real world images essentially show a cluttered arrangement of dishes, food, and people. This mental miss-match is true for most categories. Second, HOG features provide strong discriminative power, but are not necessarily intuitive or easy to read. Third, for the dining table category, despite providing a strong detection improvement, we notice that the learned DiSh features seem not to encode the object class shape. This relates to the first two points: because built over HOG, DiSh features might not be all directly readable, and more importantly, because the shape is not necessarily the most discriminative clue for the category.

Because DiSh is trained discriminatively, it will only model the object class boundaries information if it helps for the detection task. Since the models learned on the Pascal dataset seem not to describe the object class boundaries, we thus conclude that shape is not a strong cue in real world (Pascal like) images.

From observing dining table models, we hypothesise that the strong improvement is due to the DiSh features specialising on detecting tableware.

B.5.2.3 DiSh features

To further investigate the behaviour of DiSh we present in table B.2 results when using different feature types. We do so on a subset of rigid object categories[17] (shape dominant) where one could expect a priori that DiSh behaves well.

In table B.2 we see that the main jump over the DPM happens already when using order $k = 1$ single scale features. Adding higher order and multi-scale improves some more. These results indicate that adding a simple non-linearity over HOG already enables meaningful improvement. Compared to this factor, adding high-order features able to capture the overall shape has only a minor impact.

The experiments of this section indicate that shape is not clearly a discriminative element, certainly not when put in addition to the standard DPM. Based on the performance of DPM on the ETHZ shape dataset (section B.5.1), this might well be because DPM already captures the available shape cues.

Despite the intuition that shape characterises some object categories, our experiments show that it seems to have little discriminative power when trying to detect instances in real world images, because these instances might have strong clutter, occlusion, because their appearance is dominated by context (e.g. chairs commonly have persons sited over them), or simply because their shape is ambiguous with other object categories (e.g. television versus laptop).

[17] Adding motorbike, television and boat does not change the overall trends.

		+DiSh			
	DPM	o1,ss	o1,ms	o2,ms	o3,ms
aero	28.9	30.9	30.9	**32.4**	31.3
bike	57.4	**60.1**	**60.1**	59.0	59.8
bottle	23.1	23.8	**25.1**	24.6	24.2
bus	50.3	**51.9**	51.6	50.4	50.6
car	**54.5**	51.5	51.9	52.0	51.7
chair	17.8	17.6	19.2	**19.6**	18.9
diningtable	20.0	30.2	32.2	32.2	**33.3**
train	43.6	47.4	46.9	47.1	**47.7**
average	37.0	39.2	**39.7**	**39.7**	**39.7**

Table B.2: Results on Pascal VOC 2007, increasing the complexity of DiSh representations from left to right. oi: order i, ss/ms: single/multi scale.

B.6 Conclusion

In this appendix, we explored the question of shape representation via two crossed experiments. On the one hand, we proposed the novel DiSh detector, showed that it can encode shape information, and that it obtains competitive performance on the ETHZ shape dataset. We also evaluated the effectiveness of DiSh (+DPM) on the more challenging Pascal VOC 2007 dataset. On the other hand, we have taken the DPM, a top performing method on the Pascal VOC 2007 dataset, and applied it on the ETHZ shape dataset.

Our results show that (in contrast to previously reported numbers) the DPM can be quite competitive in detecting the shape dominant categories of the ETHZ Shape dataset, even with as few as 50 training samples per class. Inversely, on Pascal VOC 2007 our shape aware DiSh method does provide improvements over the DPM, but only to a small extent.

We conclude from our experiments that shape information is useful, but less than one might expect, even for shape dominant classes.

List of Figures

List of Tables

Bibliography

A. Adams, J. Baek, and M. A. Davis (2010). Fast High-Dimensional Filtering Using the Permutohedral Lattice, in *Proc. of Eurographics 2010*. Cited on page 21.

P. Agrawal, R. Girshick, and J. Malik (2014). Analyzing the Performance of Multilayer Neural Networks for Object Recognition, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 17, 40, and 49.

B. Alexe, T. Deselaers, and V. Ferrari (2010). What is an object?, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2010*. Cited on pages 19, 82, 83, 86, 94, and 96.

B. Alexe, T. Deselaers, and V. Ferrari (2012). Measuring the objectness of image windows, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI).* Cited on pages 84 and 86.

P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik (2011). Contour Detection and Hierarchical Image Segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI).* Cited on pages 85, 152, and 156.

P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marqués, and J. Malik (2014). Multiscale Combinatorial Grouping, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 84 and 86.

H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson (2015). From Generic to Specific Deep Representations for Visual Recognition, in *CVPR Workshop on Deep Vision 2015*. Cited on pages 33, 39, and 49.

A. Banerjee (1997). Initializing Neural Networks Using Decision Trees, in *Computational Learning Theory and Natural Learning Systems: Volume IV 1997*. Cited on page 42.

A. Bar-Hillel, D. Levi, E. Krupka, and C. Goldberg (2010). Part-based Feature Synthesis for Human Detection, in *Proc. of the European Conf. on Computer Vision (ECCV) 2010*. Cited on page 28.

O. Barinova, V. Lempitsky, and P. Kholi (2012). On detection of multiple object instances using hough transforms, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI).* Cited on page 20.

J. Barron and B. Poole (2015). The Fast Bilateral Solver, in *Proc. of the European Conf. on Computer Vision (ECCV) 2015*. Cited on page 149.

A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei (2016). What's the point: Semantic segmentation with point supervision, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 149 and 158.

R. Benenson, M. Mathias, R. Timofte, and L. Van Gool (2012). Pedestrian detection at 100 frames per second, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 32.

R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool (2013). Seeking the strongest rigid detector, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 28, 30, 32, 34, 44, and 68.

R. Benenson, M. Omran, J. Hosang, and B. Schiele (2014). Ten years of pedestrian detection, what have we learned?, in *ECCV, CVRSUAD workshop 2014*. Cited on pages 15, 25, 31, 39, 42, 44, 58, 59, and 74.

Y. Bengio, N. L. Roux, P. Vincent, O. Delalleau, and P. Marcotte (2005). Convex neural networks, in *Advances in Neural Information Processing Systems (NIPS) 2005*. Cited on page 42.

H. Bilen and A. Vedaldi (2016). Weakly supervised deep detection networks, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 19.

M. Blaschko, J. Kannala, and E. Rahtu (2013). Non Maximal Suppression in Cascaded Ranking Models, in *Scandanavian Conference on Image Analysis 2013*. Cited on pages 87 and 107.

L. Bourdev and J. Brandt (2005). Robust object detection via soft cascade, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2005*. Cited on page 88.

L. Bourdev, S. Maji, T. Brox, and J. Malik (2010). Detecting people using mutually consistent poselet activations, in *Proc. of the European Conf. on Computer Vision (ECCV) 2010*. Cited on page 20.

G. Burel and D. Carel (1994). Detection and localization of faces on digital images, *Pattern Recognition Letters*, vol. 15(10), pp. 963–967. Cited on pages 20 and 129.

S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool (2017). One-Shot Video Object Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on page 19.

Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos (2016). A unified multi-scale deep convolutional neural network for fast object detection, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 16, 17, and 142.

Z. Cai, M. Saberian, and N. Vasconcelos (2015). Learning Complexity-Aware Cascades for Deep Pedestrian Detection, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 16 and 58.

J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik (2016). Human Pose Estimation with Iterative Error Feedback, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 159.

J. Carreira and C. Sminchisescu (2010). Constrained Parametric Min-Cuts for Automatic Object Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2010*. Cited on pages 82 and 86.

J. Carreira and C. Sminchisescu (2012). CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts., *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 84, 85, 86, and 112.

K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai (2011). Fusing Generic Objectness and Visual Saliency for Salient Object Detection, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2011*. Cited on page 86.

N. Chavali, H. Agrawal, A. Mahendru, and D. Batra (2015). Object-Proposal Evaluation Protocol is 'Gameable', *arXiv:1505.05836*. Cited on page 89.

G. Chen, Y. Ding, J. Xiao, and T. X. Han (2013). Detection Evolution with Multi-order Contextual Co-occurrence, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 21, 28, and 31.

L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille (2015). Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, in *Proc. of the International Conf. on Learning Representations (ICLR) 2015*. Cited on pages 149, 150, 151, 154, 156, and 158.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2016). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, *arXiv:1606.00915*. Cited on pages 150, 157, 158, and 159.

X. Chen and A. Yuille (2014). Articulated Pose Estimation with Image-Dependent Preference on Pairwise Relations, in *Advances in Neural Information Processing Systems (NIPS) 2014*. Cited on page 39.

Z. Chen, O. Lam, A. Jacobson, and M. Milford (2014). Convolutional Neural Network-based Place Recognition, in *arXiv:1411.1509 2014*. Cited on page 39.

M. M. Cheng, V. Prisacariu, S. Zheng, P. Torr, and C. Rother (2015). DenseCut: Densely Connected CRFs for Realtime GrabCut, *Computer Graphics Forum*. Cited on pages 149 and 152.

M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr (2014). BING: Binarized Normed Gradients for Objectness Estimation at 300fps, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 84, 87, and 88.

M. Cho, S. Kwak, C. Schmid, and J. Ponce (2015). Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 19.

R. G. Cinbis, J. Verbeek, and C. Schmid (2013). Segmentation Driven Object Detection with Fisher Vectors, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 82 and 83.

R. G. Cinbis, J. Verbeek, and C. Schmid (2017). Weakly supervised object localization with multi-fold multiple instance learning, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 19.

K. J. Cios and N. Liu (1992). A Machine Learning Method for Generation of a Neural Network Architecture: A Continuous ID3 Algorithm, *Trans. Neur. Netw.*, vol. 3(2), pp. 280–291. Cited on page 42.

T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham (1995). Active shape models-their training and application, *Computer Vision and Image Understanding (CVIU)*. Cited on page 164.

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on pages 11 and 144.

A. D. Costea and S. Nedevschi (2014). Word Channel Based Multiscale Pedestrian Detection Without Image Resizing and Using Only One Classifier, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 28 and 33.

F. Crete-Roffet, T. Dolmiere, P. Ladret, and M. Nicolas (2007). The Blur Effect: Perception and Estimation with a New No-Reference Perceptual Blur Metric, in *SPIE Electronic Imaging Symposium Conf Human Vision and Electronic Imaging 2007*. Cited on page 63.

J. Dai, K. He, Y. Li, S. Ren, and J. Sun (2016). Instance-sensitive fully convolutional networks, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on page 18.

J. Dai, K. He, and J. Sun (2015a). Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 19, 149, 152, 154, 155, 156, 157, and 158.

J. Dai, K. He, and J. Sun (2015b). Convolutional feature masking for joint object and stuff segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 20.

N. Dalal and B. Triggs (2005). Histograms of Oriented Gradients for Human Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2005*. Cited on pages 3, 12, 17, 20, 26, 27, 28, 30, 33, 35, 44, 57, 164, 165, and 166.

O. Danielsson and S. Carlsson (2010). Generic Object Class Detection using Boosted Configurations of Oriented Edges, in *Proc. of the Asian Conf. on Computer Vision (ACCV) 2010*. Cited on pages 164 and 169.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). ImageNet: A Large-Scale Hierarchical Image Database, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on pages 82 and 96.

T. Deselaers, B. Alexe, and V. Ferrari (2010). Localizing objects while learning their appearance, in *Proc. of the European Conf. on Computer Vision (ECCV) 2010*. Cited on page 19.

P. Dollár, R. Appel, S. Belongie, and P. Perona (2014). Fast Feature Pyramids for Object Detection, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 15, 28, 50, 51, 58, 72, and 74.

P. Dollár, R. Appel, and W. Kienzle (2012a). Crosstalk Cascades for Frame-Rate Pedestrian Detection, in *Proc. of the European Conf. on Computer Vision (ECCV) 2012*. Cited on pages 28 and 88.

P. Dollár, S. Belongie, and P. Perona (2010). The Fastest Pedestrian Detector in the West, in *Proc. of the British Machine Vision Conf. (BMVC) 2010*. Cited on page 28.

P. Dollár, Z. Tu, P. Perona, and S. Belongie (2009a). Integral Channel Features, in *Proc. of the British Machine Vision Conf. (BMVC) 2009*. Cited on pages 15, 28, 30, 33, 34, 35, 41, 44, 58, and 164.

P. Dollár, C. Wojek, B. Schiele, and P. Perona (2009b). Pedestrian Detection: A Benchmark, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on pages 11, 26, 27, and 144.

P. Dollár, C. Wojek, B. Schiele, and P. Perona (2012b). Pedestrian Detection: An Evaluation of the State of the Art, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 17, 26, 27, 41, 57, 59, 67, 69, 120, and 121.

P. Dollár and C. L. Zitnick (2013). Structured Forests for Fast Edge Detection, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on page 87.

P. Dollár and C. L. Zitnick (2015). Fast Edge Detection using Structured Forests, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 85, 86, and 87.

X. Du, M. El-Khamy, J. Lee, and L. S. Davis (2016). Fused DNN: A deep neural network fusion approach to fast and robust pedestrian detection, *arXiv:1610.03466*. Cited on page 17.

E. E. Eban, M. Schain, A. Mackey, A. Gordon, R. A. Saurous, and G. Elidan (2016). Scalable Learning of Non-Decomposable Objectives, *arXiv:1608.04802*. Cited on page 143.

I. Endres and D. Hoiem (2010). Category Independent Object Proposals, in *Proc. of the European Conf. on Computer Vision (ECCV) 2010*. Cited on pages 82 and 86.

I. Endres and D. Hoiem (2014). Category-Independent Object Proposals with Diverse Ranking, in *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 2014*. Cited on pages 84 and 86.

M. Enzweiler and D. Gavrila (2011). A multilevel mixture-of-experts framework for pedestrian classification, *IEEE Transactions on Image Processing*. Cited on page 31.

M. Enzweiler and D. M. Gavrila (2009). Monocular Pedestrian Detection: Survey and Experiments, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 17, 26, and 57.

D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov (2014). Scalable Object Detection using Deep Neural Networks, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 87 and 111.

A. Ess, B. Leibe, K. Schindler, and L. Van Gool (2008). A Mobile Vision System for Robust Multi-Person Tracking, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2008*. Cited on pages 26 and 57.

A. Ess, B. Leibe, K. Schindler, and L. Van Gool (2009). Robust Multi-Person Tracking from a Mobile Platform, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31(10), pp. 1831–1846. Cited on page 30.

M. Everingham, S. Eslami, L. Van Gool, C. Williams, J. Winn, and A. Zisserman (2014). The Pascal Visual Object Classes Challenge – a Retrospective, *International Journal of Computer Vision (IJCV)*. Cited on pages 11, 47, 82, 90, 94, 96, 100, 116, 119, 120, 123, 147, and 153.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (2007a). *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*, `http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html`. Cited on page 170.

M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool (2007b). The 2007 pascal visual object classes challenge. Cited on page 2.

P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan (2010). Object Detection with Discriminatively Trained Part-Based Models, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI).* Cited on pages 12, 14, 20, 28, 32, 40, 81, 83, 99, 101, 113, 127, 163, 164, 166, 167, and 168.

P. Felzenszwalb, D. Mcallester, and D. Ramanan (2008). A Discriminatively Trained, Multiscale, Deformable Part Model, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2008*. Cited on pages 27, 28, and 32.

P. F. Felzenszwalb and D. P. Huttenlocher (2004). Efficient Graph-Based Image Segmentation, *International Journal of Computer Vision (IJCV).* Cited on pages 85 and 88.

J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun (2011). Salient object detection by composition, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2011*. Cited on page 87.

V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid (2008). Groups of Adjacent Contour Segments for Object Detection, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI).* Cited on pages 164 and 169.

V. Ferrari, F. Jurie, and C. Schmid (2007). Accurate Object Detection with Deformable Shape Models Learnt from Images, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2007*. Cited on pages 164 and 169.

V. Ferrari, T. Tuytelaars, and L. V. Gool (2006). Object Detection by Contour Segment Networks, in *Proc. of the European Conf. on Computer Vision (ECCV) 2006*. Cited on pages 163 and 169.

J. Ferryman and A. Ellis (2010). PETS2010: Dataset and challenge, in *IEEE Advanced Video and Signal-based Surveillance (AVSS) 2010*. Cited on pages 120 and 134.

S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun (2013). Bottom-up segmentation for top-down detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on page 13.

P. Fischer, A. Dosovitskiy, and T. Brox (2014). Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT, in *arXiv:1405.5769 2014*. Cited on page 39.

K. Fragkiadaki, P. Arbeláez, P. Felsen, and J. Malik (2015). Learning to Segment Moving Objects in Videos, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 18 and 85.

J. Gall and V. Lempitsky (2009). Class-Specific Hough Forests for Object Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on page 164.

D. M. Gavrila (1998). Multi-Feature Hierarchical Template Matching Using Distance Transforms, in *Proc. of the International Conf. on Pattern Recognition (ICPR) 1998*. Cited on page 164.

D. M. Gavrila (2007). A Bayesian, Exemplar-Based Approach to Hierarchical Shape Matching, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 163 and 164.

D. M. Gavrila and S. Munder (2007). Multi-Cue Pedestrian Detection and Tracking from a Moving Vehicle, *International Journal of Computer Vision (IJCV)*, vol. 73, pp. 41–59. Cited on page 17.

A. Geiger, P. Lenz, and R. Urtasun (2012). Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite, in *Conference on Computer Vision and PatternRecognition (CVPR) 2012*. Cited on pages 11, 26, 42, 50, 57, 65, 120, and 144.

A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, and L. Van Gool (2015). Deepproposal: Hunting objects by cascading deep convolutional layers, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 19.

R. Girshick (2015). Fast R-CNN, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 13, 20, 74, 82, 83, 85, 99, 102, 107, 109, 111, 113, 127, 155, and 159.

R. Girshick, J. Donahue, T. Darrell, and J. Malik (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 12, 16, 20, 33, 40, 42, 47, 57, 59, 74, 82, 83, 85, 99, 102, 113, and 127.

R. Girshick and J. Malik (2013). Training Deformable Part Models with Decorrelated Features, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 99 and 101.

R. B. Girshick, P. F. Felzenszwalb, and D. McAllester (2012). *Discriminatively Trained Deformable Part Models, Release 5*, *http://people.cs.uchicago.edu/~rbg/latent-release5/*. Cited on pages 169 and 170.

S. Gould, R. Fulton, and D. Koller (2009). Decomposing a scene into geometric and semantically consistent regions, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2009*. Cited on page 148.

K. Grauman and T. Darrell (2005). The pyramid match kernel: Discriminative classification with sets of image features, in *Proc. ICCV 2005*. Cited on page 13.

C. Gu, J. Lim, P. Arbeláez, and J. Malik (2009). Recognition Using Regions, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on page 85.

M. Guillaumin, D. Küttel, and V. Ferrari (2014). ImageNet Auto-annotation with Segmentation Propagation, *International Journal of Computer Vision (IJCV).* Cited on pages 19 and 111.

G. Guo, Y. Wang, T. Jiang, A. Yuille, F. Fang, and W. Gao (2014). A Shape Reconstructability Measure of Object Part Importance with Applications to Object Detection and Localization, *International Journal of Computer Vision (IJCV).* Cited on page 169.

B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik (2011). Semantic Contours from Inverse Detectors, in *ICCV 2011*. Cited on pages 154 and 159.

B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik (2014a). Simultaneous Detection and Segmentation, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on page 160.

B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik (2015). Hypercolumns for object segmentation and fine-grained localization, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 149, 159, and 160.

B. Hariharan, C. L. Zitnick, and P. Dollár (2014b). Detecting Objects using Deformation Dictionaries, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on page 32.

H. Harzallah, F. Jurie, and C. Schmid (2009). Combining efficient object localization and image classification, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2009*. Cited on page 88.

K. He, X. Zhang, S. Ren, and J. Sun (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 13, 40, 82, and 83.

K. He, X. Zhang, S. Ren, and J. Sun (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 118 and 133.

K. He, X. Zhang, S. Ren, and J. Sun (2016a). Deep residual learning for image recognition, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on pages 3 and 14.

K. He, X. Zhang, S. Ren, and J. Sun (2016b). Identity Mappings in Deep Residual Networks, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on page 133.

P. Henderson and V. Ferrari (2016). End-to-end training of object class detectors for mean average precision, in *Proc. of the Asian Conf. on Computer Vision (ACCV) 2016*. Cited on pages 22, 130, and 143.

D. Hoiem, Y. Chodpathumwan, and Q. Dai (2012a). Diagnosing Error in Object Detectors, in *Proc. of the European Conf. on Computer Vision (ECCV) 2012*. Cited on pages 17 and 96.

D. Hoiem, Y. Chodpathumwan, and Q. Dai (2012b). Diagnosing Error in Object Detectors, in *Proc. of the European Conf. on Computer Vision (ECCV) 2012*. Cited on page 164.

S. Hong, H. Noh, and B. Han (2015). Decoupled deep neural network for semi-supervised semantic segmentation, in *Advances in Neural Information Processing Systems (NIPS) 2015*. Cited on page 149.

J. Hosang, R. Benenson, P. Dollár, and B. Schiele (2016a). What makes for effective detection proposals?, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 81.

J. Hosang, R. Benenson, and B. Schiele (2014). How good are detection proposals, really?, in *Proc. of the British Machine Vision Conf. (BMVC) 2014*. Cited on page 81.

J. Hosang, R. Benenson, and B. Schiele (2016b). A Convnet for Non-Maximum Suppression, in *Proc. of the German Conf. on Pattern Recognition (GCPR) 2016*. Cited on page 113.

J. Hosang, M. Omran, R. Benenson, and B. Schiele (2015). Taking a deeper look at pedestrians, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 39, 43, and 49.

A. Humayun, F. Li, and J. M. Rehg (2014). RIGOR: Recycling Inference in Graph Cuts for generating Object Regions, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 84 and 86.

I. Ivanova and M. Kubat (1995). Initialization of Neural Networks by Means of Decision Trees, *Knowledge-Based Systems*, vol. 8, pp. 333–344. Cited on page 42.

V. Jampani, M. Kiefel, and P. V. Gehler (2016). Learning Sparse High Dimensional Filters: Image Filtering, Dense CRFs and Bilateral Neural Networks, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 21.

Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell (2014). Caffe: Convolutional Architecture for Fast Feature Embedding, in *ACM International Conference on Multimedia 2014*. Cited on pages 43 and 118.

A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei (2014). Large-scale Video Classification with Convolutional Neural Networks, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on page 39.

C. Keller, D. Fernandez, and D. Gavrila (2009). Dense Stereo-based ROI Generation for Pedestrian Detection, in *Proc. of the DAGM Symposium on Pattern Recognition (DAGM) 2009*. Cited on page 26.

C. G. Keller, M. Enzweiler, M. Rohrbach, D. Fernandez Llorca, C. Schnorr, and D. M. Gavrila (2011). The benefits of dense stereo for pedestrian detection, *IEEE Transactions on Intelligent Transportation Systems*. Cited on page 30.

A. Khoreva, R. Benenson, M. Omran, M. Hein, and B. Schiele (2016). Weakly Supervised Object Boundaries, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 156.

A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung (2017). Learning Video Object Segmentation from Static Images. Cited on page 19.

M. Kiefel, V. Jampani, and P. V. Gehler (2014). Permutohedral lattice CNNs, in *ICLR workshop 2014*. Cited on page 21.

J. Kim and K. Grauman (2012). Shape Sharing for Object Segmentation, in *Proc. of the European Conf. on Computer Vision (ECCV) 2012*. Cited on page 87.

D. Kingma and J. Ba (2015). Adam: A Method for Stochastic Optimization, in *Proc. of the International Conf. on Learning Representations (ICLR) 2015*. Cited on page 118.

I. Kokkinos (2016). Pushing the Boundaries of Boundary Detection using Deep Learning, in *Proc. of the International Conf. on Learning Representations (ICLR) 2016*. Cited on pages 149 and 154.

V. Kolmogorov and R. Zabih (2004). What Energy Functions Can Be Minimized via Graph Cuts?., *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 26(2), pp. 147–159. Cited on page 149.

P. Kontschieder, S. Rota Bulò, M. Donoser, M. Pelillo, and H. Bischof (2012). Evolutionary Hough Games for Coherent Object Detection, *Computer Vision and Image Understanding (CVIU)*. Cited on page 20.

P. Krähenbühl and V. Koltun (2011). Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials, in *Advances in Neural Information Processing Systems (NIPS) 2011*. Cited on pages 21, 149, 151, and 154.

P. Krähenbühl and V. Koltun (2014). Geodesic Object Proposals, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 84 and 86.

P. Krähenbühl and V. Koltun (2015). Learning to propose objects, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 149.

A. Krizhevsky (2009). *Learning Multiple Layers of Features from Tiny Images*, Master's thesis, University of Toronto. Cited on page 43.

A. Krizhevsky, I. Sutskever, and G. E. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks, in *Advances in Neural Information Processing Systems (NIPS) 2012*. Cited on pages 3, 12, 14, 39, 40, 42, 43, and 47.

W. Kuo, B. Hariharan, and J. Malik (2015). DeepBox: Learning Objectness with Convolutional Networks. Cited on page 112.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE*. Cited on page 15.

D. Lee, G. Cha, M.-H. Yang, and S. Oh (2016). Individualness and determinantal point processes for pedestrian detection, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on page 20.

B. Leibe, A. Leonardis, and B. Schiele (2004). Combined Object Categorization and Segmentation With An Implicit Shape Model, in *ECCV workshop on Statistical Learning in Computer Vision 2004*. Cited on page 164.

B. Leibe, A. Leonardis, and B. Schiele (2008). Robust Object Detection with Interleaved Categorization and Segmentation, *International Journal of Computer Vision (IJCV)*. Cited on page 20.

V. Lempitsky, P. Kohli, C. Rother, and T. Sharp (2009). Image segmentation with a bounding box prior, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2009*. Cited on page 149.

M. Leordeanu, M. Hebert, and R. Sukthankar (2007). Beyond Local Appearance: Category Recognition from Pairwise Interactions of Simple Features, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2007*. Cited on page 164.

D. Levi, S. Silberstein, , and A. Bar-Hillel (2013). Fast multiple-part based object detection using KD-Ferns, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on page 28.

J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan (2015). Scale-aware fast R-CNN for pedestrian detection, *arXiv:1510.08160*. Cited on page 58.

J. Lim, C. L. Zitnick, and P. Dollár (2013). Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 33 and 86.

D. Lin, J. Dai, J. Jia, K. He, and J. Sun (2016a). ScribbleSup: Scribble-Supervised Convolutional Networks for Semantic Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 149.

G. Lin, C. Shen, A. van dan Hengel, and I. Reid (2016b). Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 149.

T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie (2016c). Feature Pyramid Networks for Object Detection, *arXiv:1612.03144*. Cited on page 144.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). Microsoft COCO: Common Objects in Context, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 11, 82, 97, 121, 144, 147, 154, and 155.

Z. Lin and L. Davis (2008). A Pose-Invariant Descriptor for Human Detection and Segmentation, in *Proc. of the European Conf. on Computer Vision (ECCV) 2008*. Cited on page 28.

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed (2016). SSD: Single Shot MultiBox Detector, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 13, 16, 17, 18, and 127.

J. Long, E. Shelhamer, and T. Darrell (2015). Fully Convolutional Networks for Semantic Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 148.

D. G. Lowe (2004). Distinctive Image Features from Scale-Invariant Keypoints., *International Journal of Computer Vision (IJCV)*. Cited on pages 3 and 12.

P. Luo, Y. Tian, X. Wang, and X. Tang (2014). Switchable Deep Network for Pedestrian Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 15, 28, 33, 42, and 45.

W. Luo and T.-K. Kim (2013). Generic Object Crowd Tracking by Multi-Task Learning., in *Proc. of the British Machine Vision Conf. (BMVC) 2013*. Cited on page 18.

S. Maji, A. Berg, and J. Malik (2008). Classification using intersection kernel SVMs is efficient, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2008*. Cited on pages 28 and 30.

S. Maji and J. Malik (2009). Object detection using a max-margin Hough transform, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on pages 163, 164, and 169.

T. Malisiewicz and A. A. Efros (2007). Improving Spatial Support for Objects via Multiple Segmentations, in *Proc. of the British Machine Vision Conf. (BMVC) 2007*. Cited on page 112.

S. Manén, M. Guillaumin, and L. Van Gool (2013). Prime Object Proposals with Randomized Prim's Algorithm, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 84, 85, and 112.

J. Marin, D. Vazquez, A. Lopez, J. Amores, and B. Leibe (2013). Random Forests of Local Experts for Pedestrian Detection, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 28, 31, and 33.

M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool (2014). Face detection without bells and whistles, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on page 118.

M. Mathias, R. Benenson, R. Timofte, and L. Van Gool (2013). Handling Occlusions with Franken-classifiers, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on page 28.

K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool (2005). A comparison of affine region detectors, *International Journal of Computer Vision (IJCV)*. Cited on pages 83 and 90.

A. Milan, S. Roth, and K. Schindler (2014). Continuous Energy Minimization for Multitarget Tracking, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 120.

D. Mishkin and J. Matas (2016). All you need is a good init, in *Proc. of the International Conf. on Learning Representations (ICLR) 2016*. Cited on page 133.

D. Mrowca, M. Rohrbach, J. Hoffman, R. Hu, K. Saenko, and T. Darrell (2015). Spatial semantic regularisation for large scale object detection, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 20.

W. Nam, P. Dollár, and J. H. Han (2014). Local Decorrelation For Improved Detection, in *Advances in Neural Information Processing Systems (NIPS) 2014*. Cited on pages 15, 27, 34, 35, 39, 40, 51, 58, 73, and 74.

W. Nam, B. Han, and J. Han (2011). Improving Object Localization Using Macrofeature Layout Selection, in *ICCV Workshop on Visual Surveillance 2011*. Cited on pages 28 and 164.

M. Niepert, M. Ahmed, and K. Kutzkov (2016). Learning Convolutional Neural Networks for Graphs, in *Proc. of the International Conf. on Machine learning (ICML) 2016*. Cited on page 21.

A. Opelt, A. Pinz, and A. Zisserman (2006). A Boundary-Fragment-Model for Object Detection, in *Proc. of the European Conf. on Computer Vision (ECCV) 2006*. Cited on page 164.

A. Opelt, A. Pinz, and A. Zisserman (2008). Learning an Alphabet of Shape and Appearance for Multi-class Object Detection, *International Journal of Computer Vision (IJCV)*. Cited on page 164.

W. Ouyang, P. Luo, X. Zeng, S. Qiu, Y. Tian, H. Li, S. Yang, Z. Wang, Y. Xiong, C. Qian, Z. Zhu, R. Wang, C. C. Loy, X. Wang, and X. Tang (2014). DeepID-Net: multi-stage and deformable deep convolutional neural networks for object detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on page 40.

W. Ouyang and X. Wang (2012). A Discriminative Deep Model for Pedestrian Detection with Occlusion Handling, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on pages 14, 28, and 33.

W. Ouyang and X. Wang (2013a). Joint Deep Learning for Pedestrian Detection, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 15, 28, 30, 31, 32, 33, and 45.

W. Ouyang and X. Wang (2013b). Single-pedestrian detection aided by multi-pedestrian detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 14, 17, 20, 28, 31, 35, 58, and 179.

W. Ouyang, X. Zeng, and X. Wang (2013). Modeling Mutual Visibility Relationship with a Deep Model in Pedestrian Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 28, 31, 32, and 33.

H. T. P. Dollár, Z. Tu and S. Belongie (2007). Feature Mining for Image Classification, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2007*. Cited on page 28.

S. Paisitkriangkrai, C. Shen, and A. van den Hengel (2013). Efficient pedestrian detection by directly optimize the partial area under the ROC curve, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 28 and 33.

S. Paisitkriangkrai, C. Shen, and A. van den Hengel (2014). Strengthening the Effectiveness of Pedestrian Detection with Spatially Pooled Features, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 15, 34, 37, 39, 40, 46, and 50.

S. Paisitkriangkrai, C. Shen, and A. van den Hengel (2016). Pedestrian detection with spatially pooled features and structured ensemble learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38(6), pp. 1243–1257. Cited on page 51.

C. Papageorgiou and T. Poggio (2000). A Trainable System for Object Detection, *International Journal of Computer Vision (IJCV)*. Cited on page 81.

G. Papandreou, L. Chen, K. Murphy, , and A. L. Yuille (2015). Weakly- and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 149, 154, 155, 156, 157, and 158.

D. Parikh and C. Zitnick (2011). Human-debugging of machines, in *NIPS, WCSSWC workshop 2011*. Cited on page 114.

D. Park, D. Ramanan, and C. Fowlkes (2010). Multiresolution models for object detection, in *Proc. of the European Conf. on Computer Vision (ECCV) 2010*. Cited on pages 28, 30, 31, 32, 35, and 36.

D. Park, C. L. Zitnick, D. Ramanan, and P. Dollár (2013). Exploring Weak Stabilization for Motion Feature Extraction, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 17, 28, 30, 31, 58, and 179.

D. Pathak, P. Kraehenbuehl, and T. Darrell (2015a). Constrained Convolutional Neural Networks for Weakly Supervised Segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 149.

D. Pathak, E. Shelhamer, J. Long, and T. Darrell (2015b). Fully Convolutional Multi-Class Multiple Instance Learning, in *ICLRW 2015*. Cited on page 149.

M. Pedersoli, T. Tuytelaars, and L. V. Gool (2014). Using a Deformation Field Model for Localizing Faces and Facial Points under Weak Supervision, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on page 32.

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung (2016). A benchmark dataset and evaluation methodology for video object segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 19.

F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung (2015). Fully connected object proposals for video segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 18.

P. Pinheiro and R. Collobert (2014). Recurrent Convolutional Neural Networks for Scene Labeling, in *Proc. of the International Conf. on Machine learning (ICML) 2014*. Cited on pages 33 and 148.

P. Pinheiro and R. Collobert (2015). From Image-level to Pixel-level Labeling with Convolutional Network, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 19 and 149.

P. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár (2016). Learning to Refine Object Segments, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 18, 144, and 159.

P. O. Pinheiro, R. Collobert, and P. Dollár (2015). Learning to Segment Object Candidates, in *Advances in Neural Information Processing Systems (NIPS) 2015*. Cited on pages 18, 112, 117, 121, 149, 150, and 159.

J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik (2017). Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI).* Cited on pages 149 and 152.

J. Pont-Tuset and L. V. Gool (2015). Boosting Object Proposals: From Pascal to COCO, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 149, 150, 152, and 160.

C. Premebida, J. Carreira, J. Batista, and U. Nunes (2014). Pedestrian Detection Combining RGB and Dense LIDAR Data, in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS) 2014*. Cited on page 30.

A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari (2012). Learning object class detectors from weakly annotated video, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 19.

X. Qi, Z. Liu, J. Shi, H. Zhao, and J. Jia (2016). Augmented feedback in semantic segmentation under image level supervision, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on page 19.

E. Rahtu, J. Kannala, and M. Blaschko (2011). Learning a category independent object detection cascade, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2011*. Cited on pages 84 and 86.

P. Rantalankila, J. Kannala, and E. Rahtu (2014). Generating object segmentation proposals using global and local search, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 84 and 85.

S. Ravishankar, A. Jain, and A. Mittal (2008). Multi-stage contour based detection of deformable objects, in *Proc. of the European Conf. on Computer Vision (ECCV) 2008*. Cited on page 164.

A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson (2014). CNN Features off-the-shelf: an Astounding Baseline for Recognition, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 39 and 49.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi (2016). You Only Look Once: Unified, Real-Time Object Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on pages 14 and 127.

S. Ren, K. He, R. Girshick, and J. Sun (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in *Advances in Neural Information Processing Systems (NIPS) 2015*. Cited on pages 13, 16, 18, 20, 112, 113, 123, 127, and 137.

S. H. Rezatofighi, V. Kumar BG, A. Milan, E. Abbasnejad, A. Dick, and I. Reid (2016). DeepSetNet: Predicting Sets with Deep Neural Networks, *arXiv:1611.08998*. Cited on page 21.

M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert (2011). Density-aware person detection and tracking in crowds, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2011*. Cited on page 20.

R. Rothe, M. Guillaumin, and L. Van Gool (2014). Non-maximum suppression for object detection by passing messages between windows, in *Proc. of the Asian Conf. on Computer Vision (ACCV) 2014*. Cited on pages 20 and 118.

C. Rother, V. Kolmogorov, and A. Blake (2004). Grabcut: Interactive foreground extraction using iterated graph cuts, in *ACM Trans. Graphics 2004*. Cited on pages 149 and 152.

O. Russakovsky, J. Deng, Z. Huang, A. C. Berg, and L. Fei-Fei (2013). Detecting avocados to zucchinis: what have we done, and where are we going?, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on page 19.

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 (2015). ImageNet Large Scale Visual Recognition Challenge, *International Journal of Computer Vision (IJCV)*. Cited on pages 11, 39, 40, 42, 113, and 147.

P. Sabzmeydani and G. Mori (2007). Detecting pedestrians by learning shapelet features, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2007*. Cited on page 28.

M. A. Sadeghi and A. Farhadi (2011). Recognition using Visual Phrases, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on page 20.

R. E. Schapire and Y. Singer (1999). Improved Boosting Algorithms Using Confidence-rated Predictions, *Machine Learning*. Cited on page 165.

K. Schindler and D. Suter (2008). Object detection by global contour shape, *Pattern Recognition*. Cited on pages 163 and 164.

J. Schlecht and B. Ommer (2011). Contour-based Object Detection, in *Proc. of the British Machine Vision Conf. (BMVC) 2011*. Cited on pages 163, 164, and 169.

W. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis (2009). Human Detection Using Partial Least Squares Analysis, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2009*. Cited on page 28.

E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele (2005). An evaluation of local shape-based features for pedestrian detection, in *Proc. of the British Machine Vision Conf. (BMVC) 2005*. Cited on page 164.

P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun (2014). OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks, in *Proc. of the International Conf. on Learning Representations (ICLR) 2014*. Cited on pages 20 and 33.

P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun (2013). Pedestrian Detection with Unsupervised Multi-Stage Feature Learning, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 14, 28, 30, 33, 40, 41, 44, and 46.

I. K. Sethi (1990). Entropy Nets: From Decision Trees to Neural Networks, *Proceedings of the IEEE*. Cited on page 42.

R. Setiono and W. K. Leow (1999). On Mapping Decision Trees and Neural Networks, *Knowledge Based Systems*. Cited on page 42.

N. Shapovalova, A. Vahdat, K. Cannons, T. Lan, and G. Mori (2012). Similarity constrained latent support vector machine: An application to weakly supervised action classification. Cited on page 19.

J. Shotton, A. Blake, and R. Cipolla. (2008). Multi-Scale Categorical Object Recognition Using Contour Fragments, in *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 2008*. Cited on page 164.

J. Shotton, J. Winn, C. Rother, and A. Criminisi (2009). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context, *International Journal of Computer Vision (IJCV)*. Cited on page 148.

G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah (2012). Part-based multiple-person tracking with partial occlusion handling, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 120.

K. Simonyan and A. Zisserman (2014). Two-Stream Convolutional Networks for Action Recognition in Videos, in *Advances in Neural Information Processing Systems (NIPS) 2014*. Cited on page 39.

K. Simonyan and A. Zisserman (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition, in *Proc. of the International Conf. on Learning Representations (ICLR) 2015*. Cited on pages 14, 40, and 154.

P. Siva, C. Russell, and T. Xiang (2012). In defence of negative mining for annotating weakly labelled data, in *Proc. of the European Conf. on Computer Vision (ECCV) 2012*. Cited on page 19.

P. Siva, C. Russell, T. Xiang, and L. Agapito (2013). Looking beyond the image: Unsupervised learning for object saliency and detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on page 19.

R. Stewart and M. Andriluka (2016). End-to-end people detection in crowded scenes, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on pages 20, 21, 22, 117, 130, and 145.

V. B. Subburaman, A. Descamps, and C. Carincotte (2012). Counting people in the crowd using a generic head detector, in *IEEE Advanced Video and Signal-based Surveillance (AVSS) 2012*. Cited on page 120.

J. Sun and H. Ling (2011). Scale and object aware image retargeting for thumbnail browsing, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2011*. Cited on page 19.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2015a). Going Deeper with Convolutions, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 14, 39, and 40.

C. Szegedy, S. Reed, D. Erhan, and D. Anguelov (2015b). Scalable, High-Quality Object Detection, *arXiv:1412.1441*. Cited on pages 82, 83, and 87.

Y. Taigman, M. Yang, M. Ranzato, and L. Wolf (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on page 46.

K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei (2014). Co-localization in real-world images, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 19 and 111.

M. Tang, I. Ben Ayed, D. Marin, and Y. Boykov (2015a). Secrets of GrabCut and Kernel K-means, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 149 and 152.

S. Tang, B. Andres, M. Andriluka, and B. Schiele (2015b). Subgraph Decomposition for Multi-Target Tracking, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 20 and 118.

S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele (2013). Learning people detectors for tracking in crowded scenes, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 120, 121, and 134.

S. Tang, M. Andriluka, and B. Schiele (2012). Detection and Tracking of Occluded People, in *Proc. of the British Machine Vision Conf. (BMVC) 2012*. Cited on page 20.

T. Taniai, Y. Matsushita, and T. Naemura (2015). Superdifferential Cuts for Binary Energies, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 149.

Y. Tian, P. Luo, X. Wang, and X. Tang (2015a). Deep Learning Strong Parts for Pedestrian Detection, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 16 and 58.

Y. Tian, P. Luo, X. Wang, and X. Tang (2015b). Pedestrian Detection aided by Deep Learning Semantic Tasks, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 16, 59, 74, and 76.

P. Tokmakov, K. Alahari, and C. Schmid (2016). Learning Motion Patterns in Videos, *arXiv:1612.07217*. Cited on page 18.

J. Tompson, A. Jain, Y. LeCun, and C. Bregler (2014). Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation, in *Advances in Neural Information Processing Systems (NIPS) 2014*. Cited on page 39.

A. Torralba, K. P. Murphy, and W. T. Freeman (2007). Sharing Visual Features for Multiclass and Multiview Object Detection, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 88.

Z. Tu and X. Bai (2010). Auto-context and its application to high-level vision tasks and 3D brain image segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 21, 31, and 115.

T. Tuytelaars (2010). Dense interest points, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2010*. Cited on page 83.

T. Tuytelaars and K. Mikolajczyk (2008). Local invariant feature detectors: a survey, *Foundations and Trends in Computer Graphics and Vision*. Cited on page 83.

J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders (2013). Selective search for object recognition, *International Journal of Computer Vision (IJCV)*. Cited on pages 40, 84, and 85.

K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders (2011). Segmentation As Selective Search for Object Recognition, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2011*. Cited on pages 12, 82, 83, and 85.

M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool (2014). SEEDS: Superpixels Extracted via Energy-Driven Sampling, *International Journal of Computer Vision (IJCV)*. Cited on page 87.

M. Van Den Bergh, G. Roig, X. Boix, S. Manen, and L. Van Gool (2013). Online video seeds for temporal window objectness, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on page 87.

A. Vezhnevets and V. Ferrari (2015). Object localization in ImageNet by looking out of the window, in *Proc. of the British Machine Vision Conf. (BMVC) 2015*. Cited on page 21.

S. Vicente, C. Rother, and V. Kolmogorov (2011). Object cosegmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on page 111.

O. Vinyals, S. Bengio, and M. Kudlur (2016). Order matters: Sequence to sequence for sets, in *Proc. of the International Conf. on Learning Representations (ICLR) 2016*. Cited on pages 21 and 142.

P. Viola and M. Jones (2004). Robust Real-Time Face Detection, in *International Journal of Computer Vision (IJCV) 2004*. Cited on pages 20, 28, 81, and 88.

P. Viola, M. Jones, and D. Snow (2003). Detecting pedestrians using patterns of motion and appearance, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2003*. Cited on pages 27, 35, and 139.

P. Viola, M. Jones, and D. Snow (2005). Detecting pedestrians using patterns of motion and appearance, *International Journal of Computer Vision (IJCV)*, vol. 63(2), pp. 153–161. Cited on page 17.

S. Walk, N. Majer, K. Schindler, and B. Schiele (2010). New Features and Insights for Pedestrian Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2010*. Cited on pages 28, 30, 33, and 40.

L. Wan, D. Eigen, and R. Fergus (2015). End-to-End Integration of a Convolutional Network, Deformable Parts Model and Non-Maximum Suppression, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 22.

X. Wang, X. Han, and S. Yan (2009). An HOG-LBP human detector with partial occlusion handling, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2009*. Cited on pages 28 and 33.

X. Wang, M. Yang, S. Zhu, and Y. Lin (2013). Regionlets for Generic Object Detection, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 39, 41, 82, and 83.

Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, Y. Zhao, and S. Yan (2015). STC: A Simple to Complex Framework for Weakly-supervised Semantic Segmentation, *arXiv:1509.03150*. Cited on page 149.

P. Wohlhart, M. Donoser, P. M. Roth, and H. Bischof (2012). Detecting Partially Occluded Objects with an Implicit Shape Model Random Field, in *Proc. of the Asian Conf. on Computer Vision (ACCV) 2012*. Cited on page 20.

C. Wojek, G. Dorkó, A. Schulz, and B. Schiele (2008). Sliding-windows for rapid object class localization: A parallel technique, in *Proc. of the DAGM Symposium on Pattern Recognition (DAGM) 2008*. Cited on page 20.

C. Wojek and B. Schiele (2008). A Performance Evaluation of Single and Multi-Feature People Detection, in *Proc. of the DAGM Symposium on Pattern Recognition (DAGM) 2008*. Cited on pages 17, 28, 30, and 33.

C. Wojek, S. Walk, and B. Schiele (2009). Multi-cue onboard pedestrian detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on page 26.

F. Xiao and Y. Jae Lee (2016). Track and segment: An iterative unsupervised approach for video object proposals, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 18.

S. Xie and Z. Tu (2015). Holistically-Nested Edge Detection, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 152 and 156.

J. Xu, A. Schwing, and R. Urtasun (2015). Learning To Segment under Various Forms of Weak Supervision, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 149.

N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang (2016). Deep Interactive Object Selection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 159.

J. Yan, Z. Lei, L. Wen, and S. Z. Li (2014). The Fastest Deformable Part Model for Object Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on page 32.

J. Yan, Y. Yu, X. Zhu, Z. Lei, and S. Z. Li (2015). Object Detection by Labeling Superpixels, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 20.

J. Yan, X. Zhang, Z. Lei, S. Liao, and S. Z. Li (2013). Robust Multi-Resolution Pedestrian Detection in Traffic Scenes, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 28, 30, 31, and 32.

J. Yao, S. Fidler, and R. Urtasun (2012). Describing the Scene as a Whole: Joint Object Detection, Scene Classification and Semantic Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 20.

P. Yarlagadda and B. Ommer (2012). From Meaningful Contours to Discriminative Object Shape, in *ECCV 2012*. Cited on pages 169 and 170.

F. Yu and V. Koltun (2015). Multi-scale context aggregation by dilated convolutions, *arXiv:1511.07122*. Cited on page 17.

F. Yu and V. Koltun (2016). Multi-Scale Context Aggregation by Dilated Convolutions, in *Proc. of the International Conf. on Learning Representations (ICLR) 2016*. Cited on page 149.

H. Yu, Y. Zhou, H. Qian, M. Xian, Y. Lin, D. Guo, K. Zheng, K. Abdelfatah, and S. Wang (2015). LooseCut: Interactive Image Segmentation with Loosely Bounded Boxes, *arXiv preprint arXiv:1507.03060*. Cited on page 149.

J. Zbontar and Y. LeCun (2015). Computing the Stereo Matching Cost with a Convolutional Neural Network, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 39.

P. Zehnder, E. Koller-Meier, and L. Van Gool (2008). An Efficient Shared Multi-Class Detection Cascade, in *Proc. of the British Machine Vision Conf. (BMVC) 2008*. Cited on page 88.

M. Zeiler and R. Fergus (2014). Visualizing and understanding convolutional networks, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on page 14.

X. Zeng, W. Ouyang, and X. Wang (2013). Multi-Stage Contextual Deep Learning for Pedestrian Detection, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 15 and 28.

L. Zhang, L. Lin, X. Liang, and K. He (2016a). Is Faster R-CNN Doing Well for Pedestrian Detection?, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 16, 65, and 142.

S. Zhang, C. Bauckhage, and A. B. Cremers (2014). Informed Haar-like Features Improve Pedestrian Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 15, 28, 34, 39, and 40.

S. Zhang, C. Bauckhage, D. A. Klein, and A. B. Cremers (2015a). Exploring Human Vision Driven Features for Pedestrian Detection, *IEEE Transactions on Circuits and Systems for Video Technology*. Cited on page 15.

S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele (2016b). How Far are We from Solving Pedestrian Detection?, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 55.

S. Zhang, R. Benenson, and B. Schiele (2015b). Filtered channel features for pedestrian detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 15, 16, 27, 57, 58, 59, 60, 61, 64, 66, 73, 74, and 175.

Z. Zhang, J. Warrell, and P. H. S. Torr (2011). Proposal Generation for Object Detection using Cascaded Ranking SVMs, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on pages 87 and 164.

Q. Zhao, Z. Liu, and B. Yin (2014). Cracking BING and Beyond, in *Proc. of the British Machine Vision Conf. (BMVC) 2014*. Cited on pages 87, 88, 94, and 97.

S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr (2015). Conditional Random Fields as Recurrent Neural Networks, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 21, 149, and 154.

B. Zhou, J. Xiao, A. Lapedriza, A. Torralba, and A. Oliva (2014). Learning Deep Features for Scene Recognition using Places Database, in *Advances in Neural Information Processing Systems (NIPS) 2014*. Cited on pages 39, 42, and 48.

C. Zitnick and P. Dollár (2014). Edge Boxes: Locating Object Proposals from Edges, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 40, 43, 44, 84, 87, 107, and 109.

Publications

[9] *Learning non-maximum suppression.*
Jan Hosang, Rodrigo Benenson, and Bernt Schiele.
In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (**CVPR**), 2017.

[8] *Weakly supervised semantic labelling and instance segmentation*
Anna Khoreva, Rodrigo Benenson, Jan Hosang, and Bernt Schiele.
In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (**CVPR**), 2017.

[7] *A convnet for non-maximum suppression.*
Jan Hosang, Rodrigo Benenson, and Bernt Schiele.
In Proc. German Conf. on Pattern Recognition (**GCPR**), 2016.

[6] *How far are we from solving pedestrian detection?*
Shanshan Zhang, Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele.
In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (**CVPR**), 2016.

[5] *What makes for effective detection proposals?*
Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele.
IEEE Trans. on Pattern Analysis and Machine Intelligence (**PAMI**), Vol. 38, No. 4, 2016.

[4] *Taking a deeper look at pedestrians.*
Jan Hosang, Mohamed Omran, Rodrigo Benenson, and Bernt Schiele.
In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (**CVPR**), 2015.

[3] *GyroPen: Gyroscopes for Pen-Input with Mobile Phones.*
Thomas Deselaers, Daniel Keysers, Jan Hosang, and Henry Rowley.
IEEE Transactions on Human-Machine Systems (**THMS**), Vol. 45, No. 2, 2014.

[2] *Ten years of pedestrian detection, what have we learned?*
Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele.
In 2nd Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving (**CVRSUAD**) in conjuction with **ECCV** 2014, published in Computer Vision – ECCV 2014 Workshops2, 2014 Proceedings, Part II.

[1] *How good are detection proposals, really?*
Jan Hosang, Rodrigo Benenson, and Bernt Schiele.
In Proc. of the British Machine Vision Conf. (**BMVC**), 2014.