THE WHAT AND WHERE IN VISUAL OBJECT RECOGNITION

Jasper Reinout Robertus Uijlings
Copyright © by J.R.R. Uijlings
All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without written permission from the author.

ISBN 978-94-6182-047-1
THE WHAT AND WHERE IN VISUAL OBJECT RECOGNITION

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus prof. dr. D.C. van den Boom
ten overstaan van een door het college voor promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op woensdag 7 december 2011, te 16:00 uur

doors

Jasper Reinout Robertus Uijlings

geboren te Beverwijk
Promotiecommissie:

Promotor: Prof. dr ir A.W.M. Smeulders
Co-promotor: Prof. dr ir R.J.H. Scha
Overige Leden: Prof. dr J. Bergstra
              Prof. dr J. Kittler
              Prof. dr T. Gevers
              dr N. Sebe
              dr C.G.M. Snoek

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

This work was carried out within the ASCI graduate school, at the Intelligent Systems Lab Amsterdam of the University of Amsterdam. ASCI dissertation series number 245.
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td>INTRODUCTION</td>
<td>7</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>REAL-TIME VISUAL CONCEPT CLASSIFICATION</td>
<td>11</td>
</tr>
<tr>
<td>2.1</td>
<td>Introduction</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Bag-of-Words</td>
<td>12</td>
</tr>
<tr>
<td>2.3</td>
<td>Related Work</td>
<td>14</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Descriptor Extraction</td>
<td>14</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Visual Word Assignment</td>
<td>15</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Classification</td>
<td>16</td>
</tr>
<tr>
<td>2.3.4</td>
<td>Graphics Processing Unit</td>
<td>17</td>
</tr>
<tr>
<td>2.4</td>
<td>Accelerated Bag-of-Words</td>
<td>17</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Descriptor Extraction</td>
<td>17</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Visual Word Assignment</td>
<td>18</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Classification</td>
<td>19</td>
</tr>
<tr>
<td>2.5</td>
<td>Experimental Setup</td>
<td>21</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Datasets</td>
<td>21</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Baseline</td>
<td>22</td>
</tr>
<tr>
<td>2.5.3</td>
<td>Descriptors</td>
<td>23</td>
</tr>
<tr>
<td>2.5.4</td>
<td>Word Assignment</td>
<td>24</td>
</tr>
<tr>
<td>2.5.5</td>
<td>Classification</td>
<td>25</td>
</tr>
<tr>
<td>2.5.6</td>
<td>Implementation Details</td>
<td>26</td>
</tr>
<tr>
<td>2.6</td>
<td>Results</td>
<td>27</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Baseline</td>
<td>27</td>
</tr>
<tr>
<td>2.6.2</td>
<td>Descriptors</td>
<td>27</td>
</tr>
<tr>
<td>2.6.3</td>
<td>Word Assignment</td>
<td>29</td>
</tr>
<tr>
<td>2.6.4</td>
<td>Classification</td>
<td>33</td>
</tr>
<tr>
<td>2.6.5</td>
<td>The Spatial Pyramid Revisited</td>
<td>34</td>
</tr>
<tr>
<td>2.6.6</td>
<td>Supervision in Random Forests</td>
<td>35</td>
</tr>
<tr>
<td>2.7</td>
<td>Discussion on Speed</td>
<td>35</td>
</tr>
<tr>
<td>2.7.1</td>
<td>Descriptor Extraction</td>
<td>36</td>
</tr>
<tr>
<td>2.7.2</td>
<td>Word Assignment</td>
<td>36</td>
</tr>
<tr>
<td>2.7.3</td>
<td>Classification</td>
<td>37</td>
</tr>
<tr>
<td>2.8</td>
<td>Conclusions</td>
<td>37</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>THE VISUAL EXTENT OF AN OBJECT</td>
<td>41</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>41</td>
</tr>
<tr>
<td>3.2</td>
<td>Related Work</td>
<td>42</td>
</tr>
<tr>
<td>3.3</td>
<td>Methodology</td>
<td>44</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Dataset</td>
<td>45</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Evaluation Matrix</td>
<td>46</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Bag-of-Words Framework</td>
<td>46</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Analysis without knowing the object location</td>
<td>47</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Analysis using the ideal object location</td>
<td>49</td>
</tr>
<tr>
<td>3.3.6</td>
<td>Distinguishing object, surround, interior, and border</td>
<td>49</td>
</tr>
<tr>
<td>3.4</td>
<td>Results</td>
<td>50</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Classification without knowing the object location</td>
<td>50</td>
</tr>
</tbody>
</table>
INTRODUCTION

Vision comes naturally to humans. Looking around in a living room, one can effortlessly identify chairs, a table, the carpet, a cupboard somewhere in the back, or the light sources. Walking on the street, one can easily assess the traffic situation and perceive which vehicle goes where. Computers can not, not that easy that is. Where pointing at things to give them names seems very simple, in reality it is not.

Still, it would be useful to imbue computers with the same power as it is this fundamental capability upon which our understanding of the world rests. Research in this direction will yield applications such as the automatic identification of the content of a photograph, useful for search engines, or a robot which navigates, perceives, and acts in this world. And ultimately, it will give us more insights in the human representation of the world and the reasoning that goes with it.

The first capability for visual cognition must be visual re-cognition, identifying visual categories based on images that have been seen before. But as natural as visual recognition comes to humans, as fickle it has been proven to reproduce in computers. In this thesis we address the problem of the automatic understanding of images; we aim to take a leap forward in the part of computer vision which focuses on the automatic identification of the contents of an image.

But what is it we want to extract from an image? What is most useful? Our answer is that, eventually, images tell stories. When somebody goes on holiday and shoots pictures, it is to tell a story about the vacation, to record the sights and events that transpired. Be it walking on the Forum Romanum or the joy of having a feast in Spain. When a journalist shoots pictures, it is meant to give an objective, visual account of reality. Sometimes the photographs portray a war. Other times they tell stories about the life of a celebrity. An advertisement tells a deliberate story designed to leave an impression. The photo-ad may highlight how fast a blender blends or may convey the feeling of belonging when drinking coffee. In general, the story of a picture is based on the expectations in relation to the content of the image. In this thesis we focus on the content alone.

From a machine vision perspective, the stories an image tell range from simple to complex. The simplest story is naming a single scene, such as a beach or a mountain range. This is followed by naming a single object such as bicycle, or cat. A more complex story is to describe multiple objects such as horse and person or kid and ball. Spatial relations are necessary for distinguishing between person inside a car and person in front of a car. Finally, actions and interactions describe the dynamics in a scene leading to the inclusion of verbs, enabling stories like person jumping out of an aeroplane, and man running from a dog. Research in computer vision has made much progress in identifying single objects or generating single
object stories. In this thesis we provide an analysis of the recent, vast achievements in this area. We aim to contribute to an understanding of the notion of an object in computer vision. Furthermore, we aim to improve semantic labelling performance and to identify multiple objects as the next step towards generating the full narrative of an image.

As said, much progress has been made in identifying single objects within images. The reason of the progress, however, is striking. Essentially, the model of image analysis was impoverished by removing the notion of the object location and by replacing it by the use of local image details scattered over the image. This enabled the use of powerful, new machine learning techniques. Coupled with large amounts of data that became available, this led for the first time to usable content-based image search engines. However, as computational demands of such systems were high, our first research question deals with computational efficiency:

- Can we accelerate Visual Concept Classification?

We approach this question of computational efficiency in Chapter 2, which was published in IEEE Transactions on Multimedia [99]. To this end, we start with the object classification system to which we contributed and which in the 2008 benchmarks of TRECVID and Pascal VOC led to the best performance scores [86,90]. Then we review a variety of existing and novel techniques to accelerate object classification and show the trade-off between computational efficiency and accuracy.

The acceleration of image classification achieved by this chapter enables content-based image analysis on large-scale datasets such as Flickr. Importantly for this thesis, it also enables a much more thorough analysis. Specifically, it enables extracting local image details at every point or pixel in the image, necessary for addressing the subsequent research questions.

In general, the replacement of the object location with strong machine learning techniques and more data resulted in significant performance improvements, yet insights into why this happened remained behind: Which parts of the image provide evidence for the existence of an object? Only the object itself? The whole image including the context? In short:

- What is the Visual Extent of an Object?

This research question is addressed in Chapter 3, which was accepted by the International Journal of Computer Vision [96]. While it is widely acknowledged that the visual extent of an object reaches beyond the object itself, this chapter investigates to what degree taking this visual extent into account affects classification performance.

We perform our analysis from two angles. (a) Not knowing the object location, we determine where in the image the support for object classification resides. This perspective enables us to visualize and analyse how a state-of-the-art image analysis system classifies images. (b) Assuming that the object location is known, we evaluate the relative potential of the object and its surround, and of the object border and object interior.

The analysis in this chapter yields detailed insights into which parts of the image are useful for the identification of objects. And while one of the reasons for the success of current image analysis systems was the absence of explicit
object locations, results of this chapter suggest that it is beneficial to bring the object location back.

For dealing with object locations, there are two dominant directions in computer vision. The first is segmentation, which is usually defined as finding the unique outlines of all objects within an image, resulting in few but high quality image regions. The original idea was that only after the object is located, it can be identified. However, segmentation is currently viewed as fundamentally impossible as the number of possible image conditions is vast: Think of shadows and the difference in lighting indoors and outdoors. Furthermore, without identification, why would a person’s head belong to his body? The other dominant direction deals with the localisation and identification of objects. Most research in this direction ignores segmentation and relies on machine learning alone. First the appearance of the target object is learned. Then for a new image, huge numbers of locations are probed for its presence, somewhat bounded by a variety of scales, aspect ratio, and search grid. However, such an exhaustive search strategy seems highly unsatisfactory. Instead, we aim for a structured sampling that builds upon the strengths of both these directions. Like exhaustive search we want to generate many object locations in order to yield some good ones. Like segmentation we want to use the structure of the image to guide our sampling. This leads to the research question:

- Can we perform Structured Sampling for Object Recognition?

This question is addressed in Chapter 4, which is an extension of our ICCV paper [102]. Our structured sampling combines multiple complementary data-driven partitionings of the image to deal with as many image conditions as possible. Specifically, we use all regions of a hierarchical partitioning to capture all scales within the image. Furthermore, we use a variety of colour spaces with different invariance properties and a variety of different segmentation criteria such as colour and size. This results in a structured sampling strategy that is class independent, and which theoretically generates locations for objects such as car and cat, for concepts like grass and road, and even for object parts and object collections. In this chapter, however, we evaluate our structured sampling on its ability to find complete objects. In addition, we test our structured sampling on the exact localisation and identification of an object, where the reduced number of locations enables the use of more expensive and powerful features than possible in an exhaustive search strategy. We compare with the state-of-the-art on the Pascal VOC 2010 detection challenge and show significant improvements for several object classes.

In Chapter 5, the final chapter of this thesis, we want to take the analysis of image content to the next level. Performance of single-object classification is acceptable, but

- Can we do Multi-Object Classification as the next step towards generating Stories?

The joint prediction of multiple objects is only viable by combining individual object classifiers: the exponential combinatorics resulting from recognising $n$ objects from $c$ classes make it fundamentally impossible to learn classifiers for all combinations. To be able to combine different object classifier, they should
be largely independent. Therefore it is important that they use different features within the image, hence the identification should be based on localised features.

Importantly, when images contain more objects they get more cluttered leading to occlusions. Furthermore, objects may interact, leading to pose changes. An important insight is therefore not to use the complete object, but rather to focus on the most discriminative parts of the object. This realisation leads to the introduction of the “most telling window” of an image, which can be an object part, complete object, or an object collection, depending on the object class and image composition. We evaluate our strategy on single object classification and on the joint classification of two and tree objects.

The final result of this thesis is a system which jointly classifies multiple objects. To appreciate why this is a step towards the full narrative of an image, consider: What story can three objects tell about a photo?
REAL-TIME VISUAL CONCEPT CLASSIFICATION

As datasets grow increasingly large in content based image and video retrieval, computational efficiency of concept classification is important. This chapter reviews techniques to accelerate concept classification, where we show the trade-off between computational efficiency and accuracy. As a basis we use the Bag-of-Words algorithm that in the 2008 benchmarks of TRECVID and PASCAL lead to the best performance scores. We divide the evaluation in three steps: (1) Descriptor Extraction, where we evaluate SIFT, SURF, DAISY, and Semantic Textons. (2) Visual Word Assignment, where we compare a k-means visual vocabulary with a Random Forest and evaluate subsampling, dimension reduction with PCA, and division strategies of the Spatial Pyramid. (3) Classification, where we evaluate the $\chi^2$, RBF, and Fast Histogram Intersection kernel for the SVM. Apart from the evaluation, we accelerate the calculation of densely sampled SIFT and SURF, accelerate nearest neighbour assignment, and improve accuracy of the Histogram Intersection kernel. We conclude by discussing whether further acceleration of the Bag-of-Words pipeline is possible.

Our results lead to a 7-fold speed increase without accuracy loss, and a 70-fold speed increase with 3% accuracy loss. The latter system does classification in real-time, which opens up new applications for automatic concept classification. For example, this system permits 5 standard desktop PCs to automatically tag for 20 classes all images that are currently uploaded to Flickr.

2.1 INTRODUCTION

Over the last decade there has been an explosive growth of available multimedia on the internet. Good examples are the photo sharing website Flickr, hosting billions of images with thousands of uploads each minute, and the video sharing website YouTube, hosting millions of videos with hours of video uploaded each minute. The advent of such collections has sparked research on image and video retrieval within these collections, see Snoek and Worring for a recent overview [87]. One active line of research is on using only the visual contents for concept-based search tasks.

Within this domain the Bag-of-Words method [18,82] has proven to be the most efficient strategy as a generic classification scheme for individual concepts. This is proven by their top performance in various major benchmarks over the past few years such as the TRECVID high-level feature extraction task (which uses video) [83] and the Pascal VOC Challenge (which uses images) [24]. In these benchmarks, concept detectors are able to detect classes such as chair, cat, car, boat, building, meeting, and sports with varying degrees of success [24,83].
But while Bag of Words systems give superior classification results, they are computationally very expensive. For example, a complete classification run for the TRECVID high-level feature extraction task requires processing over 40,000 video frames (1 frame per shot of 180 hours of video). The state-of-the-art system used in [86] will take days to complete on a computer cluster. Hence, for large scale image retrieval with Bag-of-Words there is the need to speed up this method.

This chapter presents a comprehensive evaluation of various fast Bag-of-Words components in terms of both computational efficiency and retrieval performance, which is our main contribution. The evaluation shows an increase of accuracy for Random Forests [62] by combining them with Principal Component Analysis, and an increase in computational efficiency by determining relevant image divisions for the Spatial Pyramid [51]. Additionally, we present several improvements upon the components under consideration: We provide a modified way to speed up the calculation of densely sampled SIFT [53] descriptors. Similarly, we turn SURF [6] into a faster, densely sampled descriptor. We accelerate nearest neighbour assignment. Furthermore, we increase accuracy of the Histogram Intersection based Support Vector Machine by balancing visual word frequencies. Finally, next to the experimental evaluation, we provide a theoretical discussion on computational efficiency of the Bag-of-Words method.

This chapter is an extension of [97]. This document is structured as follows: First we give a short overview of the Bag-of-Words framework in section 2.2. Section 2.3 discusses related work. In section 2.4 we discuss various strategies for accelerating the Bag-of-Words pipeline. Section 2.5 gives an overview of our experimental setup. Section 2.6 presents and discusses our results. A discussion about the theoretical computational complexity of the Bag-of-Words pipeline is given in section 2.7. Finally our conclusions are given in section 2.8.

### 2.2 Bag-of-Words

The Bag-of-Words method is derived from text retrieval where documents are represented as the frequencies of their words. This word frequency count is used for retrieving or classifying documents. Similarly, in the Bag-of-Words method one first samples local regions from an image which are then converted to visual words. The visual word frequencies of an image are then used in subsequent classification. Figure 1 shows a schematic overview of this process, with the creation of a vocabulary of visual words on the left side, and conversion of an image to a visual word frequency histogram on the right.

The visual vocabulary within a Bag-of-Words framework is learned from a training set. First, small image regions are sampled from all images in the training set. From each region a descriptor is extracted, usually SIFT [53] which is a histogram of oriented gradients. Typically, the visual vocabulary is learned by using an unsupervised clustering algorithm such as k-means. The resulting cluster centres define the visual vocabulary in a nearest neighbour sense by partitioning the descriptor space. Each resulting partition represents a visual word. A fast alternative method evaluated in this chapter is to partition the descriptor space using binary decision trees. The size of the visual vocabulary is user defined but typically ranges in the thousands. The visual vocabulary is learned once and remains fixed afterwards.
To create a visual word frequency histogram, small regions are selected from an image from which descriptors are calculated. Each descriptor is then mapped to a visual word of the vocabulary according to the partitioning of the descriptor space. This results in a visual word frequency histogram which is used in subsequent classification.

This chapter presents an evaluation of the on-line parts of a Bag-of-Words image classification system. This excludes the creation of the visual vocabulary and learning the classifier. A typical Bag-of-Words image classification pipeline which forms the basis of several state-of-the-art image/video retrieval systems [58,86,90] is given in figure 2. The pipeline can be decomposed into three different components: Descriptor Extraction, Word Assignment, and Classification. In this pipeline,
the Descriptor Extraction phase extracts SIFT descriptors \cite{53}. The visual vocabulary is created using k-means and Word Assignment is done using nearest neighbour assignment. A weak form of spatial information is incorporated using the spatial pyramid \cite{51}. Classification is done using a Support Vector Machine with a $\chi^2$-kernel.

**Figure 2**: An example scheme for classification using the Bag of Words method used by good image/video retrieval systems \cite{58,86,90}. This particular pipeline takes 2063 ms per image and has an accuracy of 0.476 MAP. State of the art results can be obtained by using dense sampling on multiple scales and by combining a variety of (colour) descriptors in the classification phase.

### 2.3 Related Work

We divide the related work into the three components of the Bag-of-Words pipeline: Descriptor Extraction, Word Assignment, and Classification.

#### 2.3.1 Descriptor Extraction

Descriptor extraction begins with selecting small regions from the image. Lowe \cite{53} introduced a region or interest point detector based on the difference of Gaussian (DOG). Bay et al. \cite{6} improved computational efficiency with a factor six by proposing a Fast-Hessian interest point detector. Šochman and Matas \cite{109} accelerated interest point detection by emulating them using their WaldBoost algorithm. However, in the context of object recognition, Jurie and Triggs \cite{47} showed that sampling many patches on a regular dense grid outperforms the use of interest points. This chapter therefore uses the dense sampling method which eliminates the computational costs of detecting interest points altogether.

From the small, selected regions in the image one extracts descriptors. Mikolajczyk and Schmid \cite{61} compared various descriptors and found SIFT \cite{53} or SIFT-like descriptors to be the best in the context of image matching under various image transformations. Mikolajczyk et al. \cite{59} and Zhang et al. \cite{115} showed that SIFT-like descriptors are also superior for object detection. In this chapter we introduce a computationally efficient SIFT-variant which gives similar accuracy to the original SIFT.

The SIFT descriptor consists of oriented gradient responses that are summed over subregions. The summation becomes computationally expensive when lots
of descriptors are extracted. Grabner et al. [40] proposed to obtain the summation efficiently by using integral images enabling them to calculate the summation of any subregion using only its four corners. The resulting descriptor deviates from SIFT [53] in omitting two weighting schemes: First of all, SIFT uses a linear interpolation between subregions which makes it robust against small changes in position. Second, SIFT has a Gaussian weighting scheme around the origin emphasizing the importance of the detected interest point. [40] report a factor eight speed increase while matching performance decreases slightly. Unlike [40], we do not use interest points and we exploit the regularity of the dense sampling to create fast summations over subregions.

Tola et al. [92] propose a different method to speed up the summations over subregions. They use a Gaussian convolution on the pixel responses after which each pixel represents a weighted sum of responses over its neighbourhood. They then create their DAISY descriptor by taking circularly arranged responses. As [92] can be seen as a form of dense sampling, we include DAISY in our evaluation.

SIFT is based on relatively expensive oriented gradient responses. This is addressed by Bay et al. [6] with SURF, a spatial descriptor similar to SIFT based on Haar wavelet responses. Haar wavelets are cheaper to compute than the Gaussian derivatives of SIFT. We include the SURF descriptor in our experiments and examine if their results on object recognition extend to larger and more difficult datasets. Additionally, as with SIFT, we exploit the spatial nature of SURF in combination with the dense sampling strategy to speed up calculation.

Shotton et al. [79] propose to omit the extraction of descriptors altogether in the context of pixel-wise classification and segmentation. Instead, they use raw-pixel values of a small region directly. For the subsequent step of Word Assignment, they use a Random Forest, a set of supervised decision trees. We include their Semantic Textons in our evaluation.

2.3.2 Visual Word Assignment

Descriptors are assigned to a visual vocabulary by what we call Word Assignment. Commonly, large visual vocabularies are created with unsupervised k-means clustering which gives good performance (e.g. [58, 101, 115]). Typically each descriptor is assigned to a single word of this visual vocabulary using nearest neighbour e.g. [101, 113, 115]. However, it has been shown that assigning each descriptor to multiple visual words by using a so-called soft-assignment is beneficial to performance [34, 72] at the expense of extra calculation time: In soft assignment one needs to obtain the closest n visual words and calculate a posterior probability over these. As this chapter aims for acceleration of visual concept classification we exclude the use of soft-assignment in our evaluation.

Computation time for the Word Assignment is dependent on three factors: the number of descriptors extracted per image, the dimensionality of the descriptors, and the size of the visual vocabulary. Nowak et al. [66] showed that using more descriptors is better. We evaluate the speed-performance trade-off for using fewer descriptors. Mikolajczyk and Schmid [61] apply Principal Component Analysis (PCA) on their SIFT-variant called GLOH to ensure it has the same dimensionality as SIFT. We apply PCA to our descriptors to reduce their dimensionality in order to increase computation speed.
Jiang et al. [46] and Moosmann et al. [62] experimented with the size of the visual vocabulary and concluded that large vocabularies are generally beneficial for large datasets. We will fix the size of the visual vocabulary to 4096 visual words as this gives good results on both the TRECVID and Pascal VOC dataset [58,86,90]. Jurie and Triggs [47] and Wang et al. [112] propose methods to create compact yet discriminative vocabularies. These methods are orthogonal to the methods described in this chapter.

In order to make word assignment more efficient, several tree-based assignment algorithms were proposed [60,63,65]. Tree-based word assignment algorithms allow for a logarithmic rather than a linear assignment time in terms of the number of visual words. The most interesting one is the work on supervised random forests by Moosmann et al. [63]; apart from a computational advantage the chapter also reports improvements over regular k-means in terms of performance on their four-class datasets. As the creation of the trees is supervised, the relatively small number of classes they use might have positively influenced their results. This chapter considers how their method extends to more classes.

As mentioned earlier, Shotton et al. [79] use Random Forests directly on the pixel values of image regions and show its effectiveness in image segmentation. Their Random Forests differ from [63] in that each decision node in the tree works on multiple values of the descriptor instead of one. In most of our experiments we will use the Random Forests which work on a single dimension as these are faster and preliminary experiments showed no significant benefits for decisions on multiple dimensions. Only when evaluating their Semantic Textons we use decision trees as defined in [79].

Lazebnik et al. [51] proposed the spatial pyramid, introducing a weak form of spatial information by increasingly subdividing the image and obtain a visual word frequency histogram for each region separately. This results in a 5-10% performance increase at a very limited computational word assignment cost. He et al. [43] expanded on this work by learning weights for each of the resulting subregions. However, classification time is dependent on the size of the word frequency histogram and hence on the number of subregions. In this chapter we consider various image division strategies to minimize the number of subregions and speed up classification time.

### 2.3.3 Classification

Support Vector Machines (SVMs) are a very popular classifier due to its robustness against large feature vectors and sparse data. They are successfully used in Bag of Words methods. The choice of SVM-kernel has a large impact on performance. Both Zhang et al. [115] and Jiang et al. [46] determined that the $\chi^2$-kernel gives the best accuracy. We will follow their experiments but next to retrieval performance we also focus on computational efficiency.

Maji et al. [54] proposed an efficient classification scheme for SVMs when using histogram intersection kernels. In this chapter we will use their implementation for the histogram intersection kernel.
2.3.4 Graphics Processing Unit

Orthogonal to methodological improvements, researchers are looking to Graphics Processing Units (GPUs) to speed up Bag-of-Words. While hardware optimizations fall outside the scope of this chapter, the methods we evaluate in this chapter can all be implemented on a GPU: To calculate SIFT and SURF one needs matrix multiplications and (recursive) Gaussian filters, which can be found in the standard GPU software development kit CUDA. Sharp proposes an algorithm for Random Forests on the GPU. Finally, Catanzaro et al. show how to implement a Support Vector Machine on the GPU.

2.4 ACCELERATED BAG-OF-WORDS

In this section we describe the simple way of calculating densely sampled descriptors. Furthermore we discuss a fast way to calculate nearest neighbour assignment. Then we will discuss the Random Forests of Moosmann et al. and the fast Histogram Intersection SVM classifier of Maji et al.

2.4.1 Descriptor Extraction

Accelerated Dense Descriptor Extraction

Both SIFT and SURF are spatial descriptors: each is constructed of $4 \times 4$ subregions which in turn are described by the summation of pixel-wise responses over an area. In the case of SIFT the responses are oriented gradients calculated using image convolutions, for SURF these are Haar wavelet responses calculated using simple summations and subtractions.

First we observe that if the dense sampling rate is the same as the size of a subregion we can reuse these subregions for the other descriptors. For the original $4 \times 4$ SURF and SIFT descriptors this means a factor 16 speed improvement for the summations over the pixel responses.

The original SIFT uses a Gaussian weighting over the complete image patch, attributing greater importance to the values in the middle of the descriptor. This step obstructs the reuse of subregions. But the original SIFT was made for use in conjunction with interest points where the middle of the descriptor is more important by design. However, the dense sampling strategy creates arbitrary image patches hence all parts of an image patch seem equally important. As we use dense sampling, we omit the Gaussian weighting.

To sum the responses within each subregion we use two matrix multiplications: One to sum in the row direction and the other to sum over the column direction. Consider the pixel-wise responses $R$ from an image. If we want to sum the responses over subregions of $3 \times 3$ pixels we employ a matrix multiplication $ARB$, where $A$ sums over elements in the row direction and has the form of

$$
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1
\end{pmatrix}.
$$
Matrix $B$ sums over the column direction and is similar to $\Lambda^T$ but has a size adapted to $\mathbf{R}$.

For robustness against small shifts in position of the descriptor, SIFT uses a linear weighting to divide responses over neighbouring subregions. We do a linear weighting by modifying $A$ (and likewise $B$) to

$$
\begin{pmatrix}
1 & 2/3 & 1/3 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 1/3 & 2/3 & 1 & 2/3 & 1 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 1/3 & 2/3 & 1 & 2/3 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{pmatrix},
$$

where the top left entry is 1 rather than $1/3$ due to normalization at the boundary. The resulting descriptor only differs from the original SIFT in omitting the Gaussian weighting over the whole descriptor. It differs from [40] in that they omit also the linear interpolation between subregions while we keep this.

Performing the summation over the regions with matrix multiplications allows the use of specialized matrix multiplication libraries in combination with sparse matrices. The resulting speed-up is up to a factor 2 compared to a naive C++ implementation.

Once all values of subregions have been extracted we can construct any spatial form of these subregions. Traditionally the SIFT (and SURF) descriptor is composed of $4 \times 4$ subregions. In this chapter we also evaluate $2 \times 2$ descriptors which have less dimensions.

2.4.2 Visual Word Assignment

Nearest Neighbour Assignment

Each descriptor from an image is projected onto a predefined visual vocabulary (obtained by k-means) using nearest neighbour assignment. It is common practice to normalize both SIFT and SURF to unit vectors to make them robust against illumination changes. As all descriptors are unit length, nearest neighbour assignment using the Euclidean Distances is equal to using angles between vectors. By using the inner product to calculate these angles, we obtain a speed-up of 43% over using Euclidean Distances in the word assignment phase.

Random Forest

A random forest is a collection of binary decision trees whose combination leads to fast yet accurate classification performance. This chapter evaluates the decision trees in the word assignment step. Unlike k-means visual vocabularies, the decision trees are created in a supervised way. Following Moosmann et al. [62], we use [35] for the construction of the trees. For each tree we start with 250,000 labelled descriptors $D$ from our training set, where the labels are taken from the global image annotations (i.e. annotations at image level). Learning is done recursively. At each node $n$, $s$ random splits are proposed by choosing a random dimension of the descriptors and a random threshold $t$. This splits the set
of descriptors \( D_n \) at node \( n \) in \( D_a \) and \( D_b \). Each split is evaluated using the information gain \( \Delta E \), defined as [79]

\[
\Delta E = -\frac{|D_a|}{|D_n|} E(D_a) - \frac{|D_b|}{|D_n|} E(D_b),
\]

where \( E(D_a) \) is the Shannon Entropy of the class labels of \( D_a \). The split with the highest information gain is then adopted. Training continues with \( D_a \) and \( D_b \) and stops if a specific depth is reached. The end result is a binary decision tree.

Word assignment using a random forest is done by obtaining the visual word frequency histograms for all trees and simply concatenating these histograms into one vector.

Random forests are interesting from a computational point of view in two ways: First of all, the binary nature of the decision trees result in a word assignment time which is logarithmic in the number of visual words, whereas this is linear for nearest neighbour assignment. Furthermore, as at each decision node in the tree only a single dimension of the descriptor is compared to a threshold, the visual word assignment time for random forests is independent of the dimensionality of the descriptors while for nearest neighbour assignment this dependency is linear.

2.4.3 Classification

The precomputed kernel

The classification function for a SVM can be written as [8]

\[
h(x) = b + \sum_{j=1}^{m} \alpha_j t_j K(x,z_j),
\]

where \( x = \{x_1, \ldots, x_k\} \) is the vector to be classified, \( z_j = \{z_{1j}, \ldots, z_{kj}\} \) is the \( j \)-th support vector, \( \alpha_j \) is its positive support vector weight, \( t_j \in \{+1, -1\} \) is the label of the support vector, \( m \) is the number of support vectors, and \( K(\cdot, \cdot) \) is a kernel function. The time complexity of this function is dependent on its kernel function. For the common distance-based kernel functions that we use the time complexity is \( O(km) \). Suppose that the average number of support vectors for a single class is \( \bar{m} \), then classification for all classes is of complexity \( O(kmc) \).

Alternatively, as support vectors are selected from the training set one can choose to calculate the kernel function between the whole training set and a test sample and select later for each class which kernel entries to use. If we define \( q \) as the size of the training set, classification complexity for all classes becomes \( O(kq) \). This strategy is computationally advantageous if \( q < \bar{m}c \). For our datasets this is indeed the case: all datasets have 15 or more classes and each class takes 10-25% of the training vectors as support vectors.

In our experiments we calculate the kernel function between the whole training set and the whole test set at once, which is called using a pre-computed kernel. Compared to calculating the distance separately for each test sample, the pre-computed kernel reduces overhead and may allow a speed-up when using efficient matrix multiplications but its memory requirements are much higher.
Efficient Support Vector Machine classification for Histogram Intersection kernels

The classification function for a Support Vector Machine in equation 2.2 has time complexity $O(km)$. Recently, Maji et al. [54] showed that the classification function for the histogram intersection kernel

$$K(x, z) = \sum_{i=1}^{k} \min(x_i, z_i)$$

(2.3)

can be rewritten to give a time complexity of $O(k \log m)$:

$$h(x) = b + \sum_{j=1}^{m} \alpha_j t_j K(x, z_j)$$

(2.4)

$$= b + \sum_{i=1}^{k} \sum_{j=1}^{m} \alpha_j t_j \min(x_i, z_{ij})$$

(2.5)

Now for a fixed value of $i$ let $	ilde{z}_{ij}$ denote the sorted values of $z_{ij}$ in ascending order with corresponding weights $\tilde{\alpha}_j$ and labels $\tilde{t}_j$. Let $r$ be the largest integer for which $\tilde{z}_{ir} \leq x_i$. This allows rewriting equation 2.5 as

$$h(x) = b + \sum_{i=1}^{k} \left\{ \sum_{j=1}^{r} \tilde{\alpha}_j \tilde{t}_j \tilde{z}_{ir} + x_i \sum_{j=r+1}^{m} \tilde{\alpha}_j \tilde{t}_j \right\}.$$  

(2.6)

For each $i$ we can precompute for all $r$

$$\sum_{j=1}^{r} \tilde{\alpha}_j \tilde{t}_j \tilde{z}_{ir}$$  

(2.7)

and

$$\sum_{j=r+1}^{m} \tilde{\alpha}_j \tilde{t}_j$$  

(2.8)

as they have become independent of $x_i$. By exploiting the piecewise linear form of the resulting decision function the memory requirement becomes twice that of the normal SVM implementation [54]. Calculating the classifier output now amounts to finding for each dimension $i$ the correct $r$ which can be done logarithmically in the number of support vectors $m$. The total run-time complexity thus becomes $O(k \log m)$ instead of $O(km)$, which is a significant speed increase.

Maji et al. [54] also found that the piece-wise linear function between the brackets of equation 2.6 can reasonably well be approximated by a piece-wise linear function with uniform spacing between its segments (rather than a spacing which is determined by the sorted values $\tilde{z}_{ij}$ of the support vectors). The uniform spacing allows for a direct mapping of $x_i$ to a corresponding $r$, removing the dependence on the number of support vectors $m$ altogether. This leads to an even faster classification complexity of order $O(k)$ with negligible loss of accuracy. We evaluate both the exact and approximate Histogram Intersection SVM of [54].

20
2.5 EXPERIMENTAL SETUP

We compare various alternative Bag-of-Words components with respect to a basic Bag-of-Words pipeline. In our comparison we consider both retrieval performance and computational efficiency. We divide our experiments into the three stages of the Bag-of-Words pipeline:

1. **Descriptors.** The features which describe the extracted local image patches.

2. **Word Assignment.** The assignment of these descriptors to a word in the visual vocabulary, resulting in a visual word frequency histogram. The Spatial Pyramid is applied in the word assignment phase.

3. **Classification.** The classification of these visual word frequency histograms.

For measuring retrieval performance we report the standard measure for that dataset. This is either the percentage of correctly classified examples or the Mean Average Precision (MAP) over all classes. The Average Precision for a single class is defined as

$$\text{AP}_c = \frac{1}{m} \sum_{i=1}^{n} \frac{f_c(x_i)}{i},$$

where: $n$ is the number of images, $m$ is the number of images of class $c$, $x_i$ is the $i$-th image in the ranked list $X = \{x_1, \ldots, x_n\}$. Finally, $f_c$ is a function which returns the number of images of class $c$ in the first $i$ images if $x_i$ is of class $c$, and 0 otherwise.

Computational efficiency is measured in milliseconds per image, where the measurement is an average over all images in the test set. Classification time for an image is reported for the classification of all classes in the dataset. The efficiency measurements are done on a mainstream processor (a single core of a 3.16 Ghz Intel Core Duo E8500 processor).

2.5.1 **Datasets**

We perform our experiments on three different datasets. The **Pascal VOC 2007** dataset consists of 9963 images divided into a predefined training and test set of respectively 5011 and 4952 images. The general image size is $300 \times 500$ pixels. The dataset consists of 20 object classes: aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining-table, dog, horse, motorbike, person, potted-plant, sheep, sofa, train, and TV/monitor. Some images contain multiple objects of potentially different classes. Retrieval performance is measured using the Mean Average Precision (MAP) over all object classes.

The **MediaMill Challenge** dataset [85] consists of 85 hours of video from the TRECVID 2005 development set which is segmented into shots with corresponding keyframes. As is common, we only use these keyframes, discarding temporal information (unlike e.g. [4]). The set is divided into a predefined training and test set which respectively contain 30993 and 12914 keyframes. All keyframes are $240 \times 352$ pixels and contain compression artefacts. There are 101 concept categories ranging from object categories (e.g. bird, bus), to natural scenes (e.g. mountain, sky), semantic concepts (e.g. entertainment, meeting), and actions (e.g. fire weapon, people marching). Keyframes may contain zero, one, or multiple concepts.
The Fifteen Scene Categories dataset [51] expands upon earlier scene category databases [26, 68]. For each scene category there are 200-400 images with an average size of 300 × 250 pixels. We randomly divide the set into a training and test set of respectively 2245 and 2240 images. The scene categories for this dataset are mutually exclusive and contain categories like bedroom, industrial, and forest.

In our experiments we measured very similar speed improvements for all three datasets and also similar trends in accuracy. Therefore we will discuss only the results for the Pascal VOC 2007 dataset in detail. We will return to the other datasets in the conclusions.

2.5.2 Baseline

Our baseline Bag-of-Words system is modelled after the best systems of the Pascal VOC challenge 2007 and 2008 [58, 90], where we exclude the spatial pyramid for its computational demands during training. However, as the spatial pyramid seems intuitively equally powerful for all descriptors and word assignment methods we do not have to include it for most of our experiments.

We use the intensity-based SIFT descriptor extracted by our fast Dense Sampling strategy, termed DIFT from now on. We sample subregions of 6 by 6 pixels each 6-th pixel and use the original spatial configuration of 4 × 4 subregions.

Our visual vocabulary consists of 4096 words created using k-means clustering. This vocabulary size is kept constant throughout our experiments. New descriptors are projected to the visual vocabulary using nearest neighbour assignment through inner products. Classification of the resulting word frequency histograms is done using a SVM with a χ²-kernel, where we make use of the precomputed kernel which takes up most of the classification time.

The resulting Bag-of-Words pipeline for the baseline experiment is presented in figure 3. Note that the pre-assignment step is currently empty but will be used by two of our experiments.

Subsequent experiments will always affect a single element of this baseline pipeline.

![Figure 3](image-url)

**Figure 3**: The baseline Bag of Words pipeline as used in our experiments. The total computation time is 786 milliseconds per image to classify all 20 classes of the Pascal VOC dataset. Its Mean Average Precision is 0.447.
2.5.3 Descriptors

In this experiment we compare various fast alternatives to the SIFT descriptor, all extracted using dense sampling on a regular grid. We focus on computational efficiency for both the extraction of the descriptors and the visual word assignment because the dimensionality of the descriptors influences assignment time.

We compare four SIFT variants and three SURF variants. We compare our implementation of the original SIFT $4 \times 4$ descriptor which includes the Gaussian weighting with our DIFT $4 \times 4$ descriptor which omits this weighting. Both descriptors have 128 dimensions. To decrease the dimensionality and hence word assignment time we construct two DIFT versions out of four subregions resulting in 32 dimensions. DIFT $2 \times 2$ uses the same subregions as DIFT $4 \times 4$. DIFT $2 \times 2^*$ uses the exact same pixel values as DIFT $4 \times 4$. This arguably results in a more fair comparison but is less compatible with our descriptor extraction method: we need to extract the features four times to achieve the same dense sampling rate (we need four different sets of subregions). Notice that besides a dimensionality reduction, fewer subregions also result in less redundancy: neighbouring subregions are likely to describe similar statistics, so more subregions in a descriptor means a higher probability of redundancy. A schematic overview of the various SIFT variants is given in figure 4.

![Figure 4: A schematic overview of the various SIFT variants used in this chapter. The circle in figure 4a represents a Gaussian weighting centred around the origin of the descriptor. In this picture most subregions are 2 by 2 pixels while in our actual experiments these subregions are 6 by 6 pixels.](image)

We also use three variants of DURF which can be seen as the counterparts of DIFT; the responses within the same subregions are used for summation but DURF uses Haar wavelet responses rather than oriented gradients. As the Haar...
wavelets are calculated in only the horizontal and vertical directions and not in the diagonal directions, the resulting dimensionality of the descriptors is half that of DIFT. The DURF variants are termed DURF $4 \times 4$, DURF $2 \times 2$, and DURF $2 \times 2^*$, and consist of respectively 64, 16, and 16, dimensions.

We also evaluate the DAISY descriptor [92]. The DAISY descriptor can be seen as a variant of SIFT, where its subregions are circular and summed using a Gaussian weighting rather than square and summed using a linear weighting. This is visualised in figure 5. In total there are 25 subregions of 8 oriented gradients, resulting in a 200-dimensional vector. We use the implementation provided by [92].

![Figure 5: Visualisation of pixel-wise summation as done by the DAISY descriptor. Each circle describes a Gaussian summation over the pixel-wise oriented gradient responses.](image)

Finally, we evaluate Semantic Textons [79]. Semantic Textons are defined to work with a Random Forest and are not suited for nearest neighbour assignment. For fair comparison, we use Semantic Textons on the intensity values only, just like the other descriptors. A concise overview of the dimensionality and region sizes of the descriptors is given in table 1.

### 2.5.4 Word Assignment

The word assignment time when using nearest neighbour assignment depends on three factors: the size of the visual vocabulary, the number of descriptors
2.5 EXPERIMENTAL SETUP

generated per image, and the dimensionality of the descriptors. In this chapter we discuss experiments on the number and dimensionality of the descriptors. We refrain from experimenting with the size of the visual vocabulary, as this has been exhaustively studied in [46,62]. Instead we fix the vocabulary size to 4096, which gives good results on the Pascal VOC dataset [58,90]. Using a larger vocabulary implies an increased classification time for the SVM.

We decrease the number of descriptors per image by a random sub-sampling strategy.

The size of the descriptors is reduced by Principal Component Analysis. We only use the rotation component of PCA. The translation component is expensive to calculate and does not influence distances and hence does not influence the resulting word frequency histograms.

We also consider the Random Forest [62] described in section 2.4.2 as a fast alternative to k-means and nearest neighbour assignment. As a Random Forest of 4 trees gives good results for [62], our forests are made of 4 trees of depth 10 resulting in the appropriate vocabulary size of 4096 visual words.

Finally we consider various image divisions for the spatial pyramid of Lazebnik et al. [51]. The original spatial pyramid uses the same number of horizontal and vertical divisions. We consider combinations of 1-4 horizontal and 1-4 vertical divisions. The aim is to find the minimum number of resulting sub-regions with high retrieval efficiency, as classification time is linear in the number of these sub-regions.

2.5.5 Classification

The choice for Support Vector Machine kernels influences both retrieval performance and classification speed. For a word frequency histogram \( x = \{x_1, \ldots, x_k\} \) and a support vector \( z = \{z_1, \ldots, z_k\} \), the SVM kernel is defined as \( K(x, z) \). In our experiments we compare three different kernels:

1. Histogram intersection kernel, defined as

   \[
   K(x, z) = \sum_{i=1}^{k} \min(x_i, z_i).
   \]  

2. Radial Basis Function, defined as

   \[
   K(x, z) = e^{-\gamma \text{D}_{\text{euclid}}},
   \]  

   where

   \[
   \text{D}_{\text{euclid}} = \sum_{i=1}^{k} (x_i - z_i)^2
   \]  

   and \( \gamma \) is a normalization factor which in our experiments we set to \( 1/\text{D}_{\text{euclid}} \) as in [58].
3. $\chi^2$-kernel, defined as

$$K(x, z) = e^{-\gamma D_{\text{chi}}}$$

where

$$D_{\text{chi}} = \sum_{i=1}^{k} \frac{(x_i - z_i)^2}{x_i + z_i}$$

and $\gamma$ is a normalization factor which in our experiments we set to $1/D_{\text{chi}}$ as in [58].

Our implementation for pre-computing these kernels exploits the sparseness of the word frequency histograms for computational efficiency. Besides the pre-computed Histogram Intersection kernel we use the fast implementation of [54].

Finally, we revisit the spatial pyramid using the classification method which proves to be the fastest.

2.5.6 Implementation Details

For the DIFT and DURF descriptors we sum responses over subregions of 6 by 6 pixels. The horizontal and vertical responses for SIFT are calculated using a Gaussian derivative filter while the diagonal responses are calculated using a fast anisotropic Gaussian derivative filter [37], all using a sigma of 1. For DURF we calculate each 2nd pixel a Haar Wavelet response of 4 by 4 pixels. Notice that calculating responses each 2nd pixel means that the subregions of 6 by 6 pixels contains only 9 Haar responses instead of the 36 Gradient responses of DIFT. For DAISY we set the radius of the descriptor to 12 pixels resulting in comparable image-regions from which the descriptor is calculated. We use the default settings for the other parameters [92]. For the Semantic Textons we take regions of 24 by 24 pixels where we normalize each region to unit length to be invariant against intensity changes. All descriptors in this chapter are sampled at each 6-th pixel. For each descriptor type we generate about 4500 descriptors per image for the Pascal VOC dataset.

One visual vocabulary for nearest neighbour assignment is created using k-means on 250,000 descriptors with $k = 4096$. The word assignment itself is done by using the maximum inproduct as explained in section 2.4.2.

We learn the Random Forest using 250,000 descriptors. The number of proposed random splits $s$ had little influence in preliminary experiments. In this chapter we set it to half of the number of dimensions of the descriptor. We create 4 trees of depth 10 resulting in 4096 visual words.

For classification we optimize the slack parameter using 3-fold cross validation. We use the prior probabilities to set the weights for the training samples such that the positive and negative training sets are weighted equally: Positive examples get weight $1/P(\text{pos})$, negative examples get weight $1/P(\text{neg})$.

Most of our implementation is done using optimized Matlab code. We created C++ implementations (MEX-files) for the random forest assignment, calculating the maximum of a matrix, the $\chi^2$ distance, and the histogram intersection kernel.
We used the C++ implementations (MEX-files) of the anisotropic Gaussian Filtering [37], the LIBSVM implementation [15] which allows the use of precomputed kernels, and the adjusted LIBSVM implementation of [54] for the fast histogram intersection.

2.6 Results

This section presents all results. For clarity, throughout this section we use percentages to denote computational efficiency and MAP scores to denote retrieval performance.

2.6.1 Baseline

The baseline is given in figure 3, using DIFT $4 \times 4$, nearest neighbour assignment and a SVM classifier using the $\chi^2$-kernel.

The retrieval effectiveness of this pipeline is 0.447 MAP, comparable with [58, 90] for similar settings. In practice this means that for this dataset, for each class on average 80% of the top ten images and 60% of the top 100 images is correct.

The processing of the total pipeline takes 786 ms, where 10% of the time is used for descriptor extraction, 81% is used for word assignment, and 9% is used for classification. Word Assignment time mainly consists of calculating the inner product between the visual vocabulary and the descriptors. Classification time can be split into 74 ms per image for pre-calculating the $\chi^2$-kernel and 1 ms per image for calculating the classification function for all 20 classes. Notice that the 1 ms of calculating the classification function stays the same throughout all experiments on the Pascal VOC dataset as it is dependent only on the number of training and test samples and on the number of classes.

By sampling at different scales, combining different (colour) variants of SIFT during classification and by using the spatial pyramid we obtain a MAP of 0.57, comparable to the best results reported on this dataset [58, 90].

2.6.2 Descriptors

We compare the SIFT, DIFT and DURF descriptors with various spatial configurations as described in section 2.5.3, as well as the DAISY descriptor. Results are given in figure 6.

The retrieval performance among the SIFT $4 \times 4$, DIFT $4 \times 4$, DURF $4 \times 4$, and DAISY descriptors is comparable. Compared to SIFT $4 \times 4$, descriptor extraction is 17% faster for Daisy, 500% faster for DIFT $4 \times 4$, and 3200% faster for DURF $4 \times 4$. Furthermore, compared to both SIFT $4 \times 4$ and DIFT $4 \times 4$, visual word assignment is 37% faster for DURF $4 \times 4$ and 46% slower for DAISY, due to the respectively smaller and larger dimensionality of these descriptors.

The retrieval performance for the $2 \times 2$ spatial configurations for DIFT and DURF is respectively 0.01 and 0.02 MAP lower than their $4 \times 4$ counterparts. While descriptor extraction speed is only slightly faster, word assignment speed is increased significantly due to the lower dimensionality: For DIFT the speed increase of the $2 \times 2$ configuration is 100%, for DURF this is 40%.

The DAISY descriptor is not much faster than SIFT $4 \times 4$, contrary to [92]. This is because DAISY is created for sampling at each pixel rather than each 6-th pixel.
Figure 6: Retrieval performance (a), descriptor extraction time (b), and visual word assignment time (c) for various descriptors. The visual word assignment time (c) is for nearest neighbour assignment using a k-means visual vocabulary.

Figure 7: Random Forests versus k-means nearest neighbour assignment: Retrieval performance (a) and visual word assignment time (b) for various descriptors. Textons have no comparison with k-means as they are designed for use with Random Forests only.
When sampling each pixel, we measured that DAISY is 2000\% faster than SIFT 4 × 4. However, in such a scenario its speed increase relies on doing summations over subregions using convolutions. This can also be done for both SIFT and SURF, hence DAISY is never faster.

To conclude, DIFT 4 × 4 and DURF 4 × 4 have optimal accuracy at a good computation time. DIFT 2 × 2 and DURF 2 × 2 are good alternatives when speed is more important than obtaining the highest accuracy. DAISY gives also a good accuracy but is slow relative to DIFT and DURF. Therefore we will not include this descriptor in subsequent experiments.

2.6.3 Word Assignment

Random Forest

In this experiment we compare the nearest neighbour assignment with Random Forests. We do this on the SIFT, DIFT, and DURF descriptors of our previous experiment. Additionally, we will include the Semantic Textons [79] which are designed to work with Random Forests. Results are shown in figure 7.

Considering the retrieval performance in figure 7a we observe that the MAP values for the 4 × 4 configuration of DIFT stays approximately the same with a MAP decrease of 0.004. DURF 4 × 4 decreases with 0.026. However, in contrast to nearest neighbour assignment, DIFT 2 × 2 and DURF 2 × 2 outperform their 4 × 4 counterparts. DIFT 2 × 2 even has a performance comparable to the baseline. Because the 2 × 2 versions have lower dimensionality and less redundancy, it suggests that Random Forests are sensitive to these aspects. This will be further examined in our PCA experiment in section 2.6.3.

The accuracy of the Semantic Textons is comparatively low, with a MAP of 0.364. This result is obtained using a Random Forest which uses only a single dimension in its decision nodes. We also experimented with including decisions on multiple dimensions exactly as in [79]. But both execution time and accuracy were very similar (data not shown). We conclude that Semantic Textons are less powerful than histograms of oriented gradients for image retrieval using Bag-of-Words. We exclude them in subsequent experiments.

In terms of word assignment speed, figure 7b shows that the Random Forest does what it is designed for and results in a speed improvement of a 3500-4500\%. As it also gives good retrieval performance we include the Random Forest in subsequent experiments.

As discussed in section 2.4.2, the number of computational operations performed while doing Random Forest assignment is independent of the dimensionality of the region descriptors. But as can be seen in figure 7b, higher dimensional descriptors have a larger word assignment time. We attribute this to longer memory access in larger blocks of memory.

Subsampling

This experiment explores speeding up visual word assignment by sampling only part of the extracted descriptors from an image. Results are shown in 8.

We see that word assignment speed is linear in the number of descriptors. Retrieval performance is bounded: more descriptors are better but it stabilizes at a certain point. An accuracy of 0.314 MAP when using 10\% of the descriptors.
Figure 8: Subsampling: Classification accuracy (a) and visual word assignment time (b) when using fewer descriptors of the image. Dashed lines denote the baseline scores.

Figure 9: PCA: Classification accuracy (a) and visual word assignment time (b) when using Principal Component Analysis to reduce the number of dimensions. Dashed lines denote the baseline scores.
(about 450 descriptors per image) may seem low compared to other published work. Additional tests verified that this is because we use dense sampling rather than interest points (data not shown). As interest points focus on the most important parts of an image and hence generate the most important visual words, fewer descriptors are necessary to get reasonable results.

Using fewer descriptors per image is only viable when speed is more important than accuracy in which case a Random Forest assignment is preferred. Comparing the descriptor extraction time and the Random Forest assignment time we see that the Random Forest is 500% faster, therefore no significant speed-ups can be achieved by subsampling. For DURF 4 × 4, descriptor extraction time and word assignment time is about equal so here one can gain some extra speed improvement at a considerable loss in accuracy.

For this dataset we recommend not to use subsampling.

**PCA**

We now use PCA to reduce the dimensionality of the descriptors. Figure 9 shows the results.

For retrieval effectiveness we observe that the use of PCA increases performance for the Random Forest assignment. For DIFT 4 × 4 this performance is maximally 0.462, which is 0.015 MAP higher than the baseline and 0.02 MAP higher than the Random Forest without PCA.

Reduction of the number of dimensions has little influence on accuracy until a certain percentage of dimensions have been reached: for nearest neighbour assignment accuracy stays the same until using half of the total dimensions, for Random Forests accuracy stays the same until using one third of the dimensions. After this point the retrieval performance drops significantly.

At the point where retrieval performance is the same, computation time improves for nearest neighbour assignment with 34% for DIFT 4 × 4 and 24% for DURF 4 × 4. At this point computation time for Random Forests with PCA is about 10% slower than Random Forests without PCA step, which is a speed decrease of only 2% with respect to the whole Bag-of-Words pipeline.

The removal of redundancy in the dimensions by using PCA increases performance for the Random Forest. It implies that the decorrelation of the dimensions has a positive influence on the decision boundaries of the trees of the Random Forest. This can be understood by the fact that each decision node works on a single dimension. After decorrelation these dimensions contain different rather than overlapping information, making each decision and hence the overall decision better.

Because PCA gives a speed improvement for nearest neighbour assignment and an accuracy improvement for a Random Forest, its use is always beneficial.

**Spatial Pyramid**

This experiment compares a combination of 1-4 horizontal image regions and 1-4 vertical image regions for the spatial pyramid [51]. The retrieval performance and computational efficiency for DIFT 4 × 4 for different pyramid divisions is shown in figure 10 and 11 respectively. Results for DURF 4 × 4 show the same trends.

In terms of retrieval performance, division into 3 horizontal regions is optimal and increases accuracy for all pipelines by 0.02-0.04 MAP. 2 or 4 horizontal regions
also give good improvements of around 0.02 MAP. We attribute the increase in accuracy to the crude floor/object/sky distinction it makes.

Our results show that vertical divisions only decrease accuracy in this varied dataset. Even distinguishing the middle part of the image does not lead to extra performance, except marginally for the “horse” and “motorbike” classes, which are typical objects of affection.

The word assignment speed is negligibly slower for relevant image divisions, as can be seen in figure 11a. In contrast, classification time increases considerably as the visual word frequency histograms increase linear in the number of sub-regions. A division into two or three image regions increases computation time with respectively 61% and 116%. Recall that the difference in classification time is only in pre-computing the kernel: afterwards the evaluation of the classification function takes 1 ms for all experiments on this dataset.

To make sure that the increase in retrieval performance is due to the spatial pyramid rather than more statistics, we performed a control experiment in which we increased the visual vocabulary size to 16,384 visual words. For the Ran-
dom Forest we did this by increasing the depth of the trees to 12. The resulting frequency histogram is as large as a $1 \times 4$ or $2 \times 2$ subdivision of the image. However, retrieval performance did not increase significantly: For DIFT $4 \times 4$ retrieval performance using the large vocabulary is 0.443 MAP for k-means and 0.449 MAP for random forests.

Summarized, the spatial pyramid can be used for good classification improvements at a high computational cost which can not be obtained by simply increasing the visual vocabulary. For this varied dataset only horizontal image divisions should be used.

### 2.6.4 Classification

In this experiment we compare the $\chi^2$-kernel, RBF kernel, and the Histogram Intersection kernel. Results are presented in figure 12.

![Figure 12](image)

**Figure 12:** Comparison of $\chi^2$ and an Euclidean distance matrix for classification accuracy (a) and classification time (b). Classification time is measured for a single image for all twenty classes. Classification time is dependent only on the size of the visual vocabulary which is the same for all methods.

As expected, in terms of retrieval performance the $\chi^2$-kernel is best: its accuracy is about 0.03 MAP higher than the Histogram Intersection kernel and about 0.04 MAP higher than the RBF kernel. The fast, linear approximation of the Histogram Intersection kernel is as accurate as the exact version.

The classification speed of the linear approximation of the Histogram Intersection kernel is best. It is 1800% faster than the $\chi^2$-kernel. The RBF kernel is 1400% faster than the $\chi^2$-kernel. The huge difference between calculating the RBF-kernel and the $\chi^2$-kernel can be attributed to the use of efficient matrix multiplications in calculating the RBF-kernel which is not possible for the $\chi^2$-kernel.

In our experiments the linear approximation of [54] is 300% faster than the precomputed Histogram Intersection kernel, much less than the 5,000-200,000% speed improvement reported in [54]. This is because the pre-computed kernel, which takes up the majority of the classification time, is reused for all 20 classes. Furthermore, the visual word frequency histograms in our pipeline are sparse, which we exploit in our computationally efficient Histogram Intersection implementation.

Looking at figure 12, one notices that using the Histogram Intersection kernel instead of the $\chi^2$-kernel works better when using a k-means visual vocabulary
than a Random Forest. Looking at the distribution of visual word frequencies, we observed that the visual word frequencies are more unbalanced for Random Forests than for k-means. Theoretically, the $\chi^2$ distance somewhat normalises this. The Histogram Intersection kernel does not. Therefore we made the visual word counts more balanced by simply taking the square root of these counts. Computational costs are negligible compared to the rest of the pipeline and for DIFT $4 \times 4$ and DURF $4 \times 4$ this improves the MAP score with 0.01.

To conclude, the fast linear approximation of the Histogram Intersection kernel is the preferred method for computational efficiency as it gives higher accuracy and is faster than the RBF kernel. If the Histogram Intersection kernel is used in combination with the Random Forest it is advised to balance visual word frequency histograms by taking the square root. For optimal classification accuracy the $\chi^2$-kernel is preferred.

### 2.6.5 The Spatial Pyramid Revisited

We now apply the linear approximation of the Histogram Intersection kernel to the horizontal spatial pyramid divisions. Classification results are given in figure 13. Classification time is 6 ms for a division into four subregions, and 4 ms for all other divisions.

![Figure 13](image.png)

**Figure 13:** Retrieval performance for the $\chi^2$-kernel without the spatial pyramid and the fast approximate histogram intersection kernel while using only horizontal divisions of the spatial pyramid.

The retrieval performance of the approximated histogram intersection kernel using three horizontal divisions gives the same or better retrieval scores than the $\chi^2$-kernel baseline without the spatial pyramid. Classification speed at this point is 1800% faster than the baseline.

Theoretically, classification time goes up linear in the size of the word frequency histograms. As we did not observe this trend, we did some extra experiments which showed the caching problems for small word histograms. Hence for small histograms our measurements overestimate the classification times.
Summarized, the approximate Histogram Intersection kernel allows for the inclusion of the Spatial Pyramid at the expense of a small increase of total computation time. It is therefore recommended to include the (horizontal) Spatial Pyramid while using the approximate Histogram Intersection kernel.

### 2.6.6 Supervision in Random Forests

The experiments in this chapter provide a very fast Bag-of-Words pipeline which is applicable to large scale datasets in the order of hundreds of thousands of images and beyond. However, large scale datasets tend to come with a large number of classes which are typically not defined from the start. As the Random Forest in our pipeline is created in a supervised manner using the class labels of the images, this begs the question: *If we train a Random Forest on classes other than it is applied, will it still give good results?*

Using the same experimental setup as before, we learn the Random Forest on 5 classes and compare this with the fully supervised Random Forest. Retrieval performance is averaged over 4 trials where we used different classes for learning in each iteration. The results are presented in figure 14.

**Figure 14:** The influence of supervision of Random Forests on accuracy: A comparison of full supervision with supervision on only 25% of the classes.

For DIFT $4 \times 4$, using all classes for learning the Random Forest is only 0.007 MAP better than using 5 classes. For DURF $4 \times 4$ there is no difference. We furthermore examined the possibility that classification scores were different for the classes on which the tree was learned and on which the tree was not learned. Perhaps surprisingly, this was not the case; the Average Precision scores for the classes on which the tree was not learned was equal to the completely supervised trees.

We conclude that by using only a few classes for learning one obtains a visual vocabulary which generalizes well to other object classes. This makes Random Forests applicable for datasets where the labels are not defined in advance.

### 2.7 Discussion on Speed

In this chapter we have presented various fast alternatives for each of the components of the Bag-of-Words pipeline. In this section we analyse their computational efficiency. We will limit ourselves to methodological improvement. Hardware optimization falls outside the scope of this chapter.
2.7.1 Descriptor Extraction

For the descriptors used in this chapter, descriptor extraction consists of three phases: (1) Apply a gradient filter on a certain scale. (2) Differentiate between positive and negative responses using the absolute value function. (3) Sum responses over a small region.

Apply Gradient Filter

For SIFT, we use a fast (approximate) recursive Gaussian derivative filter \[37\] for the diagonals and an exact derivative filter for the horizontal and vertical directions.

The Haar-wavelets used in SURF are a crude but very fast approximation of the Gaussian derivative filter, using only additions and subtractions of the pixel values itself. By design, the number of computations is easily optimized for a certain scale and resolution.

Absolute Value

Taking the absolute value requires a single test per pixel response and is therefore optimal.

Sum Responses over Region

We sum responses by using a linear interpolation over subregions by using two matrix multiplications. As we use sparse matrices, the summed responses of a region are calculated using \(O(\alpha b + b)\) additions and the same order of multiplications, where \(\alpha \times b\) is the size of a region. This is close to the optimal of \(O(\alpha b)\) additions and multiplications, where our use of existing matrix multiplication libraries gives us an edge over a trivial optimal implementation.

Summing the responses over subregions can be done without any interpolation, in which case only additions are needed over a smaller area, resulting in a small overall speed gain. However, experimenting shows a slight drop in retrieval performance if interpolation is omitted.

2.7.2 Word Assignment

We interpret the result of the descriptor extraction phase as follows. Each descriptor which is extracted from an image can be represented as a point in descriptor space. A whole image is represented as a cloud of points in descriptor space, where the clouds of different images can be made up from a varying number of points. In most Bag-of-Words pipelines, the cloud is converted to a k-dimensional vector by dividing the descriptor space into sub-volumes and making a histogram of the number of points per sub-volume. Typically, these sub-volumes are defined by a visual vocabulary created using k-means clustering. These histograms are then compared using a distance measure, which in effect measures the overlap of the cloud. From this perspective, one can understand the work on creating visual vocabularies (e.g. [47, 62, 112]) as carving the descriptor space into sensible sub-regions. The work on distance measures or kernels (e.g. [46, 115]) can be understood as finding the relative importance of low- and high-density regions.
Alternatively, one could measure the distances between image clouds of descriptors points directly. These distances are then used in a distance based classifier such as a SVM. A known, efficient technique of comparing two clouds directly is the Earth Mover Distance which has an empirical complexity of around $O(n^3 \log n)$ [75]. This technique is successfully applied to Bag-of-Words by [115], but to make this computationally feasible they first compress each descriptor cloud to 50 points using k-means clustering. However, [115] report that calculating distances using nearest neighbour assignment and the $\chi^2$ kernel is faster and gives comparable accuracy.

In this chapter we use the Random Forest as a fast algorithm to project a cloud in descriptor space onto a $k$-dimensional vector. The Random Forest is a binary decision tree using $O(n \log k)$ operations for word assignment, where $n$ is the number of descriptors. In contrast, word assignment using a visual vocabulary with nearest neighbour assignment has a considerable larger complexity of $O(nkd)$, where $d$ is the dimensionality of the descriptor.

An alternative fast word assignment algorithm is a hash function [95] which is of complexity $O(nd)$. But as in our experiments $\log k < d$, the Random Forest uses less operations and is therefore faster.

2.7.3 Classification

The fastest classifier we use is the linear approximation of the Histogram Intersection kernel which has a computational complexity of order $O(k)$. The $\chi^2$-kernel results in a classification complexity of order $O(km)$, where $m$ is the number of support vectors.

A fast alternative would be tree-based classifiers which have a complexity linear with respect to the depth of the tree and the number of trees used. Such a classifier would be faster only if the total number of decisions is less than $k$. It is an interesting research question whether tree-based classifiers are able to perform as well as the Histogram Intersection based SVM, where we observe that a tree-based classifier works well only if there are a few highly informative visual words. In contrast, the SVM works also well if there are large numbers of visual words with low information.

2.8 Conclusions

This chapter presented an evaluation of fast Bag-of-Words components to accelerate visual concept classification. We presented results on the Pascal dataset and validated them on the MediaMill Challenge and 15 Natural Scenes databases. Results are similar for all datasets and therefore dataset independent. The presented evaluation leads us to recommend two Bag-of-Words pipelines, one which emphasises accuracy and one which emphasises computational efficiency.

For high accuracy at a minimal computational effort we recommend the Bag-of-Words pipeline shown in figure 15, which consists of DIFT $4 \times 4$ descriptors, Random Forest assignment in combination with PCA, a two-level Spatial Pyramid using only horizontal subregions, and the $\chi^2$ based SVM. On the Pascal dataset this pipeline computes classification scores in 297 ms per image with a MAP of 0.501, which is 5% more accurate and 7 times faster than the traditional scheme in figure 2.
If speed is essential, we recommend the Bag-of-Words pipeline shown in figure 16, which consists of DURF $4 \times 4$ descriptors, Random Forest assignment in combination with PCA, a two-level Spatial Pyramid using only horizontal subregions, balancing visual word counts using the square root, and the fast approximate Histogram Intersection based SVM. On the Pascal dataset this pipeline computes classification scores in 30 ms per image with a MAP of 0.464, which is 3% less accurate than the traditional scheme but 69 times faster. The discussion of section 2.7 shows that this pipeline is close to optimal in terms of computational efficiency.

Results for the MediaMill Challenge and Natural Scenes database for these two pipelines are presented in table 2. Descriptor extraction and word assignment is faster due to smaller image sizes. For the accurate pipeline the classification times are largely dependent on the size of the training set which in the MediaMill challenge becomes a serious bottleneck. After calculating the kernel matrix the classification function itself takes less than 1 ms for the 15 Natural Scenes database. For the MediaMill Challenge this classification function takes 117 ms, which takes so long because of the increased number of support vectors, larger memory access and five times as many classes. In contrast, for the fast pipeline classification times are approximately the same if taken per image per class and scale linearly in the number of classes. The difference in classification accuracy between these two pipelines is comparable for all datasets.

The increased computational efficiency of the Bag-of-Words pipeline in figure 16 opens up new applications for automatic visual concept classification: One application is in the domain of television. The system operates in real-time at a rate of 33 frames per second on a single desktop PC. This enables the tagging of all television of a single channel as it is broadcasted. Another application is in the domain of large image databases. Using 5 computers, this pipeline is able to tag 10,000 images per minute for 20 classes, where each additional computer allows the tagging of 20 extra classes. This throughput is sufficient to automatically tag all pictures that are uploaded to Flickr.

Table 2: A comparison of the accurate and fast pipeline for the Pascal VOC dataset, the Natural Scene database, and the MediaMill challenge in terms of milliseconds and MAP. Notice that for the fast pipeline the classification time can be divided by the number of classes. For the accurate pipeline the classification time mostly reflects the calculation of the kernel matrix.

<table>
<thead>
<tr>
<th></th>
<th>DURF</th>
<th>PCA-RF-SP</th>
<th>$\chi^2$</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pascal VOC</td>
<td>77</td>
<td>18</td>
<td>202</td>
<td>0.501</td>
</tr>
<tr>
<td>15 Natural Scenes</td>
<td>28</td>
<td>8</td>
<td>104</td>
<td>0.826</td>
</tr>
<tr>
<td>MediaMill Challenge</td>
<td>36</td>
<td>9</td>
<td>1638</td>
<td>0.409</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>DURF</th>
<th>PCA-RF-SP</th>
<th>Fast HI</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pascal VOC</td>
<td>14</td>
<td>10</td>
<td>6</td>
<td>0.464</td>
</tr>
<tr>
<td>15 Natural Scenes</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>0.778</td>
</tr>
<tr>
<td>MediaMill Challenge</td>
<td>6</td>
<td>7</td>
<td>26</td>
<td>0.375</td>
</tr>
</tbody>
</table>
2.8 Conclusions

Figure 15: The recommended Bag-of-Words pipeline when the focus is on accuracy. The accuracy is 0.501 MAP. The classification time is 297 ms, 600% faster and 0.025 MAP more accurate than a traditional Bag-of-Words pipeline (see figure 2).

Figure 16: The recommended Bag-of-Words pipeline when the focus is on speed. Total classification time is 30 ms per image, or 33 frames per second. The accuracy is 0.464 MAP. This is 0.01 MAP less accurate yet 6800% faster than a traditional Bag-of-Words pipeline (see figure 2).
The visual extent of an object reaches beyond the object itself. This is a long standing fact in psychology and is reflected in image retrieval techniques which aggregate statistics from the whole image in order to identify the object within. However, it is unclear to what degree and how the visual extent of an object affects classification performance. In this paper we investigate the visual extent of an object on the Pascal VOC dataset using a Bag-of-Words implementation with (colour) SIFT descriptors.

Our analysis is performed from two angles. (a) Not knowing the object location, we determine where in the image the support for object classification resides. We call this the normal situation. (b) Assuming that the object location is known, we evaluate the relative potential of the object and its surround, and of the object border and object interior. We call this the ideal situation. Our most important discoveries are: (i) Surroundings can adequately distinguish between groups of classes: furniture, animals, and land-vehicles. For distinguishing categories within one group the surroundings become a source of confusion. (ii) The physically rigid plane, bike, bus, car, and train classes are recognised by interior boundaries and shape, not by texture. The non-rigid animals dog, cat, cow, and sheep are recognised primarily by texture, i.e. fur, as their projected shape varies greatly. (iii) We confirm an early observation from human psychology [7]: in the ideal situation with known object locations, recognition is no longer improved by considering surroundings. In contrast, in the normal situation with unknown object locations, the surroundings significantly contribute to the recognition of most classes.

3.1 INTRODUCTION

It is widely acknowledged that the visual extent of an object extends beyond the object itself (e.g. [5, 7, 69, 114]). Nevertheless, in the early days of computer vision the visual extent of the object was sought to be precisely confined to its silhouette. And for good reasons as object boundaries (i) are more stable against lighting changes than the rest of the surface, (ii) indicate the object geometry directly, and (iii) reduce the processing requirements. This led to the idea that an object should be correctly segmented before it can be recognised. But the general task of finding the contour-bounded location of an object is very hard to solve and not really necessary for object recognition [84]. In recent years, the use of powerful local descriptors, the increasing size of datasets to learn from, and the
great advances in statistical pattern recognition have circumvented the necessity to know the object location before object-based image classification.

The first step on the road to less localization of the object was to use local region descriptors in a specific spatial arrangement \cite{1,11,30}. This allowed the object to be found based on only its discriminative features. The second step was the introduction of the Bag-of-Words method \cite{82}, which selects interesting regions, converts them to visual words, and uses word counts followed by a spatial verification step to retrieve matching image regions. In the third step, Csurka et al. \cite{18} generalized Bag-of-Words to image classification and removed the spatial verification, relying on interest point detectors to extract visual words from the object. In the final step, the quantity of visual words was found to be more important than the quality of the location of the visual words \cite{47,66}. Therefore these words are no longer extracted at salient points but on a dense, regular grid. This has caused the last notion of object location to be lost in the Bag-of-Words representation which therefore mixes context and object indiscriminately. This is the state-of-the-art of image classification in 2009 \cite{23,83}.

While discarding the object location has its advantages, it is also unsatisfactory. On the one hand, discarding the object location leads to computational benefits and a natural incorporation of context. On the other hand, it is unclear how much information is lost by discarding the object location: the object features of a small object in a large field of view are drowned in the information of its surroundings. Therefore this paper investigates the question: What is the visual extent of an object? This paper is an extension of \cite{98}. Specifically, we investigate the relative influence of the object and its surroundings, and of the object interior and object border.

3.2 RELATED WORK

The influence of context on recognition was researched earlier in human vision. Most notably, Biederman \cite{7} considered five types of relations between the object and its context: (1) Support reflects that objects do not float in the air. (2) Interposition deals with occlusion. (3) Probability is the likelihood that an object is present given the context. (4) Position is the location within the image (e.g. a knife can be found next to a fork). And (5) size is the familiar size of the object. He measured the time it took for humans to identify objects violating one or more of the constraints, which reflects the difficult of identification. In this paper we focus on Biederman’s probability by automatic rather than human vision, leaving the remaining four to another occasion. We measure the difficulty of identification in terms of classification accuracy.

Oliva and Torralba \cite{69} give a good overview of work in visual cognition and cognitive neuroscience on visual context and place this in light of recent advances in computer vision. They conclude that although real-world relationships between individual objects seems the most complete way to describe context, context is already described effectively by its global statistics which ignores object identities and their relations. This was also observed in earlier experimental work in computer vision by Wolf and Bileschi \cite{114}, who showed that high-level semantic context (i.e. the co-occurrence of buildings, trees, sky, etc.) provided no additional information over low-level image statistics. In our paper, we represent
context as a Bag-of-Words representation which can be seen as a form of low-level global image statistics.

The use of the term “context” in computer vision is rather broad. To make the terminology more precise, Divvala et al. [21] identify several types of context as used in the computer vision community. These include Local Pixel Context [12, 20, 33, 39, 80], 2D scene gist context [68], 3D geometric context [44, 64], and semantic context [56, 73, 81]. In their definition the Local Pixel Context captures the contextual information in terms of low-level image statistics while Semantic Context captures contextual information in terms of meaningful categories (e.g. scene class or object class). In accordance with the best image retrieval methods, in this paper we study the visual extent of an object through the use of low-level features rather than semantics; we do not use region class labels as in Markov Random Fields or Conditional Random Fields [12, 80] and we do not use a scene label, but we directly use the features which we extract from the image.

Zhang et al. [115] studied the influence of context in their work. They concluded that the influence of context is marginal within the Bag-of-Words framework. However, the dataset on which they tested it consists of only four classes. On the larger and more diverse Pascal 2010 dataset, we will challenge this finding and investigate whether the influence of context in the Bag-of-Words framework is significant.

Tuytelaars and Schmid [95] visualised a pixel-wise classification based on visual words. Using a large visual vocabulary extracted from a regular lattice, they calculated the likelihood of each visual word belonging to an object. Using an independence assumption on the visual words in the image, they used this likelihood to calculate for each pixel the probability of belonging to a certain object class. This led to an increased insight in Bag-of-Words. Similarly, in our paper we calculate for each pixel how much it contributes to the classifier output. However, as we calculate this contribution from the complete image representation rather than the individual visual words, we do not use an independence assumption. Instead, we provide a direct visualisation of the classification of a state-of-the-art Bag-of-Words framework.

Blaschko and Lampert [9] employed context to improve object localisation. But rather than only relying on only global context, they explicitly optimize over the amount of local context around the object. In this paper we use global context, but we investigate the influence of amount of context relative to the size of the object.

Harzallah et al. [42] successfully combined object localisation with object classification for content based image retrieval. Their work can be interpreted as combining object features from the localised object with context features taken from the whole image. Within video, Ullah et al. [100] automatically created object/surround distinctions using motion and object detectors, successfully improving over their normal Bag-of-Words baseline. Both works show that modelling the object location improves results. In this paper we provide an upper bound of retrieval performance when the object is localised, and compare this with the improvements obtained by [42]. Note however, that we give this upper bound while using bounding boxes. As Malisiewicz and Efros [57] showed, this bound is even higher when the object is localised by an accurate segmentation.
3.3 Methodology

This paper investigates the visual extent of an object in image classification. Over the years, the Bag-of-Words method has been established as the best framework in the major retrieval benchmarks such as the TRECVID high-level feature extraction task for retrieving video [83] and the Pascal VOC Classification task for retrieving images [23]. In this paper we build on our state-of-the-art Bag-of-Words pipeline which won the Pascal VOC 2008 classification task and which was a runner-up in 2009.

We follow two lines in our investigation, visualised in Figure 17. The first line is the normal situation where we apply a visual concept detection algorithm and determine which image parts contribute how much in identifying the target object. The second line is the ideal situation where we use the known object locations to isolate the object, surround, and object interior and object border. For each of these image parts we create a separate representation and examine their retrieval performance. The first line shows what currently is measured, and the second reveals what could be measured.

![Figure 17: The two main lines of our analysis: The ideal line on the left uses the ground truth object locations to divide the image into object and surround, and object interior and object border before classification. The normal line on the right first classifies the image, projects the classification score back on the image and then aggregates classifier scores over the object and surround, and object interior and object border.](image)

We investigate the visual extent of an object in the Bag-of-Words framework in terms of the object surround, object border, and object interior. We split this in two separate experiments. In one experiment we investigate the influence of
the surround with respect to the complete object. In the other experiment we investigate the influence of the object border with respect to the object interior.

We use the ground truth object locations to isolate the object from its surround in both lines of our investigation. As the Bag-of-Words framework thrives using lots of data, we use a large dataset where the locations are given in terms of bounding boxes. To make a better distinction between object and surround and object interior and object border, we also perform the analysis on a smaller dataset where the locations are given in terms of a segmentation. In the normal situation we make the distinction between object/surround and interior/border after classification on the test set only. In the ideal situation we make this distinction beforehand on both the training and test set. When there are multiple instances of the same class we combine their measurements to avoid measuring object features in its surround.

3.3.1 Dataset

We choose to use datasets from the widely used Pascal VOC challenge as this allows for a good interpretation and comparison with respect to other work. We benchmark our Bag-of-Words algorithm on the Pascal VOC 2007 classification challenge to show our framework is competitive. Our analysis is done on two Pascal VOC 2010 datasets. First, we use the classification dataset which provides the object locations in terms of bounding boxes. In this dataset we emphasize quantity of annotations over the quality of annotations. Second, we use the segmentation dataset which is much smaller but provides more accurate object locations in terms of segments. For the Pascal VOC 2010 datasets we use the predefined train set for training and the val set for testing.

The Pascal VOC datasets we use consist images from www.flickr.com, containing twenty different object classes: aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining-table, dog, horse, motorbike, person, potted-plant, sheep, sofa, train, and TV/monitor. Some images contain multiple classes. The 2010 classification set consist of 4998 train images and 5105 val images. The 2010 segmentation set consists of 964 train images and 964 val images.

Classification performance of the Pascal VOC dataset is measured by the interpolated Average Precision of a ranked list. In this paper we use the more standard Average Precision as it enables us to create a confusion matrix as we present shortly. The Average Precision is defined as

$$ AP = \frac{1}{m} \sum_{i=1}^{n} \frac{f_c(x_i)}{i}, $$

(3.1)

where: $n$ is the number of images. $m$ is the number of images of class $c$. $x_i$ is the $i$-th image in the ranked list $X = \{x_1, \ldots, x_n\}$. Finally, $f_c$ is a function which returns the number of images of class $c$ in the first $i$ images if $x_i$ is of class $c$ and 0 otherwise. This measure has range $[0, 1]$ where a higher number means better performance.
3.3.2 Evaluation Matrix

To facilitate analysis, we developed a confusion matrix based on the Average Precision, which we call Confusion Average Precision Matrix or CAMP. The CAMP includes the Average Precision in its diagonal elements and, similar to a confusion matrix, shows which classes are confused.

We define the confusion or off-diagonal elements of the CAMP as the total loss of Average Precision of encountering a specific non-target class in the ranked list. To calculate the loss we traverse the ranked list in decreasing order of importance. When a non-target class is encountered at position \( i \), the loss \( L \) is the difference between the AP assuming a perfect ranking from position \( i \) and the AP assuming a perfect ranking from position \( i + 1 \). More formally, let \( f_c \) be a function which returns the number of examples of class \( c \) in the first \( i \) entries in the ranked list, and let \( r = m - f_c(x_i) \). Now we can calculate the loss \( L \) at position \( i \) as

\[
L(x_i) = \frac{1}{m} \left( \sum_{j=1}^{r} \frac{f_c(x_i) + j}{i + j - 1} - \sum_{j=1}^{r} \frac{f_c(x_i) + j}{i + j} \right).
\] (3.2)

The total confusion with a non-target class \( d \) is the sum of loss to that class, calculated by \( \sum_{x_i \in d} L(x_i) \). As we measure confusion in terms of loss, by definition the AP plus the sum of the loss over all classes adds to one.

3.3.3 Bag-of-Words Framework

A condensed overview of our Bag-of-Words implementation [99] is given in Table 3. We sample small regions at each pixel which is an extreme form of sampling using a regular, dense grid. [47, 66]. From these regions we extract SIFT [53] and four colour SIFT variants [103] which have been shown to be superior for image retrieval [61, 103, 115]. Thus we use intensity-based SIFT, opponent-SIFT, rg-SIFT (normalized RGB), RGB-SIFT, and C-SIFT. Normally, SIFT consists of 4 by 4 subregions. However, we want our descriptors to be as small as possible in our experiments to be able to make the distinctions between object interior, object border, and object surround as crisp as possible. We therefore extract SIFT features of 2 by 2 subregions, which degrades performance no more than 0.02 MAP as shown in section 3.4.1. The size of such SIFT patch is 8 by 8 pixels. We later verify our results on normal 4 \times 4 SIFT, which is 16 by 16 pixels.

For the creation of a visual vocabulary we use a Random Forest [63] in combination with PCA on the descriptors to reduce the dimensionality by a factor 2. This yields equally accurate results as using a k-means visual vocabulary, yet is much faster [63, 99]. Our Random Forest consists of 4 trees of depth 10, resulting in a total size of 4,096 visual words. To train a tree from the Random Forest we use the extremely randomized trees algorithm [35], using 500,000 labelled descriptors sampled randomly from the training set, where the labels are obtained from the annotation at image level.

For classification we use a Support Vector Machine (SVM), which is currently the most popular classifier in Bag-of-Words due to its robustness against large feature vectors and sparse data. The \( \chi^2 \) kernel was found to be the best choice for the kernel function [46, 115]. However, we use the Histogram Intersection based SVM, which allows us to back-project the output of the classifier onto the
3.3 Methodology

<table>
<thead>
<tr>
<th>Descriptor Extraction</th>
<th>Word Assignment</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Sampling each pixel</td>
<td>• PCA dimension</td>
<td>• SVM:</td>
</tr>
<tr>
<td>• Size: 8 × 8 pixels</td>
<td>reduction by 50%</td>
<td>- Hist Int kernel</td>
</tr>
<tr>
<td>• Descriptors:</td>
<td>• Random Forest:</td>
<td>• Image Divisions:</td>
</tr>
<tr>
<td>- 2 × 2 SIFT</td>
<td>4 binary decision</td>
<td>* Spatial Pyramid</td>
</tr>
<tr>
<td>- 2 × 2 opp-SIFT</td>
<td>trees of depth 10</td>
<td>- 1 × 1, 1 × 3</td>
</tr>
<tr>
<td>- 2 × 2 rg-SIFT</td>
<td></td>
<td>* Ground truth loc.</td>
</tr>
<tr>
<td>- 2 × 2 RGB-SIFT</td>
<td></td>
<td>- object/surround</td>
</tr>
<tr>
<td>- 2 × 2 C-SIFT</td>
<td></td>
<td>- interior/border</td>
</tr>
</tbody>
</table>

Table 3: Overview of our Bag-of-Words implementation. In our two lines of analysis we divide the image into subregions by either using the Spatial Pyramid or the ground truth object locations.

image as we explain in section 3.3.4. By taking the square root of the visual word histograms before normalisation we compensate for high frequent visual words, which makes the Histogram Intersection kernel almost as good as the χ² kernel [99]. In fact, when we sample visual words every pixel, both the χ² kernel and the histogram intersection kernel yield similar accuracy.

The original Bag of Words framework is orderless. Therefore Lazebnik et al. [51] introduced a weak spatial order by using their spatial pyramid, in which an image is divided into regular subregions. Visual word frequency histograms are obtained from each region separately. We use the spatial pyramid in half of our experiments. In the normal setting we create visual word histograms for the whole image and a subdivision into three horizontal segments, shown to be a good division by several researchers [58, 90, 99]. In the ideal setting we divide the image into the three subregions representing surround, object interior and object border by using the ground truth bounding boxes. To keep the total size of the final histogram representations similar we refrain from using the spatial pyramid in the ideal setting. This omission means that the upper bound of retrieval performance in the ideal setting is underestimated. It does not influence the general conclusions of this paper.

3.3.4 Analysis without knowing the object location

The line of analysis where the object locations are unknown shows how all parts of the image are used for classification by current state-of-the-art methods. We first classify images using a standard, state-of-the-art Bag-of-Words framework. After classification, we project the output of the classifier back onto the image to obtain a visualisation of pixel-wise classifier contributions; the sum of the pixel-wise contributions is equal to the output of the original classifier, which measures the distance to the decision boundary.

After we have created the pixel-wise classifier contributions, we use the ground truth object locations to determine how much each image part (i.e. surround, object, object interior, object border) contributes to the classification. When an image contains multiple objects of the same class, we add contributions of all its locations together. When an image contains the target class, its location is used to make the distinction into object, surround, object interior, and object border.
If the image does not contain the target class, we use the class with the highest classification contribution to make this distinction. This allows us to create a partitioning for both target and non-target images, which we need in order to calculate the Average Precision that is defined over the whole dataset (there is no “true” partitioning into the object and its surround for non-target images).

Back-projection of the Classifier Score

We want to determine the relative contribution of each pixel in the image. This requires dissecting the classification function to determine the relative contribution of each visual word in the image. We follow [54] to rewrite the Histogram Intersection kernel, but in principle any additive kernel can be used [105].

The classification function for a Support Vector Machine can be written as [8]

\[ h(x) = b + \sum_{j=1}^{m} \alpha_j t_j k(x, z_j), \]  

where \( x = \{x_1, \ldots, x_n\} \) is the vector to be classified, \( z_j = \{z_{1j}, \ldots, z_{nj}\} \) is the \( j \)-th support vector, \( \alpha_j \) is its corresponding positive weight, \( t_j \in \{-1, 1\} \) is its corresponding label, \( m \) is the number of support vectors, and \( k(\cdot, \cdot) \) is a kernel function. For the Histogram Intersection kernel

\[ k(x, z) = \sum_{i=1}^{n} \min(x_i, z_i), \]  

the classification function can be written as [54]

\[ h(x) = b + \sum_{j=1}^{m} \alpha_j t_j k(x, z_j) \]  

\[ = b + \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_j t_j \min(x_i, z_{ij}). \]  

As the outer sum in equation 3.6 is over the visual words, the contribution per visual word channel \( w_i \) is calculated as

\[ w_i = \sum_{j=1}^{m} \alpha_j t_j \min(x_i, z_{ij}). \]  

Within an image there are often multiple visual words having the same identity \( i \). We evenly distribute the contribution \( w_i \) over all visual words with identity \( i \). This gives us per visual word in the image its contribution to the classifier score. Using the locations of the patches which generated the visual words, we can project these contributions back onto the image. Examples are shown in Figure 19.
3.3.5 Analysis using the ideal object location

In this line of analysis we use the known object locations to create different representations of the surround, object, object interior, and object border in both the training and test set, yielding hypothetical classification scores. We assign descriptors to an image part based on its centre point. For example, a descriptor is considered to come from an object when its centre is contained within the bounding box of that object. We use the ground truth object locations to create a separate visual word histogram for each of the image parts and analyse their retrieval performance. We create combinations by concatenating these word histograms.

Again, if an image contain multiple objects of the same class we combine their visual word histograms by adding them together. If the image contains the target class its location is used to divide the image into object, surround, object interior, and object border. If the image does not contain the target class, the class with the highest classification score is used to make this distinction. Note that if we would only use the locations and not the labels, i.e. we would always select the class with the highest classification score regardless if that is the target class or not, accuracy could only improve: in the rare cases that a non-target class has a higher classification score than the target class in that same image, the positive image will have a higher ranking. Hence the scores presented in this paper for the ideal setting can be seen as an upper bound if the locations of the objects are known.

3.3.6 Distinguishing object, surround, interior, and border

For boxes, the ground truth locations separate the object from the surround. Note that the nature of the boxes cause some surround to be contained in the object. To separate the object interior from the object border, we define an object interior box as being a factor $n$ smaller than the complete object box while its centre pixel remains the same. To determine the interior box we use the idea that object border contains the shape and the object interior contains texture and interior boundaries, which should be complementary. Separating complementary information should yield better results for the combination, hence we find the optimal interior boxes by optimizing classification accuracy over $n$ on the training set using cross-validation. We found a factor 0.7 to be optimal. This means that 49% of the object is interior and the rest border.

For segments the Pascal VOC dataset only annotates the interior of the object while there is a 5 pixel zone around where the borders of the objects are. We want to ensure that no surround descriptors measure this border zone, and no interior descriptors measure this border zone. As we use the middle of our descriptor as point of reference in the ideal situation, we extend this border zone with half our descriptor size both inwards and outwards. Extending the border outwards yields our outlines of the object. Extending the border inwards yields the separation between object interior and object border. Our object border hence becomes 13 pixels wide. We measured that on average over all objects, 46% of the object becomes interior and the rest border.
3.4 Results

3.4.1 Classification without knowing the object location

We first benchmark our Bag-of-Words system on the Pascal VOC 2007 dataset, on which most results are published. For our normal Bag-of-Words system where we do not know the object location we achieve an accuracy of 0.57 MAP, sufficiently close to recent state-of-the-art Bag-of-Word scores obtained by [42] and [103], which are respectively 0.60 MAP and 0.61 MAP. To enable back-projection with equation 3.7 we use the Histogram Intersection kernel instead of the widely accepted $\chi^2$ kernel [42, 46, 103, 115]. This does not influence classification accuracy: with the $\chi^2$ kernel performance stays at 0.57 MAP. Instead, most of the difference in accuracy between our work and [42, 103] can be attributed to our use of $2 \times 2$ SIFT patches: using the four times as large $4 \times 4$ SIFT descriptor results in a classification accuracy of 0.59 MAP. However, in most of our experiments we favour small SIFT descriptors to minimize the overlap between object and surround, and interior and border descriptors. From now on all results are reported on the Pascal VOC 2010 dataset using $2 \times 2$ SIFT descriptors, unless otherwise noted.

Figure 18 shows the confusion matrix of the normal Bag-of-Words system on the 2010 train+val set. One can see that the classes can be roughly divided into three clusters where most of the confusion concentrates: furniture, animals, and land-vehicles. The classes aeroplane, boat, and person behave differently and cannot be grouped. The high confusion with the person class in the right column of Figure 18 can be explained by the many person images in the dataset. We will use the identified categories in subsequent analysis.

To conclude, we have verified that our Bag-of-Words system is state of the art and we have identified categories to facilitate subsequent analysis.

Localising the Classifier Contributions.

We now investigate qualitatively where the Bag-of-Words classifier obtains the evidence to classify images. We do this for both $2 \times 2$ SIFT used in most of our paper and the more widely used $4 \times 4$ SIFT. We use the method described in section 3.3.4 and show results for top-ranked images (according to $2 \times 2$ SIFT) of the classes aeroplane, boat, cat, car, person, and sofa in Figure 19.

We first observe that the difference between the use of $4 \times 4$ and $2 \times 2$ SIFT descriptors is very small. The former seems to be a blurred version of the latter. Hence the following observations hold for both types of descriptors.

We can see that, generally, in the Bag-of-Words method often small details give either a high positive or high negative contribution to the classifier output. However, while details often stretch beyond the size of the descriptor patch, as seen for example in the ropes of the boats or the contours of the cars and persons, they never coherently cover a complete object or object part. The contours of the cars come closest, but these contours are frequently interrupted by small details with a strong negative response. In homogeneous regions the responses show a considerable amount of noise, as seen for example in the erratic responses of the sky in the boat images. This is possibly caused by local normalisation of the descriptors. Of course, the Bag-of-Words method was designed to work on local details but these visualisations show just how fragmented these details are.
Figure 18: Average Precision Confusion Matrix (CAMP) of the normal situation where the object locations are unknown (using $2 \times 2$ SIFT). By definition of the CAMP the rows sum to one (see section 3.3.2). Notice the within-category confusion in the furniture, animal, and land-vehicle classes.

For the boat class, water and sky yield both strong positive and negative contributions with an overall positive contribution. The horizon line, often sky-water, consistently yields positive information. This shows why sky and water are good contextual indicators of boat. Within the boat only the ropes and masts have a positive response, while their hulls have a strong negative response. In fact, the overall contribution within the boat region is negative(!). This shows that a boat is recognised only by the water and is therefore purely recognised by its function (being in the water).

For the cat images the fur is most discriminative. But like the sky, fur consists of a mix of positive and negative contributions which has a net positive contribution. This suggests that for these kinds of textures looking at small image patches is suboptimal. Furthermore the shape of the cat is not important. We see similar behaviour for the other animals, but for horse the shape of the legs are also im-
The visual extent of an object

Figure 19: Pixel-wise contribution to the classification for top ranked images for the categories boat, cat, car, motorbike, person, and sofa. The original image is followed by the contribution of $2 \times 2$ and $4 \times 4$ SIFT respectively. Dark-blue means a negative and light-yellow means a positive contribution to the classifier. Notice that high positive or high negative contributions are often located on small details. The $4 \times 4$ SIFT images resemble a blurred version of their $2 \times 2$ counterparts. (Best viewed in colour)
important. This suggests that most animals are recognized based on texture rather than shape.

For car, the largest positive contribution to the classifier score is concentrated on the contours and interior boundaries. For the contours especially the roof of the car, the nose, and the wheels yield high positive information. For the interior boundaries the positive information often is concentrated on the lights, grill, bumper, and window-hood boundary. The importance of the contours suggests that cars are mainly recognised through their shape and interior boundaries.

In the motorbike images, all parts of the motorbike give an equal amount of positive information to the classifier score. Only the front wheel gives generally a strong positive contribution. The highest ranked negative examples of motorbike suggest that the strong response of its front wheel causes the confusion with the bicycle class shown in Figure 18.

For person both its contours and inner boundaries are important. The shoulders, upper sides of the head, and the collar/neck boundary often yield a strong positive contribution. The clothes are mildly but erratically positive, yet their overall response is large because of the size of their surface.

In the sofa images primarily true vertical and tilted horizontal edges are important, which may be caused by a sofa or more likely a whole living room in perspective.

3.4.2 Classification in ideal setting with known object location

In this experiment we use the object location to create a separate representation for the surrounding and the object, where the representation of the object may be split into the interior and the border of the object. We compare this with the results of normal situation where the object location is not known.

Figure 20 compares the performance of the normal situation in which the object location is not known with the ideal situation where the object location is known. Clearly, for all classes knowledge of the object location greatly increases performance. The overall accuracy of the normal situation is 0.54 MAP, the accuracy of the ideal situation when making the distinction between object and surround is 0.68 MAP (where no Spatial Pyramid is applied to the object). When creating separate representations for the surround, object interior, and object border performance increases to 0.73 MAP. This shows that the potential gain of knowing the object locations is 0.19 MAP in this dataset.

Similarly, on the segmentation dataset, in the normal situation where the object location is not known the classification accuracy is 0.44 MAP. When separating the object from the surround accuracy rises to 0.62 MAP. If we make a separation between surround, object interior, and object border accuracy improves to 0.69 MAP.

The huge difference between the accuracy without and without knowing the object location shows that the classifier cannot distinguish if visual words belong to the object or surround. We investigate the cause by determining for each visual word the probability that it occurs in an object \( i.e. \) in any of the specified object classes, which is visualised in Figure 21. This graph shows that 1\% of the words have a larger than 90\% probability of describing background. We found that these words describe mostly homogeneous texture \( e.g. \) sky. In contrast, no single word has a larger than 90\% probability of occurring on an object and less than 2\% of
The visual extent of an object

Figure 20: A comparison of the normal situation when the object location is unknown and the ideal situation where the object location is known. Accuracy over all classes for the normal situation is 0.54 MAP, for object+surround this is 0.68 MAP, and for interior+border+surround this is 0.73 MAP.

the visual words occur on an object more than 75% of the cases. Note that these numbers are the same when using 4 × 4 SIFT. This means that no visual words exclusively describes objects and that these visual words are less specific than generally thought.

Results in this section suggest that performance for Bag-of-Words could be improved when the object location is explicitly modelled. Indeed, the work of Harzallah et al. [42] combined a system for object detection, i.e. localizing and classifying objects within images, and a Bag-of-Words object classification system by fusing their respective classifier outputs. In effect the object detection system explicitly modelled the object location on which it based its classification. The combination allowed them to successfully improve the classification score by 0.04 MAP to 0.64 MAP on the Pascal VOC 2007 dataset. However, our experiments on the Pascal VOC 2007 dataset result in an improvement of 0.20 MAP yielding an upper bound of 0.77 MAP if the object locations are known. This bound also applies if their labels are not as explained in Section 3.3.5. This means that while the work of [42] is encouraging, there is still a lot of room for improvement in the Bag-of-Words classification framework by attempting to locate the object within the image.
3.4 RESULTS

Figure 21: The probability of each visual word belonging to an object. The dotted red line is the prior probability. Contrary to general belief, visual words are not very object specific as only 2% of the visual words have a higher than 75% probability to come from an object.

3.4.3 Discussion on Object versus Surround

We now proceed to discuss the relative influence of the object and its surroundings. We do this first using boxes on the large Pascal VOC classification set. Then we perform the same on segments using the smaller Pascal VOC segmentation set.

Object versus Surround using Boxes.

Figure 22 plots the Average Precision for the object against the surround and against the combination of the object and surround for the normal situation where the object location is unknown, Figure 23 plots the same for the ideal setting where the object location is known.

In Figure 22a one can see that for boat and bottle the surroundings are more used than the object for classification in the normal situation. For boat this confirms that it is recognised by only water and sky as seen in Figure 19.

The retrieval performance when using only the surround is low for more than half of the classes in the normal setting. Only bus, boat, bottle, bird, chair, train, and plane yield reasonable performance. The performance for person looks also reasonable, but is close to its random score of 0.37 AP. In contrast, when training and learning on the isolated surroundings, Figure 23a shows that many classes can be retrieved a lot better. Thus, while the surroundings contain information, it is normally not the focus of the classifier.

In Figure 22b we see that the combination of object and surround is better than using the object alone for more than half of the classes. This is not surprising as the classifier was learned on the combination. However, for the classes plant, table, bike, car, motor, dog, person, and tv/monitor the performance of the combination is
**Figure 22:** The retrieval performance of the object, surround, and its combination in the normal setting where the object locations are unknown. (a) the surround versus the object. For *bottle* and *boat* the surround is more important than the object itself. (b) The object versus the combination of object and surround. For *bike, car, dog, motor, plant, person, table,* and *tv/monitor* the performance of the object is very similar to the combination, suggesting context is mostly ignored by the classifier.

**Figure 23:** The retrieval performance of the object, surround, and its combination in the ideal setting with known object locations. (a) the surround versus the object. For all classes the object is more important than the surround. For all classes performance increases significantly over Figure 22a (b) The object versus the combination of object and surround. For most classes performance is similar for the object and the combination. Hence if the object location is considered known, the surround adds little information.
equal to using only the object. For these classes the classifier learns to ignore the surround.

When the objects are considered localised in Figure 23b, for all classes except *bird* and *table*, using surroundings in addition to the object does not yield much improvement over using the object alone. Interestingly, this agrees with the research on human vision of Biederman [7], who found that objects viewed in isolation are recognised equally well as objects viewed in proper context.

Intuitively, the relative size of the object and its surround will impact the results. To see how, we analysed results on two subsets of the images: one where 5-20% of the image is object, and one where 5-20% of the image consists of the surround. In the normal situation, when the images consist mostly of object, only the *bus*, *boat*, *plane*, and *train* class can be still reasonably recognised by their surround. For the classes *bottle*, *bird*, and *chair* more surround is needed to adequately recognise them. For the images with large objects, adding the surrounding yields no performance improvements over using the object alone in the normal situation. When the objects within the images are small, performance drops using only the object features. However, many classes can still better be recognised by the object features: *bike*, *bus*, *car*, *motor*, *plane*, *cat*, *dog*, *horse*, *plant*, *table*, *tv*, and *person* are all better recognised by their object features than their surround. Except for *horse*, *plane*, *bus*, using the now large surround in addition to the object still does not improve recognition performance.

In the ideal setting, for both the sets with large and small surround, the surround does not add any extra information to using the object alone, except again for the *bird* and *table*. For the set with a large surround, the relative performance of the object and surround is similar to Figure 23a. When using only the surround for classification, recognition for images with a large surround is on average 0.11 AP higher than on the images with a small surround. In contrast, using only the object for classification, recognition for images with a large object is on average only 0.03 AP higher than for the images with a small object region, where most benefits are for the classes *bird* (0.19 AP), *chair* (0.42 AP), and *table* (0.16 AP). We conclude that the size of the surround matters in both the normal and ideal situation. For objects its size only matters in the normal situation: once the object is localised, for most objects a larger size does not result in better recognition.

We now continue with analysing the confusion matrices of using only object or surround in the idealized setting when the objects location is known, which are visualized in Figure 24. The confusion matrix of using only surround in Figure 24a looks similar to the confusion matrix of the normal setting in Figure 18. Again, most of the confusion is concentrated within the *furniture*, *animals*, and *land-vehicle* categories. This means that each category shares context, which obviously is the case. For the *car* class something interesting happens. One can see that the *car* context is strongly confused as context for other classes, but not vice versa. This suggests that while the contexts of *bicycle*, *bus*, and *motorbike* are disjunct, the car context includes them all. Indeed, in this dataset the motorbike context is dominated by the countryside and the bus-context is dominated by urban environments, whereas the *car* occurs in both.

Figure 24b displays the confusion matrix when only *object* descriptors are used. Most notably, the confusion within the *furniture* and *land-vehicle* category is very low, which means that confusion within these two categories is mainly caused by the surroundings. Although without the surround bicycles continue to be
confused with motorbikes, and buses with trains. For animals, within category confusion is still high. This means that both context and object are a source of confusion. Intuitively, object descriptors cause confusion because most of the animals are furry and have similar shapes (four legs and a head). In the Section 3.4.4 we will see what causes most confusion: fur or shape.

Object versus Surround using Segments.

We repeated the experiments to analyse the influence of the object and the surround, but this time on fewer data but using more accurate object locations in terms of segments.

The comparison of the influence between the object and its surround in the normal situation for segments looks similar to Figure 22aa, except that performance of using only the object is worse. Hence with fewer training examples the classifier is still able to learn the appearance of the surrounding but has less success in learning the appearance of the object itself. This means that the appearance of the context is simpler than that of an object. In effect, this also means that cow, sofa, bird, and chair join the boat and bottle class in that their surroundings are more important than the object itself when using fewer training examples.

Figure 25a shows that the combination of object and surround is better than using the object alone for more than half of the classes, similar as with the larger dataset on the boxes (Figure 22b). Again, for motor, dog, and plant adding the surroundings does not help. For bike, and tv using also the surroundings has even a negative effect.
3.4 Results

Figure 25: Influence of the object and its surround analysed using segments. Results are similar to Figure 22b and 23b.

In the ideal situation, when comparing object and surround again the results are similar to Figure 23b. Again, most classes can be retrieved reasonably by their surround. However, in contrast to 23a, boat and bird can be recognized equally well by their object as by their surround when this distinction is made more accurate by a segmentation. This suggests that part of the classification performance for using only object using a bounding box can be attributed to the inclusion of a bit of context.

Figure 25b compares the accuracy of the segmented object with the combination of the segmented object and its surround. We see that now beside bird, also the classes boat, chair, plant, table and train benefit from the inclusion of the surround. These classes all have a high variability in appearance, and are difficult to recognise in isolation. To verify whether this change in behaviour comes from the omission of any context while using segments, we repeated the experiment on the segmentation dataset but using boxes. Results were the same. Hence we conclude that the behaviour results from using fewer training examples: to accurately learn the appearance of these relatively difficult classes more training data is needed.

Conclusion Object versus Surround.

In the normal situation where the object location is unknown the surroundings contribute significantly to classification for more than half of the classes. For the classes boat and bird the surroundings are even more important then the object itself. This means that the findings of Zhang et al. [115] that the Bag-of-Words framework can learn to ignore the surroundings holds for some classes, but does not generalize to all classes in larger datasets. In contrast, in the ideal setting when the object locations are known, the surroundings add little additional information for most classes which is in accordance with human vision [7]. Finally,
the surroundings are a source of confusion within the furniture, animal, and land-vehicle categories, but the object itself only causes confusion within animals.

3.4.4 Discussion on Interior versus Border

We now discuss the relative influence of the interior and the border of the object. As the segmentation yields a more accurate distinction than the boxes, we will first discuss the results on the segmented object locations. Afterwards we will verify the observations on the larger dataset using boxes.

Interior versus Border using Segments.

Figure 26 plots the Average Precision for the interior against the border and against the combination of the interior and border for the normal situation with unknown object locations. Figure 27 plots the same for the ideal situation with known object locations.

First we look at the animal classes cat, cow, dog, horse, and sheep. In both the normal and ideal situation, we see from Figure 27a and 26a that the interior contains significantly more information than the border. In Figure 27b and 26b we can see that adding the border as additional information does not improve results, except for the horse when the object location is known. We conclude that the animals are recognised not based on their contours but on their interior. Hence the animals are recognised based mostly on their fur, which was observed earlier for cat in Figure 19.

We now consider the vehicles car, bus, and train. In Figure 27a and 26a we see that also for these classes the interior is more important than the border, yet in contrast with the animals, the border alone still yields good accuracy. As seen in Figure 27b, when the object location is unknown, using the border and the interior yields little improvements over using the interior alone. But Figure 26b shows that when the object location is known, using the border in addition to the interior yields improvements of around 10% for car, bus, and train. Hence both the border (shape) and interior for these classes are important, where the visualizations of the classifier contributions in Figure 19 suggest that the interior is important because of their well-defined interior boundaries.

For bike and plane, Figure 27a and 26a show that the border is more important than their interior. In fact, plane is the only class where, when the object location is known, using the interior in conjunction with the border yields no improvements over using the border alone. For bike both the interior and border are important when the location is known. Hence bike and plane are rigid classes with a well-defined shape which can be recognised best by their border, while for bike the interior is also important.

The classes chair, plant, and sofa are the only classes that behave different in the normal and ideal situation. When the object location is unknown, Figure 27a shows that the border yields more information than the interior. In contrast, for these classes when the location is known the interior yields more information in Figure 26a. This suggest that while the interior yields enough information to discriminate between classes, it yields not enough information to discriminate between the class and the background which is necessary in the normal situation when the object location is unknown. Indeed, intuitively, plant resembles
3.4 Results

Figure 26: The retrieval performance of the object interior, object border, and their combination in the normal situation with unknown object locations. (a) Object border versus object interior. The animal classes dog, cat, cow, sheep, and horse are best recognised by their interior. (b) Object interior versus the combination. Performance of the animal classes does not improve while using the border in addition to the interior.

Figure 27: The retrieval performance of the interior, border, and their combination in the idealised setting where the object locations are considered known. (a) Object border versus object interior. Most animal classes are best recognised by their interior. Most vehicle classes as well yet these are also recognized well by only their border. (b) object interior versus the combination. For the animal classes, using the border in addition to the interior does not yield additional information. For the vehicle classes, the combination of the interior and border do yield improvements.
any background vegetation and sofa may resemble carpet of curtains in the background.

**Interior versus Border using Boxes.**

Figure 28 and 29 show the performance of using only the interior versus the performance of using only the surround when this distinction is made using bounding boxes. The tendencies are similar as in the situation where the segmentation is used to make the distinction. Again, for the animal classes cat, dog, horse, and sheep the interior is more important than the border. However, the border now has a higher performance because for the large, more easily recognizable objects it includes more of the object interior. For the bus, car, train, and plane classes both the interior and the border are equally good for predicting the object class, which corresponds with our earlier observation that these rigid classes can be recognised by their well-defined shape as well as their interior borders.

**Figure 28:** The retrieval performance of the object interior, object border, and their combination for the normal situation with unknown object locations. The distinction is made through boxes. Similar to using segments (Figure 26), the animal classes dog, cat, horse, and sheep are better recognised by their interior.

**Figure 29:** The retrieval performance of the interior, border, and their combination in the idealised setting where the object locations are considered known. The distinction is made in terms of boxes. The animal classes dog, cat, horse, and sheep are best recognised by their interior. The vehicle classes car, bus, and plane are recognised equally well by the interior as their border.

For bike the interior is now more important. This is because the segmentation accurately outlines the wire-frame leaving little surface for the interior. The fact that the interior is important using boxes just shows the importance of the inner frame and parts of the spokes.

**Discussion and Conclusion Interior versus Border.**

The object interior consists of texture and of interior boundaries, reasonably captured by a Bag-of-Words representation. However, this representation may be
less appropriate for the object boundary as the object shape is intuitively better represented by larger fragments with more geometric constraints. However, we saw from Figure 19 that this representation still highlights large parts of object boundaries extending beyond the size of the local patch. Hence while the conclusions made on the relative contribution of the border and the interior may not extend to object recognition in general, it will still be indicative of the relative difficulty of obtaining information of the object border and object interior.

To conclude, our analysis of the object border and object interior showed that the non-rigid animal classes cat, cow, dog, horse, and sheep are mostly recognised by their fur while their shape adds little information. The exception is the horse whose legs likely contribute. For the rigid classes bike, bus, car, and train, both interior boundaries and the border or shape information is used for recognition. For plane only the shape is sufficient for recognition.

3.4.5 Using 4 × 4 SIFT

The results presented in this paper were based mostly on a 2 × 2 SIFT descriptor, as this small descriptor enabled a more crisp separation of the different image parts, especially for the interior/border distinction. To investigate the influence of this choice, we repeated some experiments using the larger 4 × 4 SIFT. For the object/surround distinction we repeated the experiment where the location is given by boxes. For the interior/border distinction we repeated the experiment where the location is given by a segmentation.

For the distinction between object and surround using boxes, results are almost identical to the ones presented in this paper. For the normal situation this should come as no surprise given the similarities in the visualisation of the pixel-wise contribution in Figure 19. For the ideal situation recall that a box already includes some surround. Descriptors within the box will measure a bit more of the surround but not significantly. Descriptors outside the box can measure slightly bigger parts of the object but most of the descriptor is still used for measuring the surround. Overall, using 4 × 4 SIFT yields figures nearly identical to the ones presented in Figure 22 and 23 and does not affect our conclusions.

For the distinction between interior and border, we carefully made our interior such that it does not contain any border. For the larger descriptor, this means that we had to make our interior smaller such that is was only on average 35% of the total size of the object. In both the normal and ideal situation all tendencies are very similar: The border alone becomes slightly more predictive of the class, while the predictive power of the interior remains approximately the same for all classes except bike and plant. For bike this is because half of the classes lose all of its interior. Again, for the animal classes cat, cow, dog, and sheep adding the border to the interior does not yield significant improvements over using the interior alone.

Summarized, results are almost the same when using 4 × 4 SIFT. Hence our conclusions remain valid for this larger, more commonly used descriptor.

3.5 DISCUSSION AND CONCLUSION

This paper investigated the visual extent of an object in terms of the object and its surround, and in terms of the object interior and the object border. Our inves-
tigation was performed from two perspectives: The normal situation where the location of the objects are unknown, and an ideal situation with known object locations.

For the normal perspective we visualised in Section 3.4.1 how the Bag-of-Words framework classifies images. These visualisations indicate that the support for the classifiers is found throughout the whole image occurring indiscriminately in both the object and its surround, supporting the notion that context facilitates image classification [21,69]. While for some classes with a highly varying surround Bag-of-Words learns to ignore the context, as observed by Zhang et al. [115], this does not generalise to all classes. We found that the role of the surroundings is significant for many classes, to the point where for boat and bottle they are even more important for recognition than the object itself. For boat the object area is even a negative indicator of its presence.

At the same time, we have demonstrated in Figure 23b that when the object locations are known a priori, the surroundings do not help to increase the classification performance significantly. After ideal localisation, regardless of the size of the object, the object appearance alone predicts its presence equally well as the combination of the object appearance and the surround.

We showed that no visual words uniquely describe only object or only surround. However, by making the distinction between object and surround explicit using the object locations, performance increases significantly by 0.20 MAP. This suggests that modelling the object location can lead to further improvements within the Bag-of-Words framework, where we see the work of Harzallah et al. [42] as a promising start.

Regarding the surround the following view arises. The surroundings are indispensable to distinguish between groups of classes: furniture, animals, and land-vehicles. When distinguishing among the classes within one group the surroundings are a source of confusion.

Regarding the object features, we have observed differences how classes are being recognised: (1) For the physically rigid aeroplane, bicycle, bus, car, and train classes interior and exterior boundaries are important, while texture is not. (2) The non-rigid animals dog, cat, cow, and sheep are recognised primarily by their fur while their projected shape varies greatly. While SIFT feature values respond to interior boundaries, exterior boundaries, and texture at the same time, the recognition differences suggest that using more specialised features could be beneficial.

Bag-of-Words with SIFT measure texture, interior object boundary fragments, and shape boundary fragments as local details. For identifying the context of an image this is adequate, especially considering that context partially consists of shapeless mass-goods such as grass, sky, or water. In contrast, for objects features more spatial consistency could help. This suggests that future features would render more improvements on recognising objects than on recognising context. Intuitively, this means that when the exact object location is known, context helps less for recognition than our experiment in Figure 23b. This is consistent with the observation by Biederman [7] in human vision that objects viewed in isolation are recognised as easily as objects in proper context.
This chapter addresses the problem of generating possible object locations for use in object recognition. We introduce structured sampling which combines the strength of both an exhaustive search and segmentation. Like segmentation, we use the image structure to guide our sampling process. Like exhaustive search, we aim to capture all possible object locations. Instead of a single sampling technique, we diversify our sampling and use a variety of complementary image partitionings to deal with as many image conditions as possible. Our structured sampling results in a small set of data-driven, class-independent, high quality locations, yielding 99% recall and a Mean Average Best Overlap of 0.879 at 10,097 locations. The reduced number of locations compared to an exhaustive search enable the use of stronger machine learning techniques and stronger appearance models for object recognition. In this chapter our structured sampling enables the use of a powerful Bag-of-Words implementation. We substantially improve the state-of-the-art up to 5.6% for 9 out of 20 classes on the Pascal VOC 2010 detection challenge.

4.1 INTRODUCTION

How can we determine the what and where of an object? How can we ascertain that there is a cow precisely there?

A first intuition has been that an object should first be delineated before it is identified. Segmentation aims for a unique partitioning of the image, where there is one part for one or all object silhouettes in the image, by one generic algorithm. Research on this topic has yielded tremendous progress over the past years [3,17,28,78]. But images are intrinsically hierarchical, as illustrated in Figure 30a. This makes it impossible to delineate all objects with a unique partitioning. So at the least, multiple scales are needed. This was recently overcome by, for example, the hierarchical partitioning of Arbelaez et al. [3].

Yet in spite of all advancements [3], general belief in a universal solution for generic segmentation is waning. There are many conflicting reasons why a region should be grouped together: In Figure 30b the cats can be distinguished by colour, but their texture is the same. Conversely, in Figure 30c in terms of colour the chameleon is similar to its surrounding leaves, yet their texture differs. Finally, in Figure 30d the wheels are wildly different from the car in terms of both colour and texture, yet are enclosed by the car. Single visual features therefore cannot resolve the ambiguity of segmentation. Another problem is that many coherent objects may be composed of regions with different appearance: a face has a very
there is a high variety of reasons that an image region forms an object. In (b) the cats can be distinguished by colour, not texture. In (c) the chameleon can be distinguished from the surrounding leaves by texture, not colour. In (d) the wheels can be part of the car because they are enclosed, not because they are similar in texture or colour. Therefore, to find objects in a structured way it is necessary to use a variety of diverse strategies.

different appearance than the clothing of a human. Hence without recognition it is hard to decide that a face and a sweater are part of one object [94].

A complete opposite approach is to do localisation through the identification of an object. This is the current approach in object class recognition which has made enormous progress in less than a decade [20, 27, 42, 107]. First, an appearance model of the object is learned. Then an exhaustive search is performed where every location within the image is examined as to not miss any potential object location [20, 27, 42, 107]. The accuracy gains for this strategy are due to advances in fast machine learning techniques and fast appearance models [20, 42, 107], and recently by a smart combination of a whole object and its parts [27].

However, the exhaustive search itself has several drawbacks. Searching every possible location is computationally infeasible. Therefore the search space is necessarily reduced by using a regular grid, fixed scales, and fixed aspect ratios. In most cases the number of locations to visit remains huge, so much that alternative restrictions need to be imposed. The classifier is simplified and the appearance model needs to be quick rather than strong. Furthermore, a uniform sampling yields many boxes for which it is immediately clear that is is not supportive of an object; the content of the box is useless for recognition altogether. Rather then sampling locations blindly using an exhaustive search, a key question is: Can we steer the sampling by a data-driven analysis?

In this chapter we aim to combine the best of the intuitions of segmentation and exhaustive search and propose a data-driven structured sampling. Inspired by
bottom-up segmentation, we aim to exploit the structure of the image to sample object locations. Inspired by exhaustive search, we aim to capture all possible object locations. Therefore, instead of using a single sampling technique, we aim to diversify the sampling techniques to account for as many image conditions as possible. Specifically, we use a data-driven grouping-based sampling strategy where we increase diversity by using a variety of complementary grouping criteria and a variety of complementary colour spaces with different invariance properties. The set of locations is obtained by combining the locations of these complementary partitionings. Our goal is to generate a class-independent, data-driven, structured sampling that generates a small set of high-quality object locations.

Our application domain of structured sampling is object recognition. We therefore evaluate on most commonly used dataset for this purpose, the Pascal VOC detection challenge which consists of 20 object classes. The size of this dataset yields computational constraints for our structured sampling. Furthermore, the use of this dataset means that the quality of locations is mainly evaluated in terms of bounding boxes. However, structured sampling applies to regions as well and is also applicable to concepts such as “grass”.

In this chapter we propose structured sampling for object recognition. Our main research questions are: (1) Can we create a diverse set of complementary sampling strategies? (2) Can we generate a small set of high-quality locations? (3) Can we use the reduction in locations to use more powerful classifiers and appearance models? (4) What is the desired quality of locations for object recognition?

4.2 Related Work

We keep the related work limited to the domain of object recognition. We divide related work into three categories: Exhaustive search, segmentation, and other sampling strategies that do not fall in either category.

4.2.1 Exhaustive Search

As an object can be located at any position and scale in the image, it is natural to search everywhere [20,42,108]. However, the visual search space is huge, making an exhaustive search computationally expensive. This imposes constraints on the evaluation cost per location and/or the number of locations considered. Hence most of these sliding window techniques use a coarse search grid and fixed aspect ratios, using weak classifiers and economic image features such as HOG [20,42,108]. This method is often used as a preselection step in a cascade of classifiers [42,108].

Related to the sliding window technique is the highly successful part-based object localisation method of Felzenszwalb et al. [27]. Their method also performs an exhaustive search using a linear SVM and HOG features. However, they search for objects and object parts, whose combination results in an impressive object detection performance.

Lampert et al. [49] proposed using the appearance model to guide the search. This both alleviates the constraints of using a regular grid, fixed scales, and fixed aspect ratio, while at the same time reduces the number of locations visited. This is done by directly searching for the optimal window within the image using
structured sampling for object recognition

While they obtain impressive results for linear classifiers, [2] found that for non-linear classifiers the method in practice still visits over a 100,000 windows per image.

Instead of a blind exhaustive search or a branch and bound search, we propose structured sampling. We use the underlying image structure to generate object locations. In contrast to the discussed methods, this yields a completely class-independent set of locations. Furthermore, because we do not use a fixed aspect ratio, our method is not limited to objects but should be able to find stuff like “grass” and “sand” as well (this also holds for [49]). Finally, we hope to generate fewer locations, which should make the problem easier as the variability of samples becomes lower. And more importantly, it frees up computational power which can be used for stronger machine learning techniques and more powerful appearance models.

Figure 31: Two examples of our structured sampling showing the necessity of different scales. On the left we find many objects at different scales. On the right we necessarily find the objects at different scales as the girl is contained by the tv.

4.2.2 Segmentation

Both Carreira et al. [13] and Endres and Hoiem [22] propose to generate a set of class independent object hypotheses using segmentation. Both methods generate multiple foreground/background segmentations, learn to predict the likelihood that a foreground segment is a complete object, and use this to rank the segments. Both algorithms show a promising ability to accurately delineate objects within images, confirmed by [52] who achieve state-of-the-art results on pixel-wise image classification using [13]. As common in segmentation, both methods rely on a single strong algorithm for identifying good regions. They obtain a variety of locations by using many randomly initialised foreground and background seeds. In contrast, we explicitly deal with a variety of image conditions by using different grouping criteria and different representations. This means a lower computational investment as we do not have to invest in the single best segmentation strategy, such as using the excellent yet expensive contour detector of [3]. Furthermore, as we deal with different image conditions separately, we expect our locations to have a more consistent quality. Finally, our structured sampling paradigm dictates that the most interesting question is not how our regions compare to [13, 22], but rather how they can complement each other.
Gu et al. [41] address the problem of carefully segmenting and recognizing objects based on their parts. They first generate a set of part hypotheses using a grouping method based on Arbelaez et al. [3]. Each part hypothesis is described by both appearance and shape features. Then an object is recognized and carefully delineated by using its parts, achieving good results for shape recognition. In their work, the segmentation is hierarchical and yields segments at all scales. However, they use a single grouping strategy whose power of discovering parts or objects is left unevaluated. In this work, we use multiple complementary strategies to deal with as many image conditions as possible. We include the locations generated using [3] in our evaluation.

4.2.3 Other Sampling Strategies

Alexe et al. [2] address the problem of the large sampling space of an exhaustive search by proposing to search for any object, independent of its class. In their method they train a classifier on the object windows of those objects which have a well-defined shape (as opposed to stuff like “grass” and “sand”). Then instead of a full exhaustive search they randomly sample boxes to which they apply their classifier. The boxes with the highest “objectness” measure serve as a set of object hypotheses. This set is then used to greatly reduce the number of windows evaluated by class-specific object detectors.

Another strategy is to use visual words of the Bag-of-Words model to predict the object location. Vedaldi et al. [104] use jumping windows [16], in which the relation between individual visual words and the object location is learned to predict the object location in new images. Maji and Malik [55] combine multiple of these relations to predict the object location using a Hough-transform, after which they randomly sample windows close to the Hough maximum. In contrast to learning, we use the image structure to sample a set of class-independent object hypotheses.

4.3 Structured Sampling

In this section we detail our structured sampling algorithm for object recognition and present a variety of diversification strategies to deal with as many image conditions as possible. A structured sampling algorithm is subject to the following design considerations:

**capture all scales.** Objects can occur at any scale within the image. Furthermore, some objects have less clear boundaries than other objects. Therefore, in structured sampling all object scales have to be taken into account, as illustrated in Figure 31. This is most naturally achieved by using an hierarchical algorithm.

**diversification.** There is no single optimal strategy to group regions together. As observed earlier in Figure 30, regions may form an object because only of colour, only of texture, or because parts are enclosed. Furthermore, lighting conditions such as shading and the colour of the light may influence how regions form an object. Therefore instead of a single strategy which works well in most cases, we want to have a diverse set of strategies to deal with all cases.
fast to compute. The goal of structured sampling is to yield a set of possible object locations for use in a practical object recognition framework. The creation of this set should not become a computational bottleneck, hence our algorithm should be reasonably fast.

4.3.1 Structured Sampling by Hierarchical Grouping

We take a hierarchical grouping algorithm to form the basis of our structured sampling. Bottom-up grouping is a popular approach to segmentation [17, 28], hence we adapt it for structured sampling. Because the process of grouping itself is hierarchical, we can naturally generate locations at all scales by continuing the grouping process until the whole image becomes a single region. This satisfies the condition of capturing all scales.

As regions can yield richer information than pixels, we want to use region-based features whenever possible. To get a set of small starting regions which ideally do not span multiple objects, we use the fast method of Felzenszwalb and Huttenlocher [28], which [3] found well-suited for such purpose.

Our grouping procedure now works as follows. We first use [28] to create initial regions. Then we use a greedy algorithm to iteratively group regions together: First the similarities between all neighbouring regions are calculated. The two most similar regions are grouped together, and new similarities are calculated between the resulting region and its neighbours. The process of grouping the most similar regions is repeated until the whole image becomes a single region. The general method is detailed in Algorithm 1.

**Algorithm 1: Hierarchical Grouping Algorithm**

**Input:** (colour) image  
**Output:** Set of object location hypotheses L

1. Obtain regions $R = \{r_1, \ldots, r_n\}$ using initial grouping  
2. Initialise similarity set $S = \emptyset$

   **foreach** Neighbouring region pair $(r_i, r_j)$ **do**
   
   1. Calculate similarity $s(r_i, r_j)$
   2. $S = S \cup s(r_i, r_j)$

3. **while** $S \neq \emptyset$ **do**

   1. Get highest similarity $s(r_t, r_j) = \max(S)$
   2. Merge corresponding regions $r_t = r_i \cup r_j$
   3. Remove similarities regarding $r_t$: $S = S \setminus s(r_t, r_s)$
   4. Remove similarities regarding $r_j$: $S = S \setminus s(r_s, r_j)$
   5. Calculate similarity set $S_t$ between $r_t$ and its neighbours
   6. $S = S \cup S_t$
   7. $R = R \cup r_t$

   extract object location boxes L from all regions in R

For the similarity $s(r_i, r_j)$ between region $r_i$ and $r_j$ we want a variety of complementary measures under the constraint that they are fast to compute. In effect, this means that the similarities should be based on features that can be propagated through the hierarchy, i.e. when merging region $r_i$ and $r_j$ into $r_t$, the
features of region \( r_t \) need to be calculated from the features of \( r_i \) and \( r_j \) without accessing the image pixels.

### 4.3.2 Diversification Strategies

The second design criterion for structured sampling is to diversify the sampling and create a set of complementary strategies whose locations are combined afterwards. We diversify our structured sampling (1) by using a variety of colour spaces with different invariance properties, (2) by using different similarity measures \( s_{ij} \), and (3) by varying our starting regions.

**Complementary Colour Spaces.** To account for different scene and lighting conditions, we perform hierarchical grouping in a variety of colour channels with a range of invariance properties. Specifically, we consider RGB, HSV, Opponent Colour space \( \text{Lab} \), the \( \text{rg} \) channels of normalized RGB plus intensity denoted as \( \text{rgI} \), \( \text{rgb} \) or normalized RGB, C [36] which is an Opponent Colour space where intensity is divided out, the Hue \( H \) from HSV, and finally the intensity \( I \).

All colour spaces have different invariance properties: RGB has no invariance properties. Normalized RGB or \( \text{rgb} \) is insensitive to light intensity changes, insensitive to shadow and shading edges but still sensitive to highlights. \( \text{rgI} \) is therefore mildly sensitive to light intensity changes, mildly sensitive to shadow and shading edges, and sensitive to highlights. \( \text{Lab} \) is an opponent colour space and is insensitive to highlight edges but sensitive to edges from shadows and shading. \( C \) is insensitive to highlight edges and largely insensitive to edges from shadows and shading. In addition, it is invariant to intensity. For HSV, the \( H \) and \( S \) channels are invariant to light intensity changes. In addition, \( H \) is invariant to highlights. The intensity, or \( V \) channel is invariant to light colour changes. The HSV space as a whole is not invariant, but is less sensitive to said properties than RGB.

In this chapter we always use a single colour space throughout the algorithm, meaning that both the initial grouping of [28] and the subsequent algorithm are performed in this colour space.

**Complementary Similarity Measures.** We define four complementary, fast-to-compute similarity measures. These measures are all in range \([0, 1]\) which facilitates combinations of these measures.

\( s_{\text{colour}}(r_i, r_j) \) measures colour similarity. Specifically, for each region we obtain one-dimensional colour histograms for each colour channel using 25 bins, which we found to work well. This leads to a colour histogram \( C_i = \{c^1_i, \ldots, c^n_i\} \) for each region \( r_i \) with dimensionality \( n = 75 \) when three colour channels are used. The colour histograms are normalised using the \( L_1 \) norm. Similarity is measured using the histogram intersection:

\[
s_{\text{colour}}(r_i, r_j) = \sum_{k=1}^{n} \min(c^k_i, c^k_j). \tag{4.1}
\]

The colour histograms can be efficiently propagated through the hierarchy by \( C_t = (\text{size}(r_i) \times C_i + \text{size}(r_j) \times C_j) / \text{size}(r_i) + \text{size}(r_j) \). The size of a resulting region is simply the sum of its constituents: \( \text{size}(r_t) = \text{size}(r_i) + \text{size}(r_j) \).
structured sampling for object recognition

$s_{\text{texture}}(r_i, r_j)$ measures texture similarity. We take Gaussian derivatives in eight orientations using $\text{sigma} = 1$ for each colour channel. For each orientation for each colour channel we extract a histogram using a bin size of 10. This leads to a texture histogram $T_i = \{c^k_i, \ldots, c^n_i\}$ for each region $r_i$ with dimensionality $n = 240$ when three colour channels are used. Texture histograms are normalised using the $L_1$ norm. Similarity is measured using the histogram intersection:

$$s_{\text{texture}}(r_i, r_j) = \sum_{k=1}^{n} \min(c^k_i, c^k_j). \quad (4.2)$$

Texture histograms are efficiently propagated through the hierarchy in the same way as the colour histograms.

$s_{\text{size}}$ encourages small regions to merge early. This forces regions in $S$, i.e., regions which have not yet been merged, to be of similar sizes throughout the algorithm. This is desirable because it ensures that object locations at all scales are created at all parts of the image. For example, it prevents a single region from gobbling up all other regions one by one, yielding all scales only at the location of this growing region and nowhere else. $s_{\text{size}}(r_i, r_j)$ is defined as the fraction of the image that $r_i$ and $r_j$ jointly occupy:

$$s_{\text{size}}(r_i, r_j) = \frac{\text{size}(r_i) + \text{size}(r_j)}{\text{size}(im)}, \quad (4.3)$$

where $\text{size}(im)$ denotes the size of the image in pixels.

$s_{\text{fill}}(r_i, r_j)$ measures how well region $r_i$ and $r_j$ fit into each other. The idea is to fill gaps: if $r_i$ is contained in $r_j$ it is logical to merge these first in order to avoid any holes. On the other hand, if $r_i$ and $r_j$ are hardly touching each other they will likely form a strange region and should not be merged. To keep the measure fast, we use only the size of the regions and of the containing boxes. Specifically, we define $BB_{ij}$ to be the tight bounding box around $r_i$ and $r_j$. Now $s_{\text{fill}}(r_i, r_j)$ is the fraction of the image contained in $BB_{ij}$ which is not covered by the regions of $r_i$ and $r_j$:

$$\text{fill}(r_i, r_j) = \frac{\text{size}(BB_{ij}) - \text{size}(r_i) - \text{size}(r_i)}{\text{size}(im)} \quad (4.4)$$

Note that this measure can be efficiently calculated by keeping track of the bounding boxes around each region, as the bounding box around two regions can be easily derived from these.

In this chapter our final similarity measure is a combination of the above four:

$$s(r_i, r_j) = a_1 s_{\text{colour}}(r_i, r_j) + a_2 s_{\text{texture}}(r_i, r_j) + a_3 s_{\text{size}}(r_i, r_j) + a_4 s_{\text{fill}}(r_i, r_j), \quad (4.5)$$

where $a_i \in \{0, 1\}$ denotes if the similarity measure is used or not. As we aim to diversify our strategies, we do not consider any weighted similarities.
Figure 32: The training procedure of our object recognition pipeline. As positive learning examples we use the ground truth. As negatives we use examples that have a 20-50% overlap with the positive examples. We iteratively add hard negatives using a retraining phase.

**Complementary Starting Regions.** A third diversification strategy is varying the complementary starting regions. To the best of our knowledge, the method of [28] is the fastest, publicly available algorithm that yields high quality starting locations. We could not find any other algorithm with similar computational efficiency so we stick to this work. Instead, note that different starting regions are (already) obtained by varying the colour space. In addition, we vary the threshold parameter \( k \) in [28].

4.3.3 Combining Locations

In this chapter we combine the object hypotheses of several hierarchical groupings. Ideally, we want to order the object hypotheses in such a way that the locations which are most likely an object come first. This enables one to find a good trade-off between the quality and quantity of the resulting object hypothesis set, depending on the computational efficiency of the subsequent feature extraction and classification method.

We choose to order the hypotheses of multiple groupings based on the order in which they are created in the hierarchy. However, as we combine up to 80 different groupings, such order would too heavily emphasize large regions. To get more diversity in the ranking, we therefore include some randomness. Let \( p^j_i \) be position \( i \) in the hierarchy of the \( j \)th grouping which we combine, where \( i = 1 \) represents the top of the hierarchy (whose corresponding location is the full image). We now calculate the position value \( v^j_i \) as RND × \( i \), where RND is a random number in range \([0, 1]\). The final ranking is obtained by ordering the regions using \( v^j_i \).

After the final ranking is obtained, when we use bounding box locations we filter out lower ranked duplicates. Therefore duplicate boxes, which are likely to be visual coherent as they emerge from different grouping strategies, have a better chance of a high rank. Note, however, that ideally all locations generated by the structured sampling should be examined.

4.4 Object Recognition System

This chapter uses the locations generated by our structured sampling for object recognition. This section details our framework for object recognition.
Two types of features are dominant in object recognition: histograms of oriented gradients (HOG) [20] and bag-of-words [18, 82]. HOG has been shown to be successful in combination with the part-based model by Felzenszwalb et al. [27]. However, as they use an exhaustive search, HOG features in combination with a linear classifier is the only feasible choice from a computational perspective. In contrast, our structured sampling enables the use of more expensive and potentially more powerful features. Therefore we use bag-of-words for object recognition [42, 49, 104]. However, we use a more powerful (and expensive) implementation than [42, 49, 104] by employing multiple colour spaces and a finer spatial pyramid division [51].

Specifically we sample descriptors at each pixel on a single scale ($\sigma = 1.2$). Using software from [103], we extract SIFT [53] and two recommended colour SIFTs from [103], OpponentSIFT and RGB-SIFT. We use a visual codebook of size 4,000 and a spatial pyramid with 4 levels. This gives a total feature vector length of 360,000. In image classification, features of this size are already used [70, 117]. Because a spatial pyramid results in a coarser spatial subdivision than the cells which make up a HOG descriptor, our features contain less information about the specific spatial layout of the object. Therefore, HOG is better suited for rigid objects and our features are better suited for deformable object types.

As classifier we employ a Support Vector Machine with a histogram intersection kernel using the Shogun Toolbox [89]. To apply the trained classifier, we use the fast, approximate classification strategy of [54], which was shown to work well for Bag-of-Words by [99].

Our training procedure is illustrated in Figure 32. The initial positive examples consist of all ground truth object windows. As initial negative examples we select from all object locations generated by our structured sampling that have an overlap of 20\% to 50\% with a positive example. To avoid near-duplicate negative examples, a negative example is excluded if it has more than 70\% overlap with another negative. To keep the number of initial negatives per class below 20,000, we randomly drop half of the negatives for the classes car, cat, dog and person. This selection of training examples gives reasonably good initial classification models.

Then we enter a retraining phase to iteratively add hard negative examples (e.g. [27]): We apply the learned models to the training set using the locations generated by our structured sampling. For each negative image we add the highest scoring location. As our initial training set already yields good models, our models converge in only two iterations.

For the test set, the final model is applied to all locations generated by our structured sampling. The windows are sorted by classifier score while windows which have more than 30\% overlap with a higher scoring window are considered near-duplicates and are removed.

4.5 Evaluation

In this section we evaluate the quality of our structured sampling. We divide our experiments in four parts, each spanning a separate subsection:

diversification strategies. We experiment with a variety of colour spaces, similarity measures, and thresholds of the initial regions, all which were detailed in Section 4.3.2. We seek a trade-off between the number of generated
object hypotheses, computation time, and the quality of object locations. We do this in terms of bounding boxes. This results in a selection of complementary techniques which together serve as our final structured sampling method.

**quality of locations.** In this section we test the quality of the object location hypotheses resulting from structured sampling.

**object recognition.** In this section we use the locations of our structured sampling in the Object Recognition framework detailed in Section 4.4. We evaluate performance on the Pascal VOC detection challenge.

**an upper bound of location quality.** We investigate how well our object recognition framework performs when using an object hypothesis set of “perfect” quality. How does this compare to the locations that our structured sampling generates?

To evaluate the quality of our object hypotheses we define the Average Best Overlap (ABO) and Mean Average Best Overlap (MABO) scores. To calculate the Average Best Overlap for a specific class $c$, we calculate the best overlap between each ground truth annotation $g^c_i \in G^c$ and the object hypotheses $L$ generated for the corresponding image, and average:

$$\text{ABO} = \frac{1}{|G^c|} \sum_{g^c_i \in G^c} \max_{l_j \in L} \text{Overlap}(g^c_i, l_j).$$ \hspace{1cm} (4.6)

The Overlap score is taken from [23] and measures the area of the intersection of two regions divided by its union:

$$\text{Overlap}(g^c_i, l_j) = \frac{\text{area}(g^c_i) \cap \text{area}(l_j)}{\text{area}(g^c_i) \cup \text{area}(l_j)}.$$ \hspace{1cm} (4.7)

Analogously to Average Precision and Mean Average Precision, Mean Average Best Overlap is now defined as the mean ABO over all classes.

Other work often uses the recall derived from the Pascal Overlap Criterion to measure the quality of the boxes [2, 42, 104]. This criterion considers an object to be found when the Overlap of Equation 4.7 is larger than 0.5. However, in many of our experiments we obtain a recall between 95% and 100% for most classes, making this measure too insensitive for most experiments in this chapter. However, we also use this measure to enable comparison with other work.

To avoid overfitting, we perform the diversification strategies experiments on the Pascal VOC 2007 train+val set. Other experiments are done on the Pascal VOC 2007 test set. Additionally, our object recognition system is benchmarked on the Pascal VOC 2010 detection challenge, using the independent evaluation server.

**4.5.1 Diversification Strategies**

In this section we evaluate a variety of strategies to obtain good quality object location hypotheses using a reasonable number of boxes computed within a reasonable amount of time.
Flat versus Hierarchy

In the description of our method we claim that using a full hierarchy is more natural than using multiple flat partitionings by changing a threshold. In this section we test whether the use of a hierarchy also leads to better results. We therefore compare the use of [28] with multiple thresholds against our proposed algorithm. Specifically, we perform both algorithms strategies in RGB colour space. For [28], we vary the threshold from $k = 50$ to $k = 1000$ in steps of 50. This range captures both small and large regions. Additionally, as a special type of threshold, we include the whole image as an object location because quite a few images contain a single large object only. Furthermore, we also take a coarser range from $k = 50$ to $k = 950$ in steps of 100. For our algorithm, to create initial regions we use a threshold of $k = 50$, ensuring that both strategies have an identical smallest scale. Additionally, as we generate fewer regions, we combine results using $k = 50$ and $k = 100$. As similarity measure $S$ we use the addition of all four similarities that we defined. Results are in Table 4.

<table>
<thead>
<tr>
<th></th>
<th>MABO</th>
<th># windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>[28] $k = 50, 100, \cdots, 1000$</td>
<td>0.673</td>
<td>597</td>
</tr>
<tr>
<td>[28] $k = 50, 150, \cdots, 950$</td>
<td>0.659</td>
<td>387</td>
</tr>
<tr>
<td>Ours $k = 50$</td>
<td>0.676</td>
<td>395</td>
</tr>
<tr>
<td>Ours $k = 50, 100$</td>
<td>0.719</td>
<td>625</td>
</tr>
</tbody>
</table>

Table 4: Comparison of multiple flat partitionings against hierarchical partitionings for finding good object hypotheses in terms of boxes. At a comparable number of object locations, using a hierarchy yields a higher MABO score. Furthermore, note that the increase in MABO is much higher when combining two hierarchical groupings then when using extra thresholds in the flat partitioning.

As can be seen, the trade-off between the object hypotheses quality is better for our hierarchical grouping then for the multiple flat partitionings: At a similar number of regions, our MABO score is consistently higher. Moreover, the increase in MABO achieved by combining two hierarchical groupings is much higher than the increase achieved by adding extra thresholds for the flat partitionings. We conclude that using all locations from a hierarchical grouping is not only more natural but also more effective than using multiple flat partitionings.

Individual Diversification Strategies

In this chapter we propose three diversification strategies to obtain good quality object hypotheses: varying the colour space, varying the similarity measures, and varying the thresholds to obtain the starting regions. This section investigates the influence of each strategy. As basic settings we use RGB colour space, the combination of all four similarity measures, and threshold $k = 50$. Each time we vary a single parameter. Results are given in Table 5.

We start examining the combination of similarity measures on the left part of Table 5. Looking first at colour, texture, size, and fill individually, we see that the texture similarity performs worst with a MABO of 0.581, while the other measures range between 0.63 and 0.64. While the texture similarity yields less object locations, at 300 locations the other similarity measures still yield a MABO higher
Table 5: Mean Average Best Overlap for box-based object hypotheses when using combinations of the similarity measures (C)olour, (T)exture, (S)ize, and (F)ill. As can be seen, combinations work often better than the individual measures. Texture seems particularly weak.

Looking at variations in the colour space in the top-right of Table 5, we observe large differences in results, ranging from a MABO of 0.615 with 125 locations for the C colour space to a MABO of 0.693 with 463 locations for the HSV colour space. We note that Lab-space has a particularly good MABO score of 0.690 using only 328 boxes. But again, the order of each hierarchy is effective: using the first 328 boxes of HSV colour space yields 0.690 MABO, while using the first 100 boxes yields 0.647 MABO. This confirms that when comparing single strategies the MABO score is indicative for the trade-off between quality and quantity of the object hypotheses set.

Experiments on the thresholds of [28] to generate the starting regions show, in the bottom-right of Table 5, that a lower initial threshold results in a higher MABO using more object locations.

Combinations of Diversification Strategies

We aim to generate many complementary grouping strategies in order to get a good quality set of object locations. However, an full search for the best combination is too expensive computationally. Instead, we perform three greedy searches by taking our grouping algorithm with basic settings and vary only the colour spaces, only the similarity measures, and only the thresholds, analogous to our previous experiments. Earlier we have observed that the MABO score of all the object locations of a given grouping is also representative for the trade-off between
<table>
<thead>
<tr>
<th>Strategy</th>
<th>Ordering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colour Spaces</td>
<td>HSV, Lab, rgl, H, I, rgb, C, RGB</td>
</tr>
<tr>
<td>Thresholds</td>
<td>50, 100, 300, 150, 550, 200, 400, 900, 250, 650, 350, 450, 800, 500, 1000, 600, 700, 850, 750, 950</td>
</tr>
</tbody>
</table>

Table 6: The order of the greedy algorithm for using a variety of colour spaces, similarity measures, and thresholds to create the starting regions.

the number of object locations and their quality. Furthermore, as the difference in the number of locations is not very high and calculating each grouping costs time, using the MABO score of a single grouping is good indicator for the trade-off between quantity, quality, and computational efficiency of generating object location hypotheses. We use therefore the MABO score as optimization criterion. After the greedy searches, we take the top colour space, similarity measures, and thresholds and combine the groupings using all possible permutations, which together forms the final structured sampling method.

Figure 33: Progression of MABO and the number of object boxes generated during the greedy search. Colour is the best strategy for obtaining a high quality set of boxes.

Figure 34: Evaluation of our ordering strategy. As can be seen, our ordering strategy is more effective than simply adding more hierarchical groupings.

Table 6 shows the greedy choices made for selecting colour spaces, similarity measures, and thresholds. Figure 33 shows how each additional grouping increases the MABO score during the greedy search. We first observe that varying the colour space is most effective at yielding a higher MABO score. This should not be too surprising given that the colour space affects both the starting regions and the subsequent grouping of the regions. Varying the thresholds for the starting regions and varying the similarity measures of the grouping seem about equally effective.

Looking at the greedy choices for the thresholds, the algorithm favours the small thresholds which result in more regions. For the similarity measures, after the first two combinations which include all and three measures, the algorithm favours only Fill and only Size, followed by combinations of two measures. This variety shows that complementary strategies are preferred. This is further con-
Table 7: Combination strategies resulting from the greedy search, which form the final structured sampling methods.

<table>
<thead>
<tr>
<th>Version</th>
<th>Diversification Strategies</th>
<th>MABO</th>
<th># windows</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Grouping</td>
<td>HSV</td>
<td>0.693</td>
<td>362</td>
<td>2.01</td>
</tr>
<tr>
<td></td>
<td>C+R+S+F</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>k = 100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structured Sampling</td>
<td>HSV, Lab</td>
<td>0.799</td>
<td>2147</td>
<td>8.77</td>
</tr>
<tr>
<td>Fast</td>
<td>C+T+S+F, T+S+F</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>k = 50, 100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structured Sampling</td>
<td>HSV, Lab, rgI, H, I</td>
<td>0.878</td>
<td>10,108</td>
<td>27.7</td>
</tr>
<tr>
<td>Quality</td>
<td>C+T+S+F, T+S+F, F, S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>k = 50, 100, 150, 300</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In this section we evaluate our structured sampling algorithms in terms of both Average Best Overlap and the number of locations on the Pascal VOC 2007 test set. We first evaluate box-based locations and afterwards briefly evaluate region-based locations.
Table 8: Evaluation of our structured sampling and several algorithms to find a good set of potential object locations.

<table>
<thead>
<tr>
<th>method</th>
<th>recall</th>
<th>MABO</th>
<th># windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arbelaez et al. [3]</td>
<td>0.752</td>
<td>0.649</td>
<td>418</td>
</tr>
<tr>
<td>Alexe et al. [2]</td>
<td>0.824</td>
<td>0.747</td>
<td>10,000</td>
</tr>
<tr>
<td>Harzallah et al. [42]</td>
<td>0.830</td>
<td>-</td>
<td>200 per class</td>
</tr>
<tr>
<td>Carreira et al. [13]</td>
<td>0.879</td>
<td>0.770 ± 0.084</td>
<td>517</td>
</tr>
<tr>
<td>Endres et al. [22]</td>
<td>0.912</td>
<td>0.791 ± 0.082</td>
<td>790</td>
</tr>
<tr>
<td>Felzenszwalb et al. [27]</td>
<td>0.933</td>
<td>0.829 ± 0.052</td>
<td>100,000 per class</td>
</tr>
<tr>
<td>Vedaldi et al. [104]</td>
<td>0.940</td>
<td>-</td>
<td>10,000 per class</td>
</tr>
<tr>
<td>Single Grouping</td>
<td>0.840</td>
<td>0.690</td>
<td>289</td>
</tr>
<tr>
<td>SS “Fast”</td>
<td>0.980</td>
<td>0.804 ± 0.046</td>
<td>2,134</td>
</tr>
<tr>
<td>SS “Quality”</td>
<td>0.991</td>
<td>0.879 ± 0.039</td>
<td>10,097</td>
</tr>
</tbody>
</table>

Box-based Locations

We compare with the sliding window search of [42], the sliding window search of [27] using the window ratio’s of their models, the jumping windows of [104], the “objectness” boxes of [2], the boxes around the hierarchical segmentation algorithm of [3], the boxes around the regions of [22], and the boxes around the regions of [13]. From these algorithms, only [3] is not designed for finding object locations. Yet [3] is one of the best contour detectors publicly available, and results in a natural hierarchy of regions. We include it in our evaluation to see if this algorithm designed for segmentation also performs well on finding good object locations. Furthermore, [13, 22] are designed to find good object regions rather then boxes. Results are shown in Table 8 and Figure 35.

Figure 35: Trade-off between quality and quantity of the object hypotheses in terms of bounding boxes. The dashed lines are for those methods whose quantity is expressed is the number of boxes per class.

As can be seen in Table 8, our “Fast” and “Quality” structured sampling methods yield a close to optimal recall of 98% and 99% respectively. In terms of MABO, we achieve 0.804 and 0.879 respectively. To appreciate what a Best Overlap of 0.879 means, Figure 36 shows for bike, cow, and person an example location
which has an overlap score between 0.874 and 0.884. This illustrates that our structured sampling yields high quality object locations.

Furthermore, note that the standard deviation of our MABO scores is relatively low: 0.046 for the fast structured sampling, and 0.039 for the quality structured sampling. This shows that structured sampling is robust to difference in object properties, and also to image condition often related with specific objects (one example is indoor/outdoor lighting).

If we compare with other algorithms, the second highest recall is at 0.940 and is achieved by the jumping windows [104] using 10,000 boxes per class. As we do not have the exact boxes, we were unable to obtain the MABO score. This is followed by the exhaustive search of [27] which achieves a recall of 0.933 and a MABO of 0.829 at 100,000 boxes per class. This is significantly lower than our method while using at least a factor of 10 more object locations.

Note furthermore that the segmentation methods of [13, 22] have a relatively high standard deviation. This illustrates that a single strategy can not work equally well for all classes. Instead, using multiple complementary strategies leads to more stable and reliable results.

If we compare the segmentation of Arbelaez [3] with a single grouping of our method, we see that they achieve a recall of 0.752 and a MABO of 0.649 at 418 boxes, while we achieve 0.875 recall and 0.698 MABO using 286 boxes. This suggests that a good segmentation algorithm does not automatically result in good object locations in terms of bounding boxes.

Figure 35 explores the trade-off between the quality and quantity of the object hypotheses. In terms of recall, our “Fast” method is superior to all other methods. The method of [42] seems competitive for the 200 locations they use, but in their method the number of boxes is per class while for our method the same boxes are used for all classes. In terms of MABO, both the object hypotheses generation method of [13] and [22] have a good quantity/quality trade-off for the up to 790 object-box locations per image they generate. However, these algorithms are computationally at least 8 times more expensive. Furthermore, it is unclear how the MABO score can progress as the number of regions increase and at what computational cost.

Figure 37 shows for several methods the Average Best Overlap per class. As can be seen, the exhaustive search of [27] which uses 10 times more locations which are class specific, performs similar to our method for the classes bike, table, chair, and sofa, for the other classes our method yields the best score. In general, the classes with the highest scores are cat, dog, horse, and sofa, which are easy largely
because the instances in the dataset tend to be big. The classes with the lowest scores are bottle, person, and plant, which are difficult because instances tend to be small. Nevertheless, cow, sheep, and tv are not bigger than person and yet can be found quite well by our algorithm.

To summarize, structured sampling is very effective in finding a high quality set of object hypotheses using a limited number of boxes, where the quality is reasonable consistent over the object classes. The methods of \[13\] and \[22\] have a similar quality/quantity trade-off for up to 790 object locations. However, they have more variation over the object classes. Furthermore, they are much more expensive to compute which is a problem for current dataset sizes for object recognition. In general, we conclude that structure sampling yields the best quality object locations at 0.879 MABO while using a reasonable number of 10,097 class-independent object locations.

**Region-based Locations**

In this section we examine how well the regions that our structures sampling generates captures object locations. We do this on the segmentation part of the Pascal VOC 2007 test set. We compare with the segmentation of \[3\] and with the object hypothesis regions of both \[13, 22\]. Table 9 shows the results. Note that the number of regions is larger than the number of boxes as there are almost no duplicates.

The object regions of both \[13, 22\] are of similar quality as our “Fast” structured sampling, 0.665 MABO and 0.679 MABO respectively where our “Fast” sampling yields 0.666 MABO. While \[13, 22\] use fewer regions these algorithms are respectively 54 and 28 times computationally more expensive. Our “Quality” structured sampling generates 22,491 regions and is respectively 15 and 8 times faster then \[13, 22\], and has by far the highest score of 0.730 MABO.
4.5 Evaluation

<table>
<thead>
<tr>
<th>method</th>
<th>recall</th>
<th>MABO</th>
<th># regions</th>
<th>time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3]</td>
<td>0.539</td>
<td>0.540 ± 0.117</td>
<td>1122</td>
<td>64</td>
</tr>
<tr>
<td>[22]</td>
<td>0.813</td>
<td>0.679 ± 0.108</td>
<td>2167</td>
<td>226</td>
</tr>
<tr>
<td>[13]</td>
<td>0.782</td>
<td>0.665 ± 0.118</td>
<td>697</td>
<td>432</td>
</tr>
<tr>
<td>Single Grouping</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS “Fast”</td>
<td>0.576</td>
<td>0.548 ± 0.078</td>
<td>678</td>
<td>2</td>
</tr>
<tr>
<td>SS “Quality”</td>
<td>0.829</td>
<td>0.666 ± 0.089</td>
<td>3574</td>
<td>8</td>
</tr>
<tr>
<td>[13, 22] + “Fast”</td>
<td>0.896</td>
<td>0.737 ± 0.098</td>
<td>6,438</td>
<td>666</td>
</tr>
<tr>
<td>[13, 22] + “Quality”</td>
<td>0.920</td>
<td>0.758 ± 0.096</td>
<td>25,355</td>
<td>686</td>
</tr>
</tbody>
</table>

Table 9: Comparison of algorithms to find a good set of potential object locations in terms of regions.

Figure 38 visualises the trade-off between the number of object locations and the MABO. While [13, 22] have a better trade-off, their order results from classifying each region on its likelihood of belonging to any of the 20 Pascal VOC object classes, while [22] additionally learns the appearance of background classes such as grass and road using the LabelMe dataset [76]. Both [13, 22] discard all non-object regions. In contrast, our algorithm has no notion of any object, therefore regions are ranked only based on the grouping order, and background regions are retained. Depending on the application, background regions may be useful.

Figure 39 shows the Average Best Overlap of the regions per class. For all classes except bike, our structured sampling consistently has relatively high ABO scores. The performance for bike is disproportionately lower for region-locations instead of object-locations, because bike is a wire-frame object and hence very difficult to accurately delineate.

If we compare our method to others, the method of [22] is better for train, for the other classes our “Quality” method yields similar or better scores. For bird, boat, bus, chair, person, plant, and tv scores are 0.05 ABO better. For car we obtain 0.12 higher ABO and for bottle even 0.17 higher ABO. Looking at the variation in ABO scores in table 9, we see that structured sampling has a slightly lower variation than the other methods: 0.093 MABO for “quality” and 0.108 for [22]. However, this score is biased because of the wire-framed bicycle: without bicycle the difference becomes more apparent. The standard deviation for the “quality” structured sampling becomes 0.058, and 0.100 for [22]. Again, this shows that by relying on multiple complementary strategies instead of a single strategy yields more stable results.

Now as we do structured sampling, rather than pitting methods against each other, it is more interesting to see how they can complement each other. As both [13, 22] have a very different algorithm, the combination should prove effective according to our diversification hypothesis. Indeed, as can be seen in the lower part of Table 9, combination with our “Fast” structured sampling leads to 0.737 MABO at 6,438 locations. This is a higher MABO using less locations than our “quality” structured sampling. A combination of [13, 22] with our “quality” sampling leads to 0.758 MABO at 25,355 locations. This is a good increase at only a modest extra number of locations.

To conclude, structured sampling is highly effective for generating object locations in terms of regions. The use of a variety of strategies makes it robust against
structured sampling for object recognition

Figure 38: Trade-off between quality and quantity of object region hypotheses for a variety of methods. The trade-off of the other methods are better but their hypotheses order comes from classification while our order comes from the grouping itself. Our “Quality” method achieves a MABO well beyond the other methods.

Figure 39: Comparison of the Average Best Overlap Scores per class between our method and others. Except for train, our “Quality” method consistently yields better Average Best Overlap scores.

various image conditions as well as the object class. The combination of [13], [22] and our grouping algorithms into a single structured sampling showed promising improvements. Given these improvements, and given that there are many more different partitioning algorithms out there to use in a structured sampling, it will be interesting to see how far structured sampling can still go in terms of computational efficiency, number of object locations, and the quality of object locations.

4.5.3 Object Recognition

In this section we will evaluate our structured sampling strategy for object recognition using the Pascal VOC detection task.

Our structured sampling strategy enables the use of expensive and powerful image representations and machine learning techniques. In this section we use Structured Sampling inside the Bag-of-Words based object recognition framework described in Section 4.4. The reduced number of object locations compared to an exhaustive search make it feasible to use such a strong Bag-of-Words implementation.

Results are obtained using the official evaluation server of the Pascal 2010 detection challenge. This means the evaluation is independent as the test data has not been released. We compare with the top-4 of the competition. Note that while all methods in the top-4 are based on an exhaustive search using variations on part-based model of [27] with HOG-features, our method differs substantially by using a structured sampling and Bag-of-Words features. Results are shown in Figure 40 and Table 10.

As can be seen, our method yields the best results for the classes plane, cat, cow, table, dog, plant, sheep, sofa, and tv. Except table, sofa, and tv, these classes are
This chapter

Table 10: Results from the Pascal VOC 2010 detection task test set. Our method is the only object recognition system based on Bag-of-Words. It has the best scores for 9, mostly non-rigid object categories, where the difference is up to 0.056 AP. The other methods are based on part-based HOG features, and perform better on most rigid object classes.

all non-rigid. This is expected, as Bag-of-Words is theoretically better suited for these classes than the HOG-features. Indeed, for the rigid classes bike, bottle, bus, car, person, and train the HOG-based methods perform better. The exception is the rigid class tv. This is presumably because our structured sampling performs well in locating tvs, see Figure 37.

We conclude that our structured sampling enables the use of more expensive features and classifiers, as it significantly reduces the number of locations visited compared to an exhaustive search. Yet at the same time it yields higher quality locations. Interestingly, the Bag-of-Words representation is complementary to part-based HOG-features, as the former is good at identifying non-rigid objects while the latter is good at identifying rigid objects. Overall, our object recognition system substantially improves the state-of-the-art by up to 0.056 AP for 9 out of 20 object categories.
An upper bound of location quality

In this experiment we investigate how close our structured sampling locations are to the optimal locations in terms of recognition accuracy. We do this on the Pascal VOC 2007 test set.

**Figure 41:** The influence of the quality and quantity of the object hypotheses on the Pascal VOC 2007 detection performance when using our object recognition framework. The red curve denotes the performance using the top \( n \) locations of our “quality” structured sampling method, which has a MABO of 0.758 at 500 locations, 0.855 at 3000 locations, and 0.883 at 10,000 locations. The magenta curve denotes the performance using the same top \( n \) locations but now combined with the ground truth. Hence here the MABO is always 1. At 10,000 locations, our object hypothesis set is close to optimal in terms of object recognition accuracy.

The red line in Figure 41 shows the MAP score of our object recognition system when the top \( n \) boxes of our “quality” structured sampling method are used. The performance starts at 0.283 MAP using the first 500 object locations with a MABO of 0.758. It rapidly increases to 0.356 MAP using the first 3000 object locations with a MABO of 0.855, and then ends at 0.360 MAP using all 10,097 object locations with a MABO of 0.883.

The magenta line shows the performance of our object recognition system if we include the ground truth object locations to our hypotheses set, representing an object hypothesis set of “perfect” quality with a MABO score of 1. When only the ground truth boxes are used a MAP of 0.592 is achieved, which is an upper bound of our object recognition system. However, this score rapidly declines to 0.437 MAP using as few as 500 locations per image. Remarkably, when all 10,079 boxes are used the performance drops to 0.377 MAP, only 0.017 MAP more than when not including the ground truth. This shows that at 10,000 object locations our hypotheses set is close to what can be optimally achieved for our recognition framework. The most likely explanation is our use of SIFT, which is designed to be shift invariant [53]. This causes approximate boxes, of a quality visualised in
Figure 36, to be still good enough. However, the small gap between the “perfect” object hypotheses set of 10,000 boxes and ours suggests that we arrived at the point where the degree of invariance for Bag-of-Words may have an adverse effect rather than an advantageous one.

The decrease of the “perfect” hypothesis set as the number of boxes becomes larger is due to the increased difficulty of the problem: more boxes means a higher variability, which makes the object recognition problem harder. Earlier we hypothesized that an exhaustive search examines all possible locations in the image, which makes the object recognition problem hard. To test if structured sampling alleviates the problem, we also applied our Bag-of-Words object recognition system on an exhaustive search, using the locations of [27]. This results in a MAP of 0.336, while the MABO was 0.829 and the number of object locations 100,000 per class. The same MABO is obtained using 2,000 locations with structured sampling. At 2,000 locations, the object recognition accuracy is 0.347. This shows that structured sampling makes the problem easier compared to exhaustive search.

To conclude, there is a trade-off between quality and quantity of object hypothesis and the object recognition accuracy. High quality object locations are necessary to recognise an object in the first place, but fewer object hypotheses make the problem easier. Remarkably, at a reasonable 10,000 locations, our object hypothesis set is close to optimal for our Bag-of-Words recognition system. This suggests that our locations are of such quality that a lower degree of invariance for Bag-of-Words is desirable.

4.6 CONCLUSIONS

In this chapter we proposed structured sampling for object recognition. Structured sampling combines the strengths of both segmentation and exhaustive search. Just as in segmentation, we use the structure of the image to guide our sampling in a data-driven fashion. Just as in exhaustive search, we aim to capture all relevant diversity in object locations and object appearances. At the same time, the image structure helps to reduce the number of locations compared to an exhaustive search. As there is a large variety of complementary reasons that a region forms an object, rather than relying on a single strong algorithm which deals with most reasons, structured sampling diversifies the set of sampling strategies to deal with as many reasons as possible. This makes structured sampling stable, robust, and independent of the object-class, where object types range from rigid (e.g. car) to non-rigid (e.g. cat), and even to amorphous (e.g. water). Hence structured sampling yields a data driven set of high-quality, object-independent locations which is of limited size.

The structured sampling method we proposed in this chapter combines the partitionings of a variety of hierarchical grouping algorithms, diversified by using complementary, invariant colour spaces and complementary grouping criteria. Hence we deal with a variety of lighting conditions and a variety of data-driven grouping possibilities.

In terms of region-based locations, our structured sampling obtained good quality hypotheses at 0.730 MABO using 22,491 locations. More importantly, if we combine also the methods of [13] and [22], quality increases to 0.758 MABO using 25,355 locations. This is a significant increase and shows there is still much
potential for structured sampling for finding a high quality set of regions-based locations.

In terms of box-based locations, our structured sampling yields better locations at fewer locations than an exhaustive search: whereas [27] yields a MABO of 0.829 at 100,000 boxes per class, we obtain a MABO of 0.879 using 10,097 class-independent locations. This is both a significant increase in location quality and a significant reduction in the number of locations visited. The latter can be used to use stronger machine learning and more powerful appearance models.

In this chapter, the reduced number of locations enable us to do object recognition using a powerful Bag-of-Words implementation. Our method yields competitive results compared to the state-of-the-art, which are all based on part-based models using HOG-features and an exhaustive search. Interestingly, these methods are complementary. While the part-based models perform best on most rigid classes, our Bag-of-Words method performs best on most non-rigid classes. Overall, our method yields the best results for 9 out of 20 object classes with up to 0.056 AP in improvement on the Pascal VOC 2010 detection challenge.

Finally, we tested how our structured sampling compares to a “perfect” sampling technique. We did this by adding the ground truth boxes to our structured sampling set and evaluate the resulting set in our object recognition framework. Remarkably, at a reasonable 10,097 locations the performance is close to optimal: Our locations yield 0.360 MAP, while if we include the ground truth it yields 0.377 MAP. This suggests that the level of invariance of our Bag-of-Words object recognition system is higher than needed for the quality of boxes we generate.
THE WINDOWS THAT TELL THE STORY OF AN IMAGE

As a step towards generating the full narrative of an image, this chapter aims to jointly predict the presence of multiple objects using single-object classifiers only, necessary to deal with the combinatorial explosion of multiple objects. To this end we introduce the most telling window, which focuses its window on the most discriminative features of an object for its recognition. Using different features within the image enables classifier combinations through an independence assumption. Our main contributions are: (I) We introduce the most telling window for multi-class object classification as a step towards generating the story of an image. (II) We show that, depending on the object class and image composition, the most telling window focuses on an object part, complete object, or an object collection for its recognition. For single-object classification, the most telling window outperforms Bag-of-Words. Combined with exact localisation it yields competitive results on the Pascal 2010 classification challenge. (III) For joint three- and two-object classification on Pascal 2007 our most telling window significantly outperforms Bag-of-Words and exact localisation plus Bag-of-Words: Bag-of-Words respectively yields 0.079 MAP and 0.138 MAP, exact localisation plus Bag-of-Words yields 0.109 MAP and 0.157 MAP, while the most telling window yields 0.170 MAP and 0.216 MAP.

5.1 INTRODUCTION

Every picture tells a story. For some pictures the story is very obvious, even straightforward. The showroom photo of a chair, see Figure 42a, is about nothing else but that. The picture of a motorcycle in Fig. 42b is from real life with other objects in view, but the photograph is about one object where the rest of the picture serves as a frame of context to the object. Many pictures of famous scenes as the one in Fig. 42c are about a single topic also. For other, more complex images, the story can be single topic on a higher level of abstraction, such as depictions of a city-scape or party. Automatic concept classifiers based on Bag-of-Words [18,82] have demonstrated tremendous progress in recognising single concepts in the image as a big step towards understanding the semantics of an image. Now the next horizon is to understand the story of the picture.

Some pictures tell a story of two objects. When two people meet, the picture is about person and person, although the essence is their combined presence. When aeroplanes fly in formation, see Figure 42d, the picture of the group tells more than a single plane. In Figure 42e the story is spun between three objects: car, horse and person. It specifically frames the scene at horse riding for leisure. And in Figure
42f, a chair, diningtable, and a pottedplant tell a story about a certain, cosy style of home decoration. In these cases, the story is more and more specific when the number of objects grow. Many pictures tell such a story of multiple objects. In this chapter we focus on the joint classification of one, two or three objects as the first step towards telling the story of an image.

**Figure 42:** Images tell stories. Some involving a single concept, others involving multiple concepts. This chapter aims to jointly classify multiple objects as a step towards telling the full story of an image.

We aim to identify pictures with combined objects: person with horse and car, for example. A straightforward strategy to recognise a combination is to train the pair or triplet of objects as one combined class. However, examples of horse and car will be much more rare than horse and car alone, not even mentioning the rarity of person with horse and car. In the Pascal VOC 2007 collection [23], out of all possible 190 dual-class combinations 88 are without learning example, and only 17 class-pairs have more than 50 learning examples. Out of all 1140 triple-class combinations 1029 are without learning examples and only 3 triples have more than 25 learning examples. And this is within a dataset containing just 20 classes for which the data has been especially collected. Hence ultimately learning combinations fails as the number of object pair or triplets is vast, making it prohibitively laborious to collect training samples and computationally infeasible to learn all combinations of classes.

A better strategy is to combine the outcomes of separate object classifiers. Current object classification, however great the recent progression has been, is not suited for combining. This is because the best object classifiers are based on Bag-of-Words [18, 82] and start from the underlying assumption that the all pixels in the image best provide support for the object in the image. Searching for a combination of two objects in the image by using current classifiers who gather evidence from all pixels will violate the independence assumption and lead to poor performance. In this chapter we focus the attention to the “most telling window” of an object, reintroducing locality in the Bag-of-Words classification. This opens the door to combining multiple object classifications as now the features they operate on are spatially disjunct and hence largely independent.
There is an important difference between the “most telling window” and trying to find the exact location of the object, an area where recently tremendous progress has been made \cite{20,27,42,49,104}. Our aim is not to accurately localize the object but rather we aim for the window which is most discriminative for the topic, the most informative one. The most telling window may have a different focus than the localization: a face when detected is a better identifier for a person than the total body including all possible variations in clothing. When sheep are in a group it is hard to single out their individuals, while the most telling window will instead focus on the group as a whole which is easier to identify. Occlusions, which occur more frequently as the image contains more objects, are generally difficult if one searches for the whole object, while the most telling window focuses on those parts which are visible. Finally, when objects are interacting, such as “man and horse”, the appearance of the objects may change. The man will take an atypical position and is hard to find completely, while the most telling window may focus on the still typical appearance of the upper body. Hence for object classification there are good reasons not to focus on an accurate localisation of the object - an important topic in its own right - but rather to focus on “the most telling window”.

We leave finding the most telling window to the classifier, as “most telling” is only defined in the context of object classification. In training our most telling window classifiers we limit ourselves to using only the complete object locations in terms of bounding boxes; we do not require laborious annotation in terms of object parts or groups of objects. While our method was designed for searching combined objects, we have found that the most telling window is frequently beneficial for single object recognition as well. From \cite{57,96} it was already known that using features from or close to the object may improve recognition. And while the surrounding of an object carries information to classify the object, when the location of the object is known, the rest of the image adds little more its distinction both in human vision \cite{7} and in computer vision \cite{96}. Tailoring the features to the most telling window yields better discrimination than using features derived from the whole image.

Our main research questions are: 1. Can we recognise pairs or triplets of objects as the most important ingredients of the story the picture tells? 2. Can we recognise pairs or triplets of objects when training only single object classifiers, that is while avoiding to train every possible combined pair or triplet? 3. To that end, can we find the most telling window which is most discriminative, yet permits independence in the object classification? We evaluate our most telling window method on the difficult Pascal VOC 2007 dataset on the joint classification of one, two, and three objects.

5.2 Related Work

In previous research the classification results of a global image representation such as Bag-of-Words is often used to better localise objects within the image. Many researchers use contextual or global image features as a prior for the object location \cite{44,93} or to increase confidence that the located object is indeed of the correct category \cite{21,42}. Works in pixel-wise classification incorporate global image representations \cite{106} or use the output of a presence prediction system directly \cite{19,38}. In this chapter we move in the opposite direction: we use a local
representation to improve the prediction of the presence of an object within the whole image.

Harzallah et al. [42] showed that object localisation has matured to the point that it can be used to increase image classification. Encouraged by their result, in this chapter we especially design a system which uses object features within the image for the prediction of its presence. This results in the following important differences: 1) Our most telling window uses the most discriminative object region for identifying the object. This region may be an object part, complete object, or a collection of objects. 2) In object localisation the classifier is often trained to penalise inexact object locations. In our chapter such inexact object locations may be the most discriminative, hence we encourage the use of such locations. 3) We developed our method for the joint classification of multiple objects. As will be shown, using exact localisation may not be optimal for this task.

Bosch et al. [10] showed earlier that localisation of the complete object yields better results than the standard global image representation in a Bag-of-Words framework on the Caltech-256 dataset. However, such improvements using localisation alone have not been transferred to the much more difficult Pascal VOC dataset [23,42]. In contrast to [10], in this chapter we propose a different search technique to find object regions, use a more complicated training scheme, and report results on the Pascal VOC dataset.

In this work we use the most telling window within an image to represent the object features. The use of such a “best window” is also common within instance matching, finding the exact same object within another image. This is for example done by Lowe [53] and Jamieson et al. [45] but also by Philbin et al. [71] who base image retrieval based on a query region which acts as the object instance. In this chapter we use the best classifying window within the image not for object instance matching but for object-based image classification. The most important difference is that the best matching window may be completely different from the best classifying window: all details and the spatial layout are important for image matching but only a small detail may be important for object-based image classification.

In Multiple Instance Learning the goal is to retrieve images based on discriminative regions within the image [32,116]. When applied to object-based image classification, the goal would translate to classifying image based on one or multiple discriminative object regions, similar to the goal of this chapter. But the main problem that Multiple Instance Learning addresses is to automatically generate a training set out of partially labelled data. In the case of [32] they create training regions within an image based on globally annotated images. As this is a costly procedure, the datasets used are generally of a limited size. In this chapter we already have a ground truth annotation of objects hence we bypass the main problem of Multiple Instance Learning. Instead, we show that on the highly competitive Pascal VOC dataset the use of the most telling window is beneficial both for single- and multi-class object classification. As larger datasets often have only global annotations, this result underlines the increasing importance of Multiple Instance Learning.

In their work on object identification, Ferencz et al. [29] showed that object instances can be successfully discriminated by zooming in to specific object-parts. As object instances only differ slightly in appearance, their method presupposes localized and aligned objects and uses more specific features than in general clas-
sification. In this chapter we show that the idea of recognition based on discriminative object parts transfers to the domain of image classification while using a highly varying dataset.

For computational efficiency we only want to consider a small subset of all possible windows to consider as object representation. A classical window selection method is the sliding window approach, which is an exhaustive search over all possible windows within the image using a weak but fast classifier \([31,42,74,107]\). A fast alternative is the jumping window approach which selects windows based on discriminative visual words \([16,104]\). However, as objects and object parts are usually visually coherent, there is an increasing interest in using segmentation either as additional feature for candidate window selection \([2,110]\) or to generate regions by itself \([13,22,41,91]\). In this chapter we use the segmentation as selective search method of \([102]\), which was proven to be superior to sliding \([42]\) and jumping \([104]\) windows in generating a small set of good object locations. As this method does not involve learning, we expect it to also generate object parts and object collections, which we will demonstrate in this chapter.

A good alternative to the window selection method we use would be the Efficient Subwindow Search (ESS) strategy of Lampert et al. \([49]\) which uses a branch and bound method to find the global optimal window within an image. But as stated by \([104]\) and acknowledged by \([48]\), with non-linear kernels this method still visits several thousands of regions per image, many more than evaluated in our chapter.

Just before submission of this article, Farhadi and Sadeghi \([25]\) make an interesting step towards the detection and localisation of combined objects. In their contribution they propose to detect and localize “Visual Phrases”, defined as complex composites such as person riding horse. Even with a few training examples, their detectors perform significantly better than the combination of the location of an individual person with an individual horse detector. Performance improves because objects in a relation may substantially change appearance. While their strategy is highly applicable to standard composites, we make a different contribution. We do not need the combined examples as collecting data and training models for all possible visual object phrases is too cumbersome, especially when there are three things to combine. Instead, our approach for dealing with appearance changes of objects in relation is to use the most telling windows of these objects and combine scores using an independence assumption. This way we can recognise pairs of triplets while not even needing a single example to learn them.

To summarize, our novelties are the following: 1) We provide a framework to combine single-object classifiers to jointly recognise multiple objects as step towards the full narrative of an image. 2) We reintroduce location inside the Bag-of-Words framework, enabling the combination of classifiers through an independence assumption as the used features are often spatially disjunct and therefore largely independent. 3) Our most telling window focuses on those parts of the image that are most discriminative for the identification of the target object. This enables the identification of an object by recognising an object part, complete object or object collection, and at the same time makes it robust to occlusions and pose changes.
The computational approach builds upon Bag-of-Words. The learning phase requires annotations in the form of labels and bounding boxes around the objects. The testing phase requires a candidate set of potentially interesting windows. An overview of the method is given in Figure 43.

**Figure 43:** Our framework for using the most telling window for single or multi-object narratives. For all images we assign visual words per pixel. In the training phase we use the ground truth windows to extract visual word histograms from positive and negative training examples. Then we learn a Support Vector Machine model. In the testing phase we create a set of candidate windows by a hierarchical grouping. This yields segments at different scales which may include object parts, complete objects, or object collections. Then visual word histograms are created for each candidate window. The learned SVM-model is applied to the candidate windows and the window with the highest classification score is selected. This score acts as the classification score for the target object. We add hard negative examples by taking the highest scoring window of negative training images. Because each classifier selects its own object-centred window, we use an independence assumption to generate multi-object narratives.
5.3.1 The Bag of Words implementation

This chapter builds upon Bag-of-Words. As we use windows within the image for classification, we sample SIFT descriptors [53] extremely dense at each pixel enabling small regions to still generate sufficient statistics.

To keep computational costs reasonable we use our fast Bag-of-Words implementation et al. [99]. SIFT descriptors are calculated at single scale of 16 by 16 pixels using a Gaussian derivative filter of $\sigma = 0.8$, extracted using our publicly available descriptor extractor software [99]. PCA is used to reduce intensity-based SIFT from 128 to 48 dimensions and colour SIFT [103] from 384 to 104 dimensions. The visual vocabulary is formed by a Random Forest [63] consisting of four binary decision trees of depth ten, resulting in a total of 4096 visual words per SIFT descriptor. The location of a visual word is assigned to the centre pixel of its image patch. Classification is done using a Support Vector Machine with a fast, approximate Histogram Intersection classifier [15, 54]. Uneven visual word counts are balanced by taking the square root of the visual word frequency histograms. The uneven number of positive and negative examples are balanced by weighing them inverse proportional to their prior.

All classifier combinations in this chapter, for both classes and methods, are done by multiplying probabilities. For each class $c_i$ the distance to the decision boundary $d_{c_i}$ is converted to a (pseudo-) probability using a sigmoid function

$$ P(c_i) = \frac{1}{1 + e^{d_{c_i}}} . \quad (5.1) $$

Alternatively, one could combine scores by adding distances or fitting a full sigmoid function per class [15] using

$$ P(c_i) = \frac{1}{1 + e^{\alpha d_{c_i} + \beta}} , \quad (5.2) $$

both with and without using the bias term $\beta$. Yet for all combinations we found the basic sigmoid function to perform similar or better.

All experiments use the spatial pyramid [51]. In the representation of the whole image, which is the common Bag-of-Words representation, we use the whole image and a regular subdivision into into three horizontal regions, found to work well by several researchers [58, 90]. This will serve as our Bag-of-Words baseline. In the representation of the candidate regions we use a regular subdivision into 2 by 2 regions. As the representation of both the baseline and the candidate regions use a subdivision into 4 regions, the final vector size is the same and has 81,920 dimensions.

5.3.2 Training

We want our classifiers to automatically select the “most telling” window of the object, while keeping annotation requirements limited to bounding boxes around the object. We do this by using three types of learning examples: A) Positive training examples. These are obtained using the ground truth windows of all target objects. B) Negative training examples of non-target objects. These are obtained from the ground truth windows of non-target objects in negative images.
We use only negative images to avoid any target-object features to be incorporated in negative examples. To keep the size of the training set limited, we take for each negative image a single non-target object. C) Hard negative training examples. These examples include difficult background windows, as well as non-target object-parts and non-target object collections. We obtain the hard negative examples using a retraining step on the negative training images as described below.

We collect hard negative examples using the method proposed in [50]. We first train a classifier on the target and non-target objects given by the ground truth (type A and B). Then we apply the classifier to all negative images. For each negative images, the window with the highest score serves as a hard negative example. Hence each retraining step yields \( n \) extra hard negative examples, \( n \) being the number of negative images.

Empirically we have found that only two retraining steps are sufficient to yield optimal classification performance. For all classes except person, the initial training set is about the size of the total number of training images. Each retraining step yields less than the number of training images as extra examples. In practice, the final training set is about three times the number of training images.

5.3.3 Hierarchical Grouping for Candidate Window Generation

During testing we can only evaluate a limited number of windows due to computational constraints. We generate windows using the window selection mechanism of [102] which was created for finding complete objects. In this chapter we show that this algorithm generalizes to finding object parts and object collections as well, as required by our most telling window.

The starting point of [102] is an oversegmentation created using Felzenszwalb and Huttenlocher [28], yielding relatively small, homogeneous regions which describe object parts. We hierarchically and greedily group similar regions together in order to merge the parts into complete objects, and the objects into collections. By considering all regions throughout the hierarchy we capture all levels of granularity.

Specifically, let \( R^1 \) be the set of \( t \) initial regions \( \{r_1, \ldots, r_t\} \) generated by [28]. Let \( A^1 \) be a set of undirected adjacency pairs where \((r_i, r_j) \in A^1\) for all regions \( r_i \) and \( r_j \) that are connected. For each adjacency pair \((r_i, r_j) \in A^1\) a region based similarity measure \( S(r_i, r_j) \) is defined and calculated for all pairs in \( A^n \). A merge operation \( M(r_i, r_j) \) is defined subject to \( i \neq j \) as a series of production rules on \( R^n \) and \( A^n \):

\[
R^n \rightarrow R^{n+1} \Rightarrow r_i, r_j \rightarrow r_i, r_j, r_{n+t},
\]

where \( r_{n+t} \equiv r_i \cup r_j \) is the union of both regions.

\[
A^n \rightarrow A^{n+1} \quad \Rightarrow
\]

\[
\forall x, (r_j, r_x) \notin A^n : (r_i, r_x) \rightarrow (r_{n+t}, r_x)
\]

\[
\forall x, (r_i, r_x) \notin A^n : (r_j, r_x) \rightarrow (r_{n+t}, r_x)
\]

\[
\forall x : ((r_i, r_x), (r_j, r_x)) \rightarrow (r_{n+t}, r_x),
\]

where care has been taken to not include double adjacency pairs in \( A^{n+1} \).
The regions are greedily merged together as follows. In each step \( n \) the adjacency pair \((r_i, r_j) \in A^n\) whose similarity measure \( S(r_i, r_j) \) is highest is merged according to the production rules defined above. For all new adjacency pairs \( \forall x, (r_{n+t}, r_k) \) the corresponding similarities are calculated. This is repeated until the set of adjacency pairs is empty, which is the case at step \( t-1 \): \( A^{t-1} = \emptyset \). At this point all \( t \) initial regions have been merged into a single region which is the entire image.

The similarity measure is defined as
\[
S(r_i, r_j) = S_{\text{texture}}(r_i, r_j) + S_{\text{size}}(r_i, r_j),
\]
where \( S_{\text{texture}}(r_i, r_j) \) is the histogram intersection between the SIFT-like texture histograms of \( r_i \) and \( r_j \), and where
\[
S_{\text{size}}(r_i, r_j) = \text{size}(im) - \text{size}(r_i) - \text{size}(r_j)
\]
encourages small regions to merge earlier, keeping regions in \( R^n \) of similar size throughout the algorithm. The latter is necessary to, for example, let an arm first merge with a hand and only afterwards with the body.

To account for different scene conditions the regions of eight different hierarchical groupings are used. Variations are made by selecting two parameter settings for \([28]\): \( k = 100, 200, \sigma = 1, \min\text{Size} = k \), giving different importance to colour similarity, and by using four colour spaces with different invariance properties: opponent colour space, RGB, normalized rgb, and hue. Accounting for different scene conditions was found to significantly improve the quality of the windows \([102]\).

After the regions are obtained, the tight fitting windows around the regions are used as candidates for the most telling window. Using boxes instead of regions makes the algorithm robust against mistakes in the segmentation and makes the test phase more consistent with the training phase as the ground truth is defined in terms of boxes.

In \([102]\) the described window selection mechanism was tested on finding complete objects within the Pascal VOC 2007 test set using the Pascal Overlap Criterion. With 1536 candidate windows per image a recall of 96.7% was obtained. As argued earlier, the nature of the algorithm allows it to generate object parts and object collections as well, shown qualitatively in Figure 44.

### 5.3.4 Multi-Object Classification

Joint classification of multiple objects is possible by directly learning a classifier on the given multi-class problem. Yet even for two objects there are already many possible combinations for which training data is necessary and which have to be learned. Ultimately, this strategy is therefore computationally infeasible and requires an unreasonable amount of training samples.

We designed our most telling window to support multi-object classification by combining single-object probabilities. We let the classifier select the most telling window, whose focus is likely different for each class. Therefore we make the assumption that given any two classes \( c_i \) and \( c_j \), their corresponding features \( x_{c_i} \) and \( x_{c_j} \) are independent. Furthermore, we assume that each feature \( x_{c_i} \) is only generated by its corresponding class \( c_i \). The derivation below is made on a two-class problem, but can easily be extended to a multi-class problem.
Figure 44: Examples of windows generated by Segmentation as Selective Search. The first column depicts the complete images. The other columns show example windows. These windows include complete objects, object parts, and object collections.

For the classification of two classes, we use Bayes Rule to describe the probability of occurring jointly:

$$P(c_1, c_2 | x) = \frac{P(x | c_1, c_2) P(c_1, c_2)}{P(x)}$$  \hspace{1cm} (5.7)

where $x$ represents all the features of the image. Now using the assumptions above, we can rewrite this to

$$P(c_1, c_2 | x) = \frac{P(x_{c_1} | c_1) P(x_{c_2} | c_2) P(c_1, c_2)}{P(x)}.$$  \hspace{1cm} (5.8)

In our framework we learn classifiers using a one-versus-all strategy, where we balance the number of positive and negative training examples. Therefore our models have no class-bias. This means we can assume equal single-class priors, leading to

$$P(c_1, c_2 | x) = \frac{P(c_1 | x_{c_1}) P(c_2 | x_{c_2}) P(c_1, c_2)}{K}.$$  \hspace{1cm} (5.9)

where $K$ is a normalization factor that is constant for all class-pairs and which includes both the prior on the data and the single-class priors.

In this chapter we evaluate our object-based image classification on a ranking problem. Therefore we consider the joint class distribution $P(c_1, c_2)$ as a constant.
5.4 Evaluation

We use the Pascal VOC 2007 dataset as it is one of the hardest and most frequently studied dataset with box annotations. This dataset consists of 5011 training images and 4952 test images, and contains the following twenty object classes: aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining-table, dog, horse, motorbike, person, potted-plant, sheep, sofa, train, and TV/monitor. The classes are annotated by a label and bounding box around each object. A single image may contain multiple instances of the same or different object classes. As evaluation measure we use Average Precision (AP) and Mean Average Precision (MAP). Invariably, any parameter optimization was done on the training set only.

Our evaluation is divided in two sections. First we evaluate single object classification. We use Bag-of-Words, which uses features from the whole image as evidence, as a baseline. We compare with the use of exact localisation and the most telling window. We are particularly interested in how the most telling window deviates from the exact location of the object. Is it bigger, smaller, or identical to the ground truth box location? And, can the most telling window improve object classification? Then, we focus on joint two and three object classification by combining results of the classifiers without learning the combination.

5.4.1 Single Object Classification

We first investigate the validity of using the most telling window per image for single-class object classification. We use a state-of-the-art Bag-of-Words method of 2010 as the baseline [99] and compare with the most telling window.

As can be seen in Figure 45 the most telling window significantly outperforms the use of the complete image with Bag-of-Words. While Bag-of-Words yields a MAP of 0.588, the most telling window yields a MAP of 0.630, which is a significant improvement.

![Figure 45: Results of presence prediction on the Pascal VOC 2007 classification task. Comparison of Bag-of-Words with the most telling window. The most telling window clearly outperforms Bag-of-Words for most classes. The overall performance is 0.588 MAP for Bag-of-Words and 0.630 MAP for the most telling window.](image-url)
The most telling window results in better performance for objects like *cow, dining table, motorbike, potted plant, sheep*, and *tv/monitor*. These objects have a distinct appearance but share their context with other competing objects. As cow and sheep share a grass-field context it is harder to discriminate them when the information is derived from the whole scene. The same holds for motorbikes which share their context with cars and buses, as well as dining tables and potted plants which share their context with sofas and chairs.

<table>
<thead>
<tr>
<th>Object</th>
<th>Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>aeroplane</td>
<td><img src="image" alt="Image of aeroplane" /></td>
</tr>
<tr>
<td>bicycle</td>
<td><img src="image" alt="Image of bicycle" /></td>
</tr>
<tr>
<td>boat</td>
<td><img src="image" alt="Image of boat" /></td>
</tr>
<tr>
<td>cat</td>
<td><img src="image" alt="Image of cat" /></td>
</tr>
<tr>
<td>motorbike</td>
<td><img src="image" alt="Image of motorbike" /></td>
</tr>
<tr>
<td>person</td>
<td><img src="image" alt="Image of person" /></td>
</tr>
</tbody>
</table>

**Figure 46**: Visualisations of the most telling window for the highest ranked images various classes. Note that often the most discriminative image window corresponds to an object part or collection rather than the complete object.

Figure 46 shows the most telling windows of the highest ranked images for six of the twenty objects. The most telling window indeed tells the story of the object: it zooms in to the object features which are most informative for the objects to discriminate the topic of the image against other objects. *Aeroplanes* and *motorbikes* are generally characterised by the complete object. However, the fifth picture of *plane* is characterised by the more informative formation of planes. The last picture of *motorbike* captures two instances. For *bicycles*, a variety of most telling frames is selected: images 1, 2, 3, and 6 capture the whole bike, images 4 and 8 capture the front wheel and part of the frame, ignoring the rear wheel, while images 5 and 7 capture a collection of three wheels in two bikes. *Boats* and *cats* are characterised by parts. For *boat* often a sail or mast against the sky is most telling of the class. For *cat* the classifier often ignores the body and focuses on its head instead, which means that classifier correctly learns that fur occurs in all animals and its head is the most informative part. *Trains* (data not shown) are regularly characterised by their first wagon; the rest of the train is just a repetition of the same structure. Finally, *persons* are frequently recognised as groups, often focusing on the upper bodies and heads. To conclude, the most telling window flexibly focuses on the most informative parts of an image, which may be an
object part, complete object, or object collection, depending on the object class and image composition.

To get a better insight into which windows are selected to be the most telling, we analyse the overlap between the most telling window $w_t$ and the set of ground truth object boxes $W_{gt}$ for all positive images of each class. First we measure how well the most telling window captures a complete object, using the ground truth window with the Maximum Pascal Overlap (MPO), where the Pascal Overlap is defined by dividing the area of intersection of two boxes by the area of their union:

$$MPO = \max_{\tilde{w}_{gt} \in W_{gt}} \frac{\text{area}(w_t \cap \tilde{w}_{gt})}{\text{area}(w_t \cup \tilde{w}_{gt})}.$$  \hspace{1cm} (5.10)

We also measure what area of the most telling window contains object features, which we call Object Area Overlap (OAO) and is defined as

$$OAO = \frac{\text{area}(w_t \cap \bigcup W_{gt})}{\text{area}(w_t)},$$  \hspace{1cm} (5.11)

where in a slight abuse of notation $\bigcup W_{gt}$ is the union of all boxes in $W_{gt}$. In Figure 47 we plot against threshold $t$, the fraction of most telling windows for which $MPO > t$ and $OAO > t$. Note that for a threshold $t$, the shown fraction of most telling windows is the average over all classes.

![Figure 47](image.png)

**Figure 47:** The fraction of the most telling windows which have a higher overlap score than a given threshold.

As is apparent from Figure 47, approximately 60% of the most telling windows have a Pascal score over 0.5. These cases can be considered to represent a complete object. In contrast, over 80% of the selected windows contain more than 50% of the object, and over 20% of the selected windows contain exclusively object features. This shows that the most telling window in a majority of cases describes object features, but regularly focuses on object parts or object collections. The distinction between the curves for the PASCAL overlap measure and the OAO is most revealing that the most telling window is by far not always the object’s silhouette, and reverse.
We conclude that the most telling window, even though it has been trained using only boxes around complete objects as positive learning examples, chooses to select object parts and object collections as better discriminating. This provides insight for that complete segmentation is rarely needed for cognition, and should follow after classification of the object. The most telling window for object recognition is not limited to complete objects, but can also be a part or collection depending on the object class and image composition.

Next we compare the most telling window with two other systems using local object information. As a baseline for this experiment we use the results of Harzallah et al. [42], who combine Bag-of-Words classification scores with an exact object localisation method. Analogously, we combine Bag-of-Words with the exact localisation of [27], which is a state-of-the-art, part-based localisation framework which uses HOG features [20].

![Figure 48: Result for the classification of a single object, evaluated on Pascal VOC 2007. Overall [42] yields a MAP of 0.635, exact localisation [27] plus Bag-of-Words yields 0.657 MAP, and the most telling window plus Bag-of-Words yields the highest score of 0.674 MAP. In general, exact localisation performs better on rigid objects, while the most telling window performs better on non-rigid objects.](image)

The results in Figure 48 show that the combination of the most telling window with Bag-of-Words has the highest overall performance: the method of [42] yields a MAP of 0.635 MAP, [27] plus Bag-of-Words yields 0.657 MAP, while the most telling window plus Bag-of-Words yields 0.674 MAP. Quite surprisingly, results differ widely for individual objects (compared to differences between state of the art methods commonly reported in literature). The method of [42] is significantly better for bottle (+0.07 AP) and chair (+0.04 AP). Exact localisation [27] plus Bag-of-Words is significantly better for bottle (+0.07 AP), bicycle (+0.09 AP), and bus (+0.03 AP). Bottle, bicycle, and bus are all rigid classes for which the HOG-based method of [27] is expected to do well. In contrast, the combination of our most telling window plus Bag-of-Words is superior for the classes tv, plant, cow, sheep, and dog (+0.04 - 0.06 AP). These classes include most of the non-rigid, flexible objects where exact localization generally yields much worse classification performance.

From Table 11 it is evident that the most telling window plus Bag-of-Words in all but one case (bus) outperforms the most telling window alone. Furthermore, the combination is always better than using the Bag-of-Words based on the entire
image. Finally, the most telling window is complementary to exact object localisation, as a combination of exact localisation [27], Bag-of-Words and the most telling window raises performance to 0.707 MAP. In conclusion, for single-object classification our most telling window is complementary to using exact localisation scores and works especially well for non-rigid objects with a high variety in appearance.

To be sure that the results remain valid for the newest data, we apply combination of the most telling window, Bag-of-Words [99] and the exact localisation [27] on the Pascal VOC 2010 challenge. For this challenge the test labels have not been released. Instead, results are analysed using an online evaluation server. A comparison with the top results of this competition is shown in table 12.

As can be seen, our method yields competitive results, yielding the best scores for 5 out of 20 objects. Overall, we achieve a MAP of 0.711, comparable to 0.709 MAP of NEC Labs America and 0.712 MAP of NLPR (National Laboratory of Pat-

<table>
<thead>
<tr>
<th>plane</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>.802</td>
<td>.606</td>
<td>.567</td>
<td>.722</td>
<td>.243</td>
<td>.643</td>
<td>.770</td>
<td>.575</td>
<td>.493</td>
<td>.403</td>
</tr>
<tr>
<td>[42] Loc only</td>
<td>.582</td>
<td>.595</td>
<td>.193</td>
<td>.269</td>
<td>.375</td>
<td>.608</td>
<td>.802</td>
<td>.367</td>
<td>.393</td>
<td>.407</td>
</tr>
<tr>
<td>[27] Loc only</td>
<td>.562</td>
<td>.760</td>
<td>.147</td>
<td>.344</td>
<td>.421</td>
<td>.683</td>
<td>.856</td>
<td>.414</td>
<td>.447</td>
<td>.407</td>
</tr>
<tr>
<td>Ours</td>
<td>.754</td>
<td>.676</td>
<td>.461</td>
<td>.595</td>
<td>.290</td>
<td>.675</td>
<td>.861</td>
<td>.590</td>
<td>.482</td>
<td>.548</td>
</tr>
<tr>
<td>[42]</td>
<td>.772</td>
<td>.693</td>
<td>.562</td>
<td>.666</td>
<td>.455</td>
<td>.681</td>
<td>.834</td>
<td>.536</td>
<td>.583</td>
<td>.511</td>
</tr>
<tr>
<td>Base + [27]</td>
<td>.823</td>
<td>.781</td>
<td>.587</td>
<td>.733</td>
<td>.424</td>
<td>.748</td>
<td>.891</td>
<td>.596</td>
<td>.539</td>
<td>.477</td>
</tr>
<tr>
<td>Base + Ours</td>
<td>.829</td>
<td>.695</td>
<td>.598</td>
<td>.759</td>
<td>.384</td>
<td>.718</td>
<td>.867</td>
<td>.609</td>
<td>.545</td>
<td>.561</td>
</tr>
<tr>
<td>Base + Ours + [27]</td>
<td>.826</td>
<td>.789</td>
<td>.605</td>
<td>.761</td>
<td>.504</td>
<td>.755</td>
<td>.903</td>
<td>.628</td>
<td>.577</td>
<td>.584</td>
</tr>
</tbody>
</table>

Table 11: Detailed overview of classification scores of the different methods compared in this chapter. The upper table presents scores of single methods, the middle table presents scores of the combination of two methods, the last row is a combination of Bag-of-Words, the most discriminative window, and exact object localisation [27].

<table>
<thead>
<tr>
<th>dog</th>
<th>horse</th>
<th>motor</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv/monitor</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>.471</td>
<td>.807</td>
<td>.639</td>
<td>.866</td>
<td>.364</td>
<td>.470</td>
<td>.503</td>
<td>.816</td>
<td>.530</td>
</tr>
<tr>
<td>[42] Loc only</td>
<td>.352</td>
<td>.679</td>
<td>.638</td>
<td>.740</td>
<td>.384</td>
<td>.401</td>
<td>.357</td>
<td>.567</td>
<td>.543</td>
</tr>
<tr>
<td>[27] Loc only</td>
<td>.231</td>
<td>.756</td>
<td>.664</td>
<td>.881</td>
<td>.296</td>
<td>.371</td>
<td>.449</td>
<td>.676</td>
<td>.602</td>
</tr>
<tr>
<td>Ours</td>
<td>.471</td>
<td>.794</td>
<td>.735</td>
<td>.897</td>
<td>.551</td>
<td>.579</td>
<td>.516</td>
<td>.770</td>
<td>.724</td>
</tr>
<tr>
<td>[42]</td>
<td>.452</td>
<td>.784</td>
<td>.697</td>
<td>.861</td>
<td>.524</td>
<td>.544</td>
<td>.543</td>
<td>.758</td>
<td>.621</td>
</tr>
<tr>
<td>Base + [27]</td>
<td>.474</td>
<td>.848</td>
<td>.736</td>
<td>.922</td>
<td>.409</td>
<td>.529</td>
<td>.577</td>
<td>.849</td>
<td>.688</td>
</tr>
<tr>
<td>Base + Ours</td>
<td>.516</td>
<td>.829</td>
<td>.725</td>
<td>.908</td>
<td>.581</td>
<td>.609</td>
<td>.571</td>
<td>.832</td>
<td>.726</td>
</tr>
<tr>
<td>Base + Ours + [27]</td>
<td>.518</td>
<td>.850</td>
<td>.778</td>
<td>.932</td>
<td>.601</td>
<td>.626</td>
<td>.831</td>
<td>.846</td>
<td>.762</td>
</tr>
</tbody>
</table>

Table 12: Comparison with results on the Pascal VOC 2010 challenge. Our method yields competitive results.
tern Recognition, China), but 0.02 MAP less than the 0.738 MAP of the National University of Singapore. We attribute the excellent performance of the other method to the use of Multiple Kernel Learning, the use of additional descriptors such as Local Binary Patterns or HOG, the use of sparse coding to create the visual vocabulary (e.g. [111]), and using a contextualized support vector machine [88], all independent improvements, applicable separately from the most telling window.

5.4.2 Joint Multi-Object Classification

We now proceed to the task our method was designed for: the joint classification of multiple objects without learning anything about the combination, i.e. we using combinations of single-object classifiers only. As a baseline we use Bag-of-Words on the whole image. We report results for the exact, part-based localisation method of [27] combined with Bag-of-Words, yielding better performance than [27] alone. We compare using our most telling window only, as in contrast to single-object classification the combination with Bag-of-Words does not improve performance for multi-object classification. We evaluate the joint classification of two classes and three classes on the Pascal VOC 2007 dataset. To ensure reliable Average Precision scores, we evaluate those pairs and triplets which occur at least 5 times in the test set.

Joint Classification of Two Objects

As can be seen in Figure 49 the most telling window significantly outperforms both Bag-of-Words and the combination of exact localisation [27] plus Bag-of-Words. While Bag-of-Words yields 0.138 MAP and exact localisation plus Bag-of-Words yields 0.156, the most telling window achieves 0.216 MAP.

Out of the 50 object pairs which occur more than 5 times in the test set, Bag-of-Words performs best on 3. For aeroplane+person the context is important, as the other two methods find many contextually implausible planes and persons. The high performance on horse+person reflects the fact that recognising the object horse implies recognising a person as well as the vast majority of horse pictures are when they are being ridden. Exact localisation plus Bag-of-Words performs best on 6 out of 70 object pairs. Among them are car+person and car+motorbike as there are many images in which both these objects are fully visible, while this method performed best on these three rigid objects individually (see Figure 48).

The most telling window performs best on 21 out of 50 object pairs. In most cases the difference with Bag-of-Words is significant. This happens often for object pairs which share context, such as plant+tv, chair+tv, chair+diningtable, car+train and car+motor. This is understandable as Bag-of-Words uses the entire image and returns substantial evidence on the usual context of an object even when the object is absent. By zooming in to the most telling window, mostly object features are used which avoids such contextual confusion. In general, using windows within the image makes the features and hence classification independent, making a combination of classifiers more meaningful.

The most telling window outperforms exact localisation plus Bag-of-Words almost always. To appreciate why, consider the top-ranked images of person+sheep and motorbike+person visualised in Figure 50. Two out of the three found per-
5.4 Evaluation

Figure 49: Results for the joint classification of two objects, evaluated on Pascal VOC 2007. Combining classifiers for Bag-of-Words results in 0.138 MAP, combining classifiers for the localization method of Felzenszwalb et al. [27] results in 0.157 MAP, combining classifiers for our most telling window method results in 0.216 MAP, significantly better than either of the two. The numbers next to each pair report the number of train/test samples in the dataset.

son+sheep images contain occlusions. The most telling window deals with this occlusion by focusing on the upper part of the body. Exact localisation has difficulties with occlusion, focusing rather on complete cows, which share many characteristics with sheep. In general, when occlusion is more frequent, the most telling window gets an advantage. For motorbike+person the individual person and motorbike scores are higher for exact localisation plus Bag-of-Words (see table 11), yet the combination is lower than the most telling window. The top-results of exact localisation plus Bag-of-Words show two images where the person is not located on the motorbike, but is standing next to the motor. This suggests that for exact localisation it is difficult to find persons riding a motor, which makes sense as such persons have a changed pose. Instead, the most telling window finds persons on motorbikes, where it in three out of the four cases ignores the difficult legs and focuses on the still recognisable upper body parts instead. The better performance on horse+person and bicycle+person in Figure 49 can be attributed to the same phenomenon, where the new method is telling a better story of the image indeed.

5.4.3 Joint Classification of Three Objects

Figure 51 shows the results for all 24 triplets that occur 5 times or more in the test set. Again, our most telling window method significantly outperforms the other methods: Bag-of-Words using the entire image yields 0.079 MAP, exact part-based localisation [27] plus Bag-of-Words yields 0.104 MAP, while the most telling window yields a result of 0.170 MAP.
**Figure 50**: Visualisations of the top ranked images for *person+sheep* and *motor+person*, for the most telling window, exact localisation [27] plus Bag-of-Words, and Bag-of-Words. Above each image it is shown if the object is present. Below each image are the probabilities per object class.
5.5 Discussion and Conclusion

This chapter introduced the most telling window for classifying multiple objects in a picture by combining single-object classifiers. The aim is to find combinations of objects as part of telling the story of pictures.

The most telling window concentrates on the most discriminative parts of an object for its recognition. The most discriminative parts can encompass the complete object, but in many occasions the most telling window describes an object

Figure 51: Results of the joint classification of three objects, evaluated on Pascal VOC 2007. Combining classifiers for Bag-of-Words results in 0.079 MAP, combining classifiers for exact localization [27] plus Bag-of-Words results in 0.109 MAP, combining classifiers for the most telling window results in 0.170 MAP, significantly better than the other two methods.

Bag-of-Words on the entire image is never leading. Exact localisation plus Bag-of-Words wins on bottle + person + sofa, while the most telling window outperforms others on bicycle + car + person, car + horse + person, and on 8 other, living room related triples. The top-ranked results in Figure 52 give an indication why. Bag-of-Words, which uses the whole image, relies heavily on the context of the living room triplets: for bottle + person + sofa this can be seen as all four images capture the context, but for three images there is only one out of three target objects present. Exact localisation plus Bag-of-Words often finds two objects, but less often three complete objects, as these images tend to be more crowded resulting in more occlusions and atypical viewpoints for some objects in the image. The most telling window has less problems with occlusion and atypical viewpoints.

The results of the multi-object classification show that the use of the most telling window yields superior retrieval rates. In fact the advantage of the most telling window has become bigger than in the case of two object classification. It outperforms Bag-of-Words as it succeeds in better combining single-object classifiers while ignoring the context which often is a source of confusion. It outperforms exact localisation plus Bag-of-Words as the most telling window focuses on those parts of the image which are still recognisable. This helps with occlusion and with interaction of objects in the picture. When the story told by a picture grows in complexity, occlusions and interactions will happen more frequently. Therefore the use of the most telling window is an important step towards generating the full narrative of an image.
Figure 52: Visualisations of the top ranked images for joint classification of three objects. Results are shown for chair+table+pottedplant and car+horse+person. Above each image the presence/absence of each object is reported (according to the dataset the upper right image of chair+table+pottedplant does not contain a chair nor a pottedplant). Below each image the probabilities of each object is displayed.
part or a collection of objects. Using an object part for recognition is advantageous when objects are occluded, changes pose, or have a characteristic part, such as a face. Using a collection for recognition is advantageous when they are in a group and hard to distinguish individually, such as sheep in a flock. This shows that a perfect segmentation, finding the exact contours of the object, is not only hard but also unnecessary for recognition; often object parts or object collections are more informative. Therefore we designed the most telling window to flexibly adjust its focus depending on the object type and image composition.

For single object classification the most telling window significantly outperforms Bag-of-Words. Comparing exact localisation [27] plus Bag-of-Words, introduced by [42], with the most telling window plus Bag-of-Words shows that the exact localisation performs better on rigid objects, while the most telling window is superior on non-rigid objects. More importantly, these methods are complementary as the most telling window plus Bag-of-Words yields an overall score of 0.674 MAP, while adding exact localisation [27] brings the score to 0.707 MAP. On the Pascal VOC 2010 dataset this combination results in the best performance of 5 objects and yields overall 0.711 MAP, just below the highest score of 0.738 MAP of the National University of Singapore. This shows that the most telling window is a useful addition to current object classification techniques on single object classification.

For multi-object classification the advantages of the most telling window become more apparent. For the joint classification of two objects Bag-of-Words achieves 0.138 MAP, exact localisation [27] plus Bag-of-Words achieves 0.157 MAP, while the most telling window yields a performance of 0.216 MAP. For the joint classification of three objects the differences grow further: Bag-of-Words yields 0.079 MAP, exact localisation plus Bag-of-Words yields 0.109 MAP, while the most telling window yields 0.170 MAP. By experiment and visual illustrations, we have shown that the most telling window outperforms Bag-of-Words through decorrelating the features by focusing on the most discriminative parts of the object, which also ignores the often confusing context. The most telling window outperforms exact localisation plus Bag-of-Words as the most telling window is robust to occlusions, atypical views, and pose changes, which occur more often as the picture tells a more complex story. We conclude that the use of the most telling window is an important step towards generating the full narrative of an image.

We emphasize that the most telling window combines objects by straightforward combination of the outcomes of classifiers trained on single objects. In that sense, our method is capable to tell stories it has not yet seen. Furthermore it is able to address recognition problems which occur as objects interact, such as occlusion and pose changes, a topic very recently addressed by [25]. Other than the reference, where for each combined object pair a few training samples need to be found and learned, once our method has learned single-object classifiers it is capable of combining them without any additional learning. As combinations become more and more abundant, the advantage of our method grows with the complexity of the story that pictures tell.
CONCLUSION

6.1 SUMMARY OF CONTRIBUTIONS

This thesis started out with the current method of choice for single-object image classification: Bag-of-Words. We performed a thorough analysis of this system and made a variety of improvements. The end result is a state-of-the-art system for object recognition, as well as a system which jointly classifies multiple objects. The latter is a significant step towards generating the full narrative of an image.

Chapter 2 reviewed and created various techniques to accelerate Bag-of-Word based concept classification. We improved the computational efficiency of densely sampled SIFT and SURF. Furthermore, our experiments resulted in several recommendations for practitioners: (1) From the descriptors SIFT [53], SURF [6], DAISY [92], and Semantic Textons [79], we found that SIFT is the descriptor of choice for classification accuracy and that SURF is the descriptor of choice for computational efficiency. (2) For a visual vocabulary, we found that a Random Forest [63] when used in combination with Principal Component Analysis is much faster than k-means, and yet performs similarly. And while it does require labelled data for learning, the vocabulary is equally powerful for classes on which it is not learned. This makes a Random Forest the visual vocabulary of choice. (3) For the Spatial Pyramid we found that only a division into horizontal regions improves performance, which is a crude floor-object-sky distinction. This makes intuitively sense as flipping a picture horizontally still displays the same scenery. (4) As kernel for the Support Vector Machine we compared a RBF-kernel, $\chi^2$ kernel, and a Histogram Intersection kernel. We found the $\chi^2$ kernel to be superior for accuracy. The Histogram Intersection kernel is much faster if the approximation of the SVM classification function is used [54], but gave slightly lower results in the experiments of this chapter. However, we later found that when a visual word is extracted from each pixel, the accuracy difference between the $\chi^2$ kernel and Histogram Intersection kernel disappears. Hence when accuracy is important, we recommend to use extremely dense sampling and the SVM with Histogram Intersection kernel.

The increased computational efficiency of the Bag-of-Words pipeline opened up new applications for automatic visual concept classification. On the one hand we were able to create a real-time visual concept classification system, which would be able to automatically tag 10,000 images per minute for 20 classes on 5 desktop computers, where each additional computer allows the tagging of 20 extra classes. This throughput is sufficient to automatically label all images as they are uploaded to Flickr. On the other hand our Bag-of-Words pipeline facili-
tates extreme dense sampling of visual words, which enables the exploration of region-based techniques for classification.

In chapter 3 we investigated the visual extent of an object in terms of the object and its surround, and in terms of the object interior and the object border. We performed our investigation from two perspectives: (a) Without knowing the object location we determined and visualised where the support for object classification resides. (b) Assuming known object locations we investigated the relative potential of the object and its surround, and of the object border and the object interior. We found the following: (1) In the situation where the object location is not known, support for the classifiers is found throughout the whole image, supporting the notion that context facilitates image classification. (2) We found the surround to adequately distinguish between the groups of classes furniture, animals, and land-vehicles, but for distinguishing categories within one group the surroundings become a source of confusion. (3) The physically rigid classes plane, bike, bus, car, and train are recognised by their interior boundaries and shape, not by texture. In contrast, the non-rigid animals dog, cat, cow, and sheep are primarily recognised by texture, i.e. fur, as their projected shape varies greatly. (4) We confirmed an early observation from human psychology [7]: An object in isolation is recognised equally well as an object in proper context, suggesting that once the object has been located the surroundings help little for the identification of the object. (5) We showed that isolating the object leads to a significant increase in classification accuracy. This last insight inspired us to pursue the isolation of the object location.

Chapter 4 addressed the problem of finding good candidate locations within an image. Our structured sampling combines the advantages of both segmentation and exhaustive search. Like segmentation, it uses the image structure to guide the sampling process. Like exhaustive search, it aims to capture all possible object locations. Instead of a single sampling strategy, structured search combines the locations a set of diversified strategies to deal with as many image conditions as possible. Specifically, we use a variety of different colour spaces to deal with different scene conditions, and we use a variety of similarity measures to account for a variety of reasons why pieces of an image belong together. This results in a data-driven, class-independent structured sampling which yields a small set of high-quality locations. We showed that the resulting set outperforms exhaustive search both in terms of quality and quantity. Furthermore, it enabled the use of powerful Bag-of-Words implementation for object recognition, resulting in significant improvements of 9 out of 20 classes on the Pascal VOC 2010 detection challenge. Finally, we showed that the object hypothesis set generated by our structured search is close to optimal in terms of object recognition accuracy for our Bag-of-Words framework when generating 10,097 locations.

In addition to finding complete objects, our selective search strategy theoretically also captures concepts like sand and water, as well as object parts and object collections. Hence our selective search strategy is highly versatile and its use extends beyond object recognition, as evidenced by its use in chapter 5.

In chapter 5 we predicted the joint presence of multiple objects as a step towards generating the story of an image. Multi-object classification only scales if it is done through the combination of single-class classifiers due to the exponential combinatorics which arise from classifying n objects from c classes. To be able to combine single-class classifiers, they should be largely independent which can
be achieved by using localised features from the image. An important insight of this chapter is not to focus on the whole object, but on the most discriminative parts of an object for its classification: the “most telling window”. By focusing on the “most telling window”, we account for variations in image composition that are increasingly common as the number of objects in an image grow: (1) Aggregations of objects may be better recognised as a complete group, (2) Non-visible parts of occluded objects should be ignored. (3) Objects that change pose, possibly because of an interaction, may still retain some characteristic features through which they can be recognised. The end result is a system that obtains strong results for the joint prediction of two and three object classes.

6.2 GENERAL CONCLUSION

This thesis brought several improvements to object recognition and image understanding. But looking at the bigger picture, where exactly did this brought us?

In terms of Bag-of-Words this thesis made almost full circle: The extremely successful Bag-of-Words method impoverished image analysis model by removing the object location and replacing it by local details scattered throughout the image, captured by highly invariant descriptors. This enabled the use of strong machine learning techniques, favouring quantity of annotation over quality of annotation. In this thesis, to improve object classification accuracy further, we reintroduced the object location on the basis of a data-driven sampling strategy. We kept strong machine learning techniques and invariant descriptors, yet reinstated a more precise appearance model of the object, putting again more emphasis of the quality of annotation. And while the object locations generated in Chapter 4 are suboptimal in terms of object coverage, they are close to optimal in terms of object recognition performance. This suggests the necessity of a future shift towards more variant descriptors, which would complete the circle to return Bag-of-Words to a computer vision model with a strong structure.

In this thesis we enabled the joint recognition of multiple object in one image as a step towards the narrative of an image. But the next challenge towards generating stories has to be the inclusion of spatial relations between the concepts. Spatial relations are important to distinguish between a person in a car and a person in front of a car, which would be a completely different type of picture. Spatial relations can also be used to improve identification. This can be appreciated an abstract landscape painting: a greenish blob can only be identified as water through its relative position in the image. Another major step will be the inclusion of action relations or interaction relations leading to the identification of verbs in the narrative of a picture. Verbs are essential in creating proper sentences. They convey rich semantics such as a person driving a car and a car stopping for a person, both different stories. The results of this thesis, combined with spatial relations and (inter-)action relations, open the road from the where and the what to the full story of the image. Nevertheless, the three object stories which we are able to generate now can already leave a powerful impression.

Sand, Sea, Cocktail.


SAMENVATTING

Kijk eens om je heen. Overal zie je objecten die je onmiddellijk kunt benoemen, zoals een tafel, een stoel, of een lamp. Maar zo makkelijk als dit is voor ons mensen, zo moeilijk is dit voor een computer. Dit komt doordat een computer niets meer is dan een grote rekenmachine, die goed werkt wanneer een taak exact geformuleerd kan worden, maar het een stuk lastiger krijgt wanneer de uit te voeren taak onduidelijk of ambigu is. Een voorbeeld van een exacte taak is spellingcontrole, waar elk geschreven woord wordt getoetst aan het woordenboek. Iets minder exact is schaken, wat weliswaar een spel is met een kleine set van strikte regels, maar waarbij het soms moeilijk is om de positieve en negative punten van een stelling exact te beoordelen. Hoe anders is het met visuele perceptie. Wat maakt bijvoorbeeld een auto tot een auto? De eerste ingeving is dat het rijdt en wielen heeft. Maar dat geldt ook voor een bus en voor een driewieler. Een auto heeft dus precies vier wielen, een overdekte cabine en is normaal geschikt voor tot vijf personen. Maar is een auto ontstaan van wielen geen auto? Verder is een cabriolet niet overdekt en past geen mens in een speelgoedauto. Gaat het dan om de vorm? Maar een Formule-1 auto lijkt qua vorm weinig op een Lada, om nog maar te zwijgen over de vele verschijningsvormen van speelgoed auto’s. Het blijkt dus moeilijk om te omschrijven hoe een “auto” er precies uitziet. Het gebrek aan een strikte definitie, en het gebrek van vaste verschijningsvormen van objecten in het algemeen, maakt het moeilijk voor een computer om deze te herkennen.

Niettemin zou het nuttig zijn als een computer de visuele wereld om ons heen kan interpreteren, aangezien visuele perceptie een belangrijke basis vormt voor ons begrip van de wereld. Toepassingen van onderzoek in deze richting zijn het automatisch doorzoeken van persoonlijke fotocollecties, die met de introductie van digitale camera’s snel zijn gegroeid, of van afbeeldingen op het internet. Op langere termijn valt te denken aan een robot die rondloopt, observeert en handelingen kan verrichten in onze wereld. Uiteindelijk is de hoop dat werk aan visuele perceptie door een computer ook inzichten oplevert in de wijze waarop de mens de wereld om zich heen waarnemt, interpreteert, en analyseert.

De eerste vraag die opkomt bij het interpreteren van beelden is: In welke termen wil men beelden uiteindelijk herkennen? In dit proefschrift is het antwoord dat beelden uiteindelijk een verhaal vertellen. Vakantiefoto’s verhalen over alle belevenissen en bezienswaardigheden van de vakantie. Dit kan bijvoorbeeld een bezoek aan het Forum Romanum zijn of een dorpsfeest in Arriondas. Een journaliste probeert met foto’s zijn verhaal in beelden vast te leggen. Dit kan een oorlog zijn of een publiek schandaal. Een foto advertentie vertelt waarom een blender beter blendt of waarom het aroma van deze koffie het aardse overstijgt. In het al-
gemeen is het verhaal wat een beeld vertelt zowel gebaseerd op de verwachtingen van de toeschouwer als op de inhoud van het beeld zelf. Dit proefschrift handelt uitsluitend over de inhoud van het beeld.

Vanuit het perspectief van een computer kunnen de verhalen die een beeld kan vertellen geordend worden van simpel naar complex. Het simpelste verhaal is het benoemen van de omgeving, zoals een strand of gebergte. Een iets moeilijker verhaal is het benoemen van een enkel object, zoals een fiets of kat. Nog complexer is de benoeming van meerdere objecten, zoals paard en mens, of kind en bal. Spatiële relaties zijn nodig om bijvoorbeeld een onderscheid te maken tussen een man in de auto en een man voor de auto. Tenslotte zijn er nog acties en interacties, veelal geduid door werkwoorden. Deze maken kleine verhalen mogelijk zoals vrouw springt uit vliegtuig of man rent weg voor hond. De beeldherkenningsgemeenschap heeft de afgelopen jaren veel voortgang geboekt met het identificeren van individuele objecten, oftewel het genereren van verhalen van één woord. Dit proefschrift geeft een analyse van de recente, grote vooruitgang op dit gebied en probeert inzicht te geven in de redenen van het succes van de huidige technieken. Vervolgens wordt een stap gezet in de richting van complexere verhalen door het mogelijk te maken om meerdere objecten binnen een beeld te herkennen.

De oorsprong van de huidige vooruitgang in de beeldherkenning is contra-intuitief. Als je je afvraagt hoe een mens een object herkent, is de eerste gedachte dat je moet beginnen met het bepalen van de contouren van een object om vervolgens aan de vorm zijn identiteit af te leiden. Hoewel dit lang is geprobeerd, komt de recente vooruitgang juist voort uit het loslaten van dit idee. De nieuwe technieken houden geen rekening met de locatie of ruimtelijke samenhang van de onderdelen van het beeld. Ze gebruiken in plaats daarvan heel kleine details uit het beeld om vervolgens met sterke leeralgoritmes te bepalen welk object is afgebeeld. Dit werkt als volgt: Een plaatje wordt opgedeeld in kleine vierkanten en voor elk vierkant wordt bepaald of het “hoekig”, “streperig”, een “t-splitsing”, “egaal”, etc. is. Waar elk vierkant zich in het plaatje bevindt doet in deze representatie niet ter zake. Vervolgens kan bijvoorbeeld worden afgeleid dat als een beeld veel stukjes “hoekig” bevat, dat er dan waarschijnlijk een kast is afgebeeld. Veel “streperigheid” duid op de aanwezigheid van gras, wat gecombineerd met “vlekkelig” suggereert dat er een koe is afgebeeld.

Een probleem van de hierboven geschetste methode is dat deze erg veel rekenkracht kost. Hoofdstuk 2 behandelt het versnellen van de achterliggende technieken en de verhouding tussen snelheid en kwaliteit, zodat een afweging kan worden gemaakt tussen hoe goed het algoritme beelden kan erkennen en hoeveel rekenkracht daarvoor nodig is. Het resultaat van dit hoofdstuk is tweeledig. Enerzijds levert dit hoofdstuk een systeem op dat het mogelijk maakt om alle foto’s die op de foto website Flickr worden gezet met slechts 5 computers te categoriseren. Anderzijds kunnen de versnelde methodes het beeld fijnmaziger analyseren, wat het mogelijk maakt om niet alleen een goede representatie te maken van het hele beeld, maar ook van relatief kleine delen binnen het beeld.

In hoofdstuk 3 wordt nader onderzocht waarom de geschetste beeldherkenningsmethode zo goed werkt. Het vermoeden bestaat dat een koe wordt herkend aan “streperigheid” en “vlekkelig”, maar het is onduidelijk welke elementen van het beeld precies worden gebruikt: Wordt een object geïdentificeerd vanwege zijn eigen uiterlijk (zoals “vlekkeligheid” in dit voorbeeld) of wordt een object meer geïdentificeerd vanwege de omgeving waar het zich bevind (zoals in de
“streperigheid” van gras)? In dit hoofdstuk meten wij waar precies de informatie vandaan komt. De omslag van dit proefschrift visualiseert voor enkele beelden de delen waar “kattigheid” gemeten wordt. Geel wordt beschouwt als positief bewijs voor de aanwezigheid van een kat, blauw als negatief bewijs. De som van geel en blauw is de totale hoeveelheid “kattigheid” in de afbeelding volgens de computer. De belangrijkste vinding in dit hoofdstuk is dat, ondanks het feit dat het recente succes van beeldherkenning te danken is aan het loslaten van de object locatie, het herintroduceren van de objectlocatie gewenst is.

Hoofdstuk 4 gaat daarom in op de vraag hoe een object het beste gelokaliseerd kan worden — hier wordt dus een overgang gemaakt van de vraag wat er op het beeld staat naar waar het op het beeld staat. De meest succesvolle technieken op het gebied van het lokaliseren en identificeren van complete objecten doen dit door (bijna) alle mogelijke locaties binnen een beeld te doorzoeken met gebruikmaking van een zoekraster. Meestal worden per plaatje zo’n 100.000 tot 1.000.000 locaties afgezocht naar een object, wat erg veel rekenkracht vereist. Een andere richting binnen de beeldverwerking is segmentatie, wat als doel heeft om het beeld op te delen in coherente regio’s zodat elke regio overeenkomt met een enkel object. Deze methode levert ongeveer 10 tot 100 locaties op per beeld, maar veel objectlocaties worden op deze manier overgeslagen. Er zijn namelijk een hoop verschillende redenen waarom een object één geheel vormt: soms is kleur genoeg, zoals bij een witte muur, in andere gevallen heb je alleen textuur nodig en maakt kleur de taak juist moeilijker, zoals bijvoorbeeld bij een lapjeskat. In dit hoofdstuk wordt daarom voortgebouwd op de voordelen van beide lokalisatie technieken: net zoals segmentatie gebruikt dit hoofdstuk de structuur van het beeld om het zoekproces te leiden, maar het doel is om alle mogelijke object locaties vinden. Daarom worden meerdere complementaire segmentaties gegenereerd door te variëren in onder andere het gebruik van kleur en textuur. Als resultaat levert dit 1.000 tot 10.000 locaties op die meer objecten bevatten dan bij een segmentatie, en die van hogere kwaliteit zijn dan locaties verkregen via een zoekraster. Het gebruik van deze relatief kleine set van locaties maakt het mogelijk om krachtige identificatie technieken te gebruiken. Het resulterende systeem voor objectherkenning en -localisatie levert vooral verbeteringen op voor niet-rigide objecten zoals kat, hond, en plant.

In hoofdstuk 5 gebruiken we de locaties uit hoofdstuk 4 om de beeldherkenning naar een nieuw niveau te tillen: we willen niet één object, maar meerdere objecten binnen een plaatje herkennen om zo het “verhaal” van een afbeelding uit te kunnen breiden. In tegenstelling tot het identificeren van een compleet object (gebruikelijk binnen de beeldverwerking) introduceren we het “meest vertellende beelddeel” van een object. Het idee is dat een object wordt herkend aan dat deel van het beeld wat het meest veelzeggend is voor dat object. Concreet betekent dit dat een kat kan worden herkend aan zijn kop terwijl zijn lijf kan worden genegeerd: het lijf van een hond lijkt tenslotte veel op dat van een kat. Het gebruik van het meest vertellende beelddeel heeft veel voordelen ten opzichte van het gebruik van complete objecten wanneer meerdere objecten geïdentificeerd moeten worden: meer objecten in een beeld betekent vaak dat het beeld drukker is, wat de kans vergroot dat delen van een object achter andere objecten verdwijnen. Als je bijvoorbeeld op zoek bent naar een auto, maar deze staat half achter een bus, dan ligt de focus van het meest vertellende beelddeel op het zichtbare deel van de auto terwijl het zoeken naar een volledige auto fout gaat. Ook kan het voorkomen
dat het uiterlijk van objecten verandert omdat ze een interactie aangaan. Een persoon op een paard ziet er bijvoorbeeld anders uit dan een persoon op de grond. In deze situatie is het wederom beter om de focus te leggen op het herkenbare bovenlijf van de paardrijder dan om de benen van de persoon te proberen mee te nemen. Het resultaat van dit hoofdstuk is een systeem dat tot drie objecten in een beeld nauwkeurig kan herkennen.

Aan het einde van dit proefschrift is de beeldherkenning naar een nieuw verhalend niveau getild: in plaats van slechts individuele objecten te kunnen herkennen, is het nu mogelijk om meerdere objecten binnen een beeld te identificeren. Natuurlijk is er nog een lange weg te gaan naar het automatisch bepalen van het volledig verhaal van een afbeelding. Hiervoor zal het nodig zijn om ook spatiële relaties, acties en interacties te kunnen herkennen. Niettemin kan het noemen van slechts drie objecten al een krachtige impressie oproepen.

Zand, Zee, Cocktail.
DANKWOORD


Verder wil ik mijn co-promotor, Remko Scha, bedanken. Remko was de begeleider van mijn master scriptie en hij heeft mij geïntroduceerd als promotiekandidaat voor een samenwerking tussen hem en Arnold. Als buitenstaander in de beeldverwerking zorgde hij veelal voor frisse invalshoeken, ongehinderd door mode in de beeldverwerking. Discussies met zowel Remko als Arnold leverden altijd filosofische hoogtepunten op.

Ook wil ik mijn wiskundeleraar, Jaap de Jonge, bedanken, die mij al vroeg kennis deed maken met creatief oplossen van problemen.


Nicu, thanks for giving me and Vika the opportunity to continue my work in Trento, between the snowy peaks of the dolomites. Jacopo, thanks for the warm welcome and all the cold beers.


Tenslotte wil ik mijn ouders bedanken, die mij altijd overal in hebben gesteund, en mijn broer, die altijd klaarstaat en als een van de eerste ons bezocht in Italië. Finally, Vika, thank you for all our wonderful trips. You are where my heart is.