Lappeenranta
University of Technology

Ekaterina Riabchenko

GENERATIVE PART-BASED GABOR OBJECT DETECTOR

Thesis for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism in the Auditorium 1383 at Lappeenranta University of Technology, Lappeenranta, Finland on the 25th of September, 2015, at noon.

Acta Universitatis
Lappeenrantaensis 661

Supervisor Professor Joni-Kristian Kämäräinen

Machine Vision and Pattern Recognition Laboratory
School of Engineering Science
Faculty of Technology Management
Lappeenranta University of Technology
Finland

Reviewers Professor Aleš Leonardis
School of Computer Science
The University of Birmingham
United Kingdom

Docent Esa Rahtu
Department of Computer Science and Engineering
The University of Oulu
Finland

Opponents Professor Aleš Leonardis
School of Computer Science
The University of Birmingham
United Kingdom

Dr. Tech. Jorma Laaksonen
Department of Computer Science
Aalto University
Finland

ISBN 978-952-265-851-7
ISBN 978-952-265-852-4 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

Lappeenrannan teknillinen yliopisto
Yliopistopaino 2015

Preface

The work presented in this thesis was undertaken in the Machine Vision and Pattern Recognition Laboratory of Lappeenranta University of Technology and the Department of Signal Processing of Tampere University of Technology during the years 2011-2015.

I want to express my gratitude to my supervisor, Professor Joni Kämäräinen, for his supportive guidance as well as for providing me with facilities and financial support for my research.

I thank the reviewers, Aleš Leonardis and Esa Rahtu, for their critical reading of the manuscript and valuable comments.

I also wish to thank my colleagues at the Machine Vision and Pattern Recognition Laboratory, Lappeenranta University of Technology: Jukka Lankinen, Natalia Strokina and Lauri Laaksonen, and the Department of Signal Processing Tampere University of Technology: Ke Chen, Fatemeh Shockrollahdi, Katariina Mahkonen, Antti Hietanen, Yan Lin and Yuan Liu for their constant support and insightful comments on my work. Special thanks to Tarja Nikkinen, Riitta Laari and Ilmari Laakkonen for technical and organizational support.

Lappeenranta, September 2015

Ekaterina Riabchenko

Abstract

Ekaterina Riabchenko
Generative Part-Based Gabor Object Detector
Lappeenranta, 2015
107 p.

Acta Universitatis Lappeenrantaensis 661
Diss. Lappeenranta University of Technology

ISBN 978-952-265-851-7
ISBN 978-952-265-852-4 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

Object detection is a fundamental task of computer vision that is utilized as a core part in a number of industrial and scientific applications, for example, in robotics, where objects need to be correctly detected and localized prior to being grasped and manipulated. Existing object detectors vary in (i) the amount of supervision they need for training, (ii) the type of a learning method adopted (generative or discriminative) and (iii) the amount of spatial information used in the object model (model-free, using no spatial information in the object model, or model-based, with the explicit spatial model of an object). Although some existing methods report good performance in the detection of certain objects, the results tend to be application specific and no universal method has been found that clearly outperforms all others in all areas.

This work proposes a novel generative part-based object detector. The generative learning procedure of the developed method allows learning from positive examples only. The detector is based on finding semantically meaningful parts of the object (i.e. a part detector) that can provide additional information to object location, for example, pose. The object class model, i.e. the appearance of the object parts and their spatial variance, constellation, is explicitly modelled in a fully probabilistic manner. The appearance is based on bio-inspired complex-valued Gabor features that are transformed to part probabilities by an unsupervised Gaussian Mixture Model (GMM). The proposed novel randomized GMM enables learning from only a few training examples. The probabilistic spatial model of the part configurations is constructed with a mixture of 2D Gaussians. The appearance of the parts of the object is learned in an object canonical space that removes geometric variations from the part appearance model. Robustness to pose variations is achieved by object pose quantization, which is more efficient than previously used scale and orientation shifts in the Gabor feature space. Performance of the resulting generative object detector is characterized by high recall with low precision, i.e. the generative detector produces large number of false positive detections. Thus a discriminative classifier is used to prune false positive candidate detections produced by the generative detector improving its precision while keeping high recall. Using only a

small number of positive examples, the developed object detector performs comparably to state-of-the-art discriminative methods.

Keywords: generative learning, part detector, part-based object class detector, Gabor features, Gaussian mixture model, hybrid generative-discriminative detector

BoW	Bag of Words
CNN	Convolutional Neural Network
D	Discriminative
DF	Deep Features
DNN	Deep Neural Network
DPM	Deformable Part-Based Model
DOD	Discriminative Object Detector
DoG	Difference of Gaussians
EER	Equal Error Rate
EM	Expectation Maximization
G	Generative
GEM	Greedy Expectation Maximization
GMM	Gaussian Mixture Model
GOD	Generative Object Detector
G-DOD	Discriminative Object Detector in generative mode
HOG	Histogram of Oriented Gradients
IP	Interest Point
LBP	Local Binary Pattern
pdf	Probability Density Function
RGB	Red Green Blue color space
ROC	Receiver Operating Characteristic
SIFT	Scale Invariant Feature Transform
WHO	Whitened Histogram of Orientations

$\boldsymbol{a}$	vector
$\boldsymbol{A}$	matrix
$\boldsymbol{A}^H$	conjugate (Hermitian) transpose of $\boldsymbol{A}$
$\boldsymbol{I(x, y)}$	intensity image
$D(x, y, \sigma)$	Difference of Gaussians
I_i	integral image
$s(x, y)$	cumulative row sum
p	crossing point between adjacent Gabor filters
k	scaling factor for Gabor filter frequencies in a bank

γ	Gabor filter sharpness along the major axis
η	Gabor filter sharpness along the minor axis
M	number of filters with different frequencies in a bank
N	number of filters with different orientations in a bank
f_{max}	highest central frequency of a Gabor filter in a bank
θ	orientation of a Gabor filter
$\psi(x, y)$	Gabor filter in a spatial domain
$\Psi(u, v)$	Gabor filter in a frequency domain
$r(x, y, f, \theta)$	Gabor response for $I(x, y)$
$N(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$	multidimensional Gaussian distribution
$\boldsymbol{G}$	feature matrix of Gabor responses
$\boldsymbol{g}$	multiresolution Gabor feature vector
$p(x, y)$	joint probability of random variables x and y
$p(x\|y)$	conditional probability of random variables x given y

Chapter I

Introduction

1.1 Motivation

Computers are used in many areas of human life and are especially successful when applied to areas demanding heavy computation (e.g. simulation of various processes) where computers greatly outperform humans. The central role played by information technology in modern life has naturally led to a desire to equip computers with the ability to see and understand the perceived information. The ability of vision that most humans have and use every day effortlessly, has turned out to be a challenging task for computers. People can recognize objects regardless of their viewpoint, the position of the object in the image or the viewing conditions (fog, shadow, etc.) without particular effort, but machines tend to have problems even in relatively controlled conditions. The main challenges facing computer vision can be categorized based on their source. One challenge comes from the camera: sensor noise and lens distortions. Another major challenge relates to the fact that in machine vision a 3D scene is captured in 2D losing information in the process and producing problems related to viewpoint and occlusions. External factors, such as lighting or background clutter, also have a strong influence on the performance of computer vision systems. However, variation of an object class appearance from image to image (object class detection) or single object from view to view (single object detection) is one of the most influential factors in solving vision tasks. Hence a good automated vision system should be computationally efficient and general enough to capture natural appearance variation of an object or a class of objects but discriminative enough not to confuse it with either the background or other objects.

The ultimate goal of computer vision is scene understanding with close to human perception (Figure 1.1). For example, given an image of a scene an automatic system should be able to determine the classes of the objects present in the image, their locations and properties (color, sitting/standing/walking, frontal/side/rear view etc).

Even though the final goal of computer vision is general scene understanding, machine vision approaches have generally broken the task into parts. For example, some methods provide information about which objects are in the image (classification task) [97, 103]

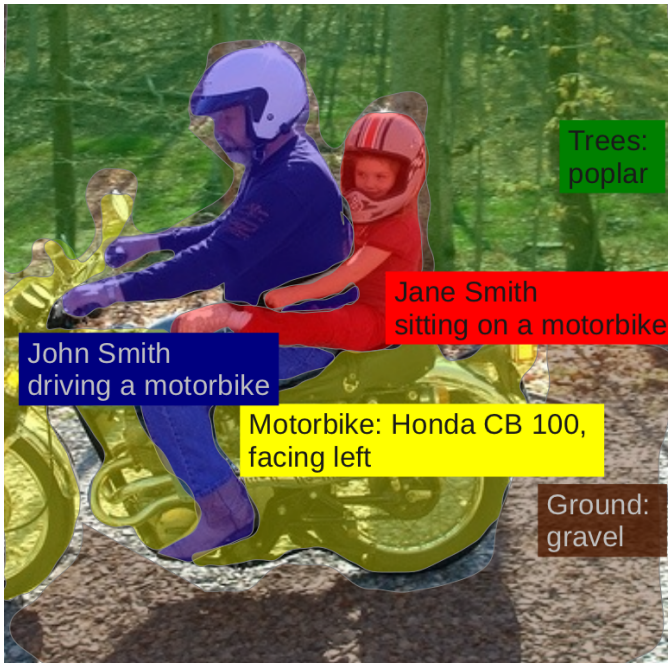


Figure 1.1: An example of scene understanding.

and are not interested in the exact locations, whereas others define objects' locations with tight bounding boxes (detection task) [78, 55, 140] or by labelling pixels belonging to the object (segmentation task) [117, 11, 92]. Computer vision methods also differ by the required amount of supervision, i.e., how much of additional data is provided during training. In unsupervised approaches only a set of images is given to the system as an input [31], semi-supervised approaches also provide labels, together with training images [56], and supervised methods utilize object locations in the form of bounding boxes or segmentation masks as additional input [55, 140, 106].

The task of arbitrary visual class detection is far from being solved, but certain tasks of object detection in restricted conditions are almost solved. For example, face detection implemented in modern cameras [177] enables the camera to focus on human faces and sometimes even take a picture at the moment a person is smiling. Pedestrian detection [66, 6] has been implemented in some top-of-the-line cars to prevent accidents involving pedestrians.

A number of different object detection methods exist: part-based [56, 183, 51], model-free [13, 44, 158], generative [34, 50, 186] and discriminative [55, 157, 1]. These approaches have their advantages and disadvantages, and there is no one superior method capable of overcoming all the computer vision challenges. Recently discriminative methods have been the subject of great research interest [39, 55], especially with the development of methods based on neural networks [97, 69]. However, generative detectors can learn object representation without negative examples, which seems a more natural way of learning. Thus, this work focuses on building an efficient general object class detector

that searches for predefined objects in given images and produces their locations. In the proposed generative part-based object detection algorithm, both bounding boxes containing the whole object and manually-labelled object parts are used to train the detection model. Therefore, the developed part-based detector employs both local discriminative appearance of parts and their spatial arrangement.

1.2 Author's Contribution

The developed methods are reported in four peer-reviewed conference papers: Publication I [141], Publication II [140], Publication III [142] and Publication IV [143], and one journal article [139].

Publication I introduces a part detector where the object parts are modelled with biologically inspired Gabor features [65], which have been successfully used in many vision applications [41, 74, 192]. To exclude the effect of geometric distortions on the object part appearance, all training images are aligned to the same frame, "mean object space", prior to feature extraction. Images are aligned using similarity transformation matching their parts' locations. In order to reduce the dimensionality of the features and provide a specifically optimized descriptor for each object part, a randomized Gaussian mixture model is employed in forming the appearance model.

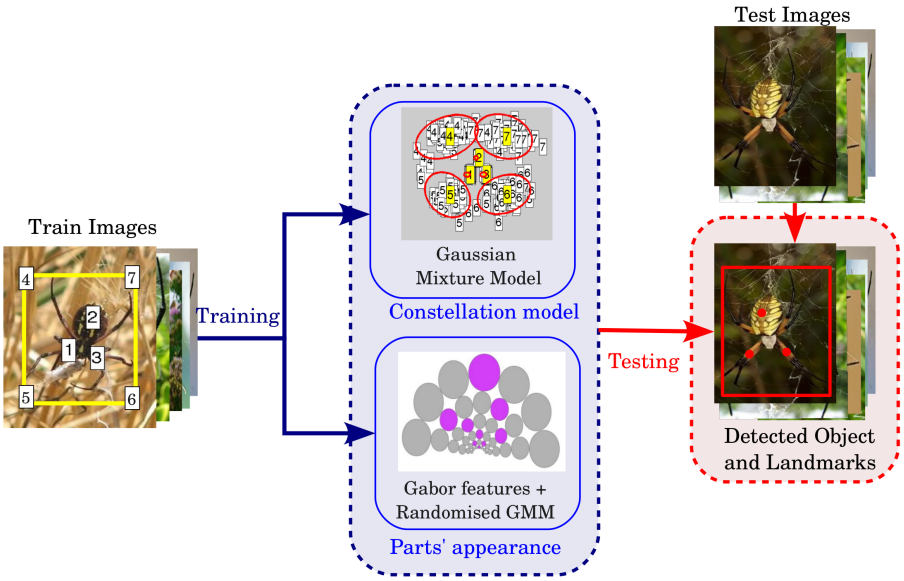


Figure 1.2: Workflow of the developed generative part-based object class detector.

Publication II is devoted to development of a generative part-based object class detector (Figure 1.2) with a fully probabilistic model. The detector uses privileged information in the form of manually annotated object parts with semantic meaning (the part detector from Publication I) and is thus a strongly-supervised method. The mean object space,

introduced in Publication I, is also used in the object detector. In this mean space the object's spatial structure becomes undistorted (Figure 1.2: Constellation model block) and is modelled along with the relative locations of the bounding box corners by the Gaussian mixture model. The final object model is robust to occlusions and can provide information about object pose in the image.

During testing, the object appearance model produces likelihood maps, which are then sampled for global maxima, candidate locations of object parts, with a consecutive suppression procedure. The final step of the object detection is the search for a feasible object hypothesis (the required number of hypotheses can be predetermined) when candidate locations are pruned using a constellation model and prior information about data statistics. As the developed detector is generative and based on likelihood scores rather than probabilities it produces a lot of false positive detections (i.e. has low precision with high recall). Therefore in **Publication III** false positive detections are re-scored and further pruned by the state-of-the-art discriminative object classifiers.

Publication IV investigates a color normalization procedure based on part-based object alignment in the color space, i.e. annotated object color regions represented as 3D points in the RGB space are aligned to form tight clusters. Consequently, objects from the same class obtain similar photometric appearance. This normalization procedure makes color a more stable cue, increasing its value for visual class detection tasks.

The **journal article** introduces an interesting property of modern datasets, the quantised poses in which the objects appear in the images. This property is caused by the laws of physics and common sense, e.g. trees, doors, buildings are vertically oriented while vehicles moving on or parallel to the ground are generally horisontally oriented and thus most of the objects occur in images in their usual orientation.

1.3 Outline of the Thesis

The thesis is organized as follows:

Chapter 2 presents some of the most common problems of computer vision, followed by an overview of various image databases. Chapter 2 also introduces popular image features and generative and discriminative approaches to object detection and classification.

Chapter 3 describes a generative part detector based on Gabor features and a Gaussian mixture model. An extensive description of Gabor features is given in the chapter as well as a novel randomization procedure, a randomized Gaussian mixture model that allows learning of the appearance model of the object parts with fewer training samples.

Chapter 4 introduces a part-based object class detector based on the part detector from Chapter 3. This chapter also presents incorporation of prior knowledge, such as the object spatial structure in the training images, object pose and bounding box statistics, into the object detection pipeline.

Chapter 5 develops a generative-discriminative hybrid approach to object detection and classification. The hybrid method uses the strengths of both generative and discriminative methods by applying them consecutively, which solves the problem of excessive false positives from the generative detector but still allows learning from positive examples.

True and false positive detections of the generative method are used as positive and negative examples in training of the discriminative method. Chapter 5 also introduces an object class specific color normalization procedure that increases photometric consistency of the images in a class by aligning class specific colors in a 3D RGB space.

Chapter II

Literature Review

One of the key problems of computer vision is the need for invariant object class and location predictions. Predictions should be invariant to different types of input image transformations, such as changes in object pose and its non-rigid transformations: translation and changes in orientation and scale of an object; changes in viewpoint; variations in nature, intensity and position of a lighting source. Another challenge is to recognize objects even if they are occluded. In many cases, local features, i.e. features obtained from image patches, provide a solution to these problems. This chapter describes the most common challenges, features and approaches to object detection. Evolution of the datasets widely used in visual class detection is also presented.

2.1 Object Detection Pipeline

Before presenting the object detection pipeline, the term object detection should be defined. While a classification method produces only a class label for an unseen image, a detection method assigns a label to a certain area in the image, related to the object's location. Potentially, object detection can also give information about an object's pose in the image, in addition to its location, achieving a deeper level of scene understanding.

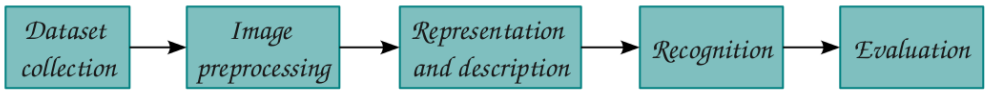


Figure 2.1: A general object detection pipeline.

Figure 2.1 presents a general pipeline followed in object detection. At the beginning of the process data is collected into a dataset like the UIUC car dataset [2], Caltech-101 [51] or ImageNet [148]. Depending on the application, images are either taken by researchers or collected from various internet resources, e.g. Flickr or Google Images. The images are then provided with the required ground truth annotations. Traditionally, object detection

annotations include class labels and tight bounding boxes, defining object location in the images, for all images in the dataset. Evolution of object detection datasets is investigated in Section 2.3.

Dataset collection can be followed by image preprocessing. The goal of preprocessing is image enhancement. The most common preprocessing steps are related to de-noising, changes in contrast and lighting or color normalization. For example, color information is an important cue in object detection and especially segmentation. However, color variation even of the same object from image to image can be rather large (Figure 2.2 left). Preprocessing in the form of color normalization [143] can eliminate this undesirable variation in the object's appearance (Figure 2.2 right).

Figure 2.2: Original Caltech-101 faces (left) and after part-based color normalization [143] (right).

The main part of the pipeline, related to object appearance learning, is called "Representation and description". A large variety of approaches exist for object representation and description. For example, in the majority of object detection methods the objects are described with visual features that are defined explicitly (SIFT [115], HOG [39]) or implicitly (deep features [97]) (see Section 2.4 for more details); however, in template matching feature extraction is not needed. Depending on the type of an object some features are more suitable than the others. For example, in modelling a cup the main focus should be on its shape, better described with edge features, but modelling an animal like a leopard is better done using textural features. Object representation methods can be divided into two groups based on the use of spatial information in the object model formulation, i.e. model-free (BOW [159], CNN [97]) and model-based (DPM [55]) methods. Another way to categorize object representation methods is based on the nature of object model learning: generative, modelling the joint distribution of input vectors and class labels, or discriminative, defining only a decision boundary between true-class and not-true-class distributions. Object representation is discussed in Section 2.5.

Recognition in object detection is based on scores associated with a certain class label and object location. For each unseen (test) image the detection system should produce a bounding box, defining the object's position in the image, with a corresponding class label and detection score. In some cases, location and/or score can be changed during post-processing based on prior information, e.g. updating of the location of the bounding box corners relative to the locations of the object parts or re-scoring based on co-occurrence of classes in the training set [55].

The success of any object detection method is measured with performance evaluation metrics. Object detection competitions, such as Pascal VOC [46] and ILSVRC [149], have established common evaluation procedure based on precision-recall curves. During evaluation double detections as well as detections with wrong localization are penalized. An object is considered to be found correctly if its overlap ratio A is greater than 0.5, i.e. $A = BB_{gt} \bigcap BB_{pred} / BB_{gt} \bigcup BB_{pred} \geq 0.5$, where BB_{gt} is a groundtruth bounding box and BB_{pred} is a predicted box. Detection performance of a method for a class is evaluated based on averaged precision calculated over 11 uniformly distributed levels of recall.

2.2 Challenges of Object Detection

The ultimate goal of object detection is to reach human capability of recognition across thousands of object categories. Fundamentally, image acquisition is a lossy process as a 3D scene is projected onto a 2D image plane. From this incomplete data, a detection system should be able to detect an object in the image. Therefore, detection systems are required to be robust to:

- Large intra-class variation, for example, the presence of undefined subclasses (Figure 2.3).

  The definition of the class is vague, as it is often defined by the object's function rather than its appearance. For example, different types of airplanes share the main function, flying, and construction concept, i.e., wings, engine, fuselage. However each of the construction elements might have a different appearance and relative position, producing intra-class variation of appearance.

- Viewpoint changes, which can drastically affect an object's appearance (Figure 2.4).

  Large viewpoint changes might reveal object structures that have not been seen previously due to self-occlusion. Moreover, different views of the same object may not share appearance similarity with each other, e.g. front and side views of a car.

- Deformation of non-rigid objects from image to image (Figure 2.5).

  The majority of objects in the human environment are strictly non-rigid, i.e. the doors of a car can be opened, the hands of a clock can point to different time, humans and animals can take different poses, thus there is a vast number of possible appearance combinations.

- Occlusion, self-occlusion or truncation, depending on the object's pose and the viewpoint (Figure 2.6).

Figure 2.3: Different subclasses from the category *airplanes* of Caltech-101 [51] dataset. Each subclass can be divided into further sub-classes, e.g. engine types and shape of passenger airplanes (top row), variation in wings and placement of propellers in retro airplanes (middle row), finally, military planes have very specific appearance depending on their purpose (bottom row).

Figure 2.4: *Airplanes* from ImageNet [148] dataset sown from different viewpoints. Even though images represent similar types of airplane (a rigid object), the object's appearance varies considerably from image to image due to 3D changes in the viewpoint.

The main causes of occlusion, self-occlusion or truncation are changes in the object's pose or configuration, the viewpoint or from zooming, which often occurs in uncontrolled conditions of natural images.

- Illumination variation, which can bring undesired variability into the object appearance representation (Figure 2.7).

Figure 2.5: *Snails* and *people* from the ImageNet [148] database representing various object deformations.

Figure 2.6: Examples of truncation, occlusion and self-occlusion (only one eye of an owl is visible) shown from left to right on the ImageNet [148] *owls*.

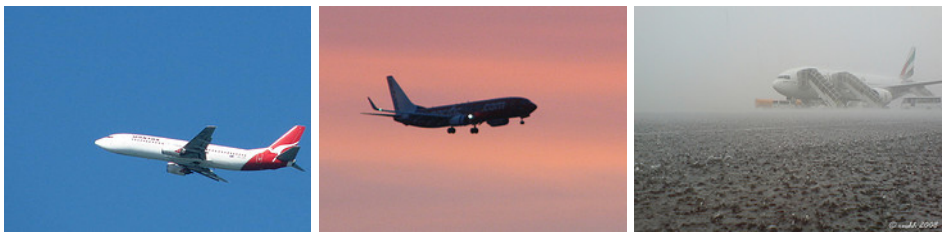


Figure 2.7: Effect of lighting conditions. In the first image the object is a bright spot on a dark background, whereas on the second picture a dark object is seen on a light background. In the last image the difference in contrast between the background and the object is very small.

Figure 2.8: *Cars* category from the ImageNet [148] dataset. The groundtruth, tight bounding boxes, is shown as red rectangles. It can be seen that both with and without the presence of a big object in the foreground small objects are difficult to notice.

- Changes in scale, which play an important role in object detection. For example, it is easier to detect a dominant (occupying a big portion of the image) object rather than a non-dominant tiny object (Figure 2.8).

Another group of challenges are caused by the data used in experiments. Firstly, requirements for the amount of training data and appropriate annotations should be fulfilled for a chosen method; however, having large amounts of human annotated images or negative examples might be infeasible. There is also a problem of subjective and/or incomplete annotations (Figure 2.9); what one user calls a car, another user would define with the label "minivan" or "Mercedes". Finally, the assumption used by all machine learning methods that training data represent fully all possible object appearance variations occurring in the test data is rarely checked and not necessarily true.

Figure 2.9: Examples of inconsistent groundtruth (bounding boxes and class labels). Some of the players (blue boxes) and cars (red boxes) in the leftmost image are annotated, while some other are not. The shoulder bag shown in the middle image has a label "backpack". The groom in the picture on the right is marked as a person but the bride is not. Images are taken from the ImageNet [148] dataset.

Another assumption used by model-based methods, namely that non-rigid objects can be represented with a set of rigid parts grouped together by non-rigid connections, is not always applicable. For example, cats are very flexible animals, thus attempts to describe them with a standard constellation model would fail. Most successful methods for detecting highly deformable objects use an object spatial model to locate a rigid part of an object, such as cat's face, combined with model-free methods like segmentation [131] or Bag-of-Words [132] for final object localization.

2.3 Object Detection Datasets

Data play an important role in object detection and classification tasks. Different detection and classification approaches have developed in conjunction with changes in the available datasets. Methods for detection of a single object based on a template matching [146], have been extended to single object detection in 3D [58, 114, 116] and then to object category detection [56, 15]. Datasets have gradually been extended in terms of complexity in object appearance: pose variation has become more complex (from 2D to 3D changes), multiple instances appear in images and occluded and truncated objects occur more often. Rapid development of internet resources, e.g. crowd sourcing, have made it possible to collect and annotate millions of images (LabelMe [151], ImageNet [148], Microsoft COCO [111]). The increase in the amount of images, data diversity and vast additional information (annotations) have stimulated development of completely new approaches to image classification and visual class detection that have not been possible before (e.g. Neural Networks [97]). Examples of images from different databases are presented in the Figure 2.10.

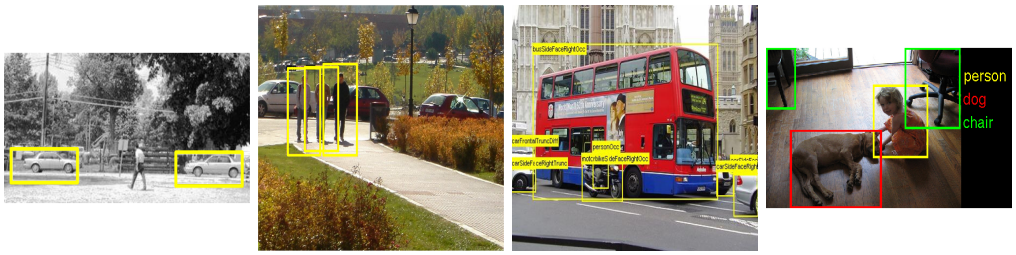


Figure 2.10: Images from UIUC car [2], INRIA person [40], Pascal VOC [48] and ImageNet [148] databases with example detections by different state-of-the-art methods.

The first generation of datasets was often gathered by members of a single group for a specific task, therefore many early datasets have only a small number of categories, e.g. MIT CBCL: faces [7], cars [130], pedestrians [128] or INRIA person [40], Caltech-4 [56] and UIUC car dataset [2]. Images in these datasets were often of poor quality, pre-scaled and centred, and sometimes histogram normalization was also performed. The objects in the datasets appeared with small variations in their appearance. Detection tasks for such datasets were almost perfectly solved already in the early stages of object detection algorithm development ([122, 106, 56, 40]). At that time, generative part-based models [34, 56] were competing with Bag-of-Word detectors (BoW) [159]. The generative models described the appearance of local parts and tolerated their spatial distortion, whereas the visual Bag-of-Words approaches omitted the spatial structure of object parts and described the classes via their local part histograms. With the help of strong discriminative learning methods, the BoW approach obtained the greater accuracy [18] on second generation datasets such as Caltech-101 [51].

The Clatech-101 dataset consists of a diverse set of image categories (100 objects and a background category). Each category contains from 40 to 800 images, though most of the categories are represented with approximately 50 images. Each image has only one object, and the objects are cropped and placed in the middle of the image. They

are also rotated to appear in the same pose, i.e. 3D pose variation is almost completely excluded. Nevertheless, there is great variability between images within each category, intra-class variability, e.g. part of the images are natural while others are drawings, or in some categories (e.g. *chairs*) images are grouped based on their functionality rather than appearance (see Figure 2.11). The big variety of categories in one dataset has stimulated the popularity of classification task development. Despite difficulties in modelling classes with high intra-class variability from a small number of training examples (30 training images), already in 2005 48% classification accuracy was achieved [14], which in 2006 improved to 66% [195] and in 2009 a 84.8% accuracy was demonstrated by Yang et al. [190]. However, deep neural networks, which show excellent results on big data problems, do not have record-breaking performance with small datasets like Caltech 101, achieving 87% classification accuracy [200] in 2014.

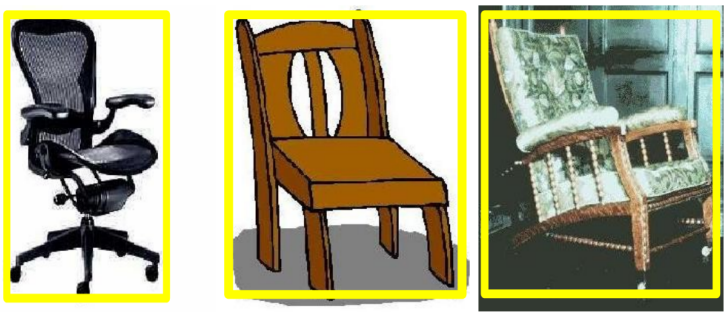


Figure 2.11: Examples of *chair* category in Caltech 101 with annotated groundtruth (bounding boxes).

In [134] the authors point out some disadvantages of Caltech-101 and earlier datasets, stating that the images are not challenging enough due to the similar viewpoint and orientation of objects within one category, the position of the objects in the images (which tend to be centred), the presence of only one instance per image, and little or no occlusion or background clutter. Some of these issues were resolved in the Caltech-101 extension Caltech-256 [73], containing many of the Caltech-101 old categories. Along with the increase in number of categories, the average number of images per category was also significantly increased in Caltech-256. Objects in the images became more challenging, as more variation in viewpoint was introduced, e.g. mirroring or 3D pose changes. Additionally, the quality of the images improved due to higher resolution.

The Pascal VOC challenge [48], presenting a third generation dataset, was initiated in 2005 to boost the development of sophisticated methods to solve different computer vision tasks. Pascal VOC included classification, detection, segmentation and action classification challenges, providing researchers from all over the world with a standard tool for evaluation of their success and fair comparison to others. The challenge ended in 2012. New images were added to the dataset each year, and between 2004 and 2012 the total number of images increased by almost five times, finally containing 20 object categories in more than 11 000 images [46]. Images in the Pascal VOC challenge represent real-life scenes with multiple instances of different categories in each image. Here, objects are shown with a lot of variation in scale, rotation and viewpoint. A big portion of objects are truncated or self-occluded. Most of the categories have very big intra-class variations and

can be divided into sub-categories either based on the viewpoint or appearance variation. Results for the detection challenge have progressed over the years at a rather steady pace thanks to the discriminative part-based approach by Felzenszwalb [55] and methods based on it, which until 2012 were constantly within the top performers. The Felzenszwalb method's accuracy on the Pascal VOC 2007 dataset was 29.1% mean average precision in 2010, while in 2014 RCNN (trained on the ImageNet dataset[148]) showed a detection result of 58.5% [69].

Finally, the fourth generation represented by large scale datasets (like ImageNet [148], COCO [111] or LabelMe [151]) have emerged. Millions of images and thousands of categories are now available. The ImageNet dataset is organized as a tree, so it can also be used for fine-grained classification. ImageNet challenges present 200 categories with 456 567 images for detection and 1000 object categories with 1 431 167 images for classification [150]. The structure of the images is simpler than in Pascal VOC (fewer objects in the image with a smaller number of truncated or occluded objects), but the amount of data has opened the door to a new, very powerful tool for object classification: deep neural networks (DNN). Current results for ImageNet are 6.7% error for classification and 43.9% mean average precision for detection (with an image classification dataset as extra training data) [163]. The best detection performance based only on provided data was 37.2% [109]. It is worth noting that a correct classification label is considered among the 5 top hypotheses what explains the big gap between classification and detection results. Generative methods have not been successful with ImageNet, even the DPM model is clearly below the state-of-the-art [163, 69, 197], but other discriminative models still dominate the field, in particular, deep neural networks [97, 69], which have been shown to implicitly learn local part detector layers [70].

2.4 Features for Object Detection

Image features have been one of the most popular tools for image representation in object class detection and classification tasks. Image features can represent the content of either the whole image, global features, or small parts of the image, i.e. local features. As global features aim to represent an image as a whole, it means that only a single feature vector is produced per image and thus a content of two images can be compared by comparing their feature vectors. On the other hand, to represent an image with local features usually a set of multiple local features, extracted from different parts of an image, is used. For local features, feature extraction can often be divided into two parts: feature detection and description. The main task of a detector is to find a set of stable (invariant) distinctive regions, while the descriptor encodes information about determined regions mathematically to enable efficient matching. Compared to global features, local features are more robust to occlusions and spatial variations: global features describe the image as a whole, thus traditionally they do not contain information of the spatial structure of the image and do not provide sufficient information for object localization. Local features are more stable and their relative locations can encode the spatial structure of objects, which is used in the part-based approaches to object class detection.

2.4.1 Global Features

In early stages of computer vision development, global features were widely used to solve scene classification and single object detection problems. Popular global features were color histograms and moments, edge orientations, frequency distributions and their combinations [171, 165, 72, 162, 124].

Color histograms were mostly used in 3D object recognition, where part of the object's views was used for training and another part for testing. This approach is mostly applicable if objects are presented on the uniform background and the lighting conditions are controlled [162]. To achieve illumination invariance, different color constancy methods [125, 63] are used as a preprocessing step. In scene classification, color information is useful to differentiate between landscape images (sunset, mountains, forest). Scenes of nature tend to have uniform and stable (similar from image to image) color regions like blue sky, green grass and trees, orange sunset etc. For man-made objects color as a cue is unstable as the objects can be made in an arbitrary color, e.g. a house can be yellow, blue or red, a car silver, black or green [171].

Statistics of edge orientations in the image, extracted from texture and frequency features, are useful in classifying indoor vs. outdoor and city vs. rural classes of images [165]. Man-made objects, like furniture and buildings, have distinct domination of vertical and horizontal edges clearly separating them from the nature landscapes with randomly distributed edge directions. In city scenes, horizontal edges are less stable than vertical ones because of variation introduced by perspective. In terms of frequency distribution, most rural images are dominated by high and low frequencies corresponding to high textural areas, like grass and trees, and low textural areas, like water or sky. In city images middle range frequencies dominate the images.

The use of global features coupled with the local ones recently found a new application in generation of category-independent region proposals, e.g. objectness [3, 4]. The objectness paradigm is based on the following properties of an object: an object in the image is defined by a closed boundary; an object has a different appearance from its surroundings; the object stands out as a salient region in the image. The candidate regions (windows) are proposed based on a combination of global and local features: global multi-scale saliency, color contrast (measure of dissimilarity of the proposed window and its surroundings), density of edges near to the window borders, and superpixel straddling (images are segmented into regions with uniform texture or color, superpixels; the window containing connected segments inside of a tight window scores highest).

2.4.2 Local Features

Edge features

Scale Invariant Feature Transform (SIFT) was proposed by Lowe in 1999 [114], and a more stable version was subsequently presented in 2004 [115]. Based on SIFT features, a widely used Bag-of-Words approach [37, 103, 132] was developed. SIFT features are interest point based and can be used in unsupervised learning [169]. Recent studies [80] have reported that SIFT descriptors demonstrate best performance even when compared to modern fast descriptors.

SIFT features are defined by an interest point detector and a local image descriptor. Interest points, found as local peaks of difference-of-Gaussian (DoG) functions, correspond to strong edges, corners and intersections. The scale invariance is achieved by the search of interest points (local maxima) across the scales in a scale-space DoG pyramid. Orientation invariance is based on the dominant orientation assigned to every interest point. Dominant orientations are calculated from the histogram of gradient orientations in the interest point neighbourhood. The highest peak in the orientation histogram defines the orientation of the interest point. However, other local peaks within 80% of the highest peak produce secondary interest points with corresponding orientations.

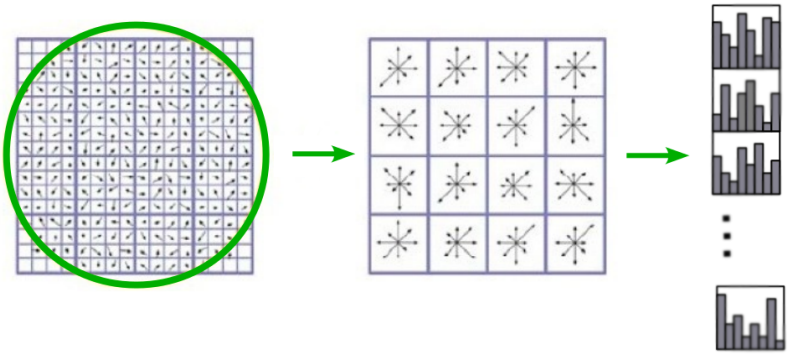


Figure 2.12: Illustration of SIFT descriptor formulation.

The SIFT descriptor is calculated for the scale level defined by the detector's interest point scale and gradient orientations are rotated to align their dominant orientation, thus enabling SIFT features to achieve scale and orientation invariance. The SIFT descriptor is composed of the Gaussian weighted gradient amplitudes calculated in eight directions. Figure 2.12 illustrates the descriptor construction. Errors in the image represent extracted gradient orientations and magnitudes. The green circle shows the Gaussian used for weighting gradient magnitudes, which makes the descriptor more robust to small changes in interest point position. The area around the interest point, divided into 16 sub-regions, produces sixteen 8 direction bin histograms from the weighted gradients that are subsequently concatenated into 128 dimensional descriptor vector. Nowadays popular and most efficient implementations of SIFT features are: VLFeat [175], OpenCV [23] and UBC (D. Lowe's) implementation [113].

Another very popular contour feature often used for object detection is HOG (Histogram of Oriented Gradients), which was proposed by Dalal and Triggs for pedestrian detection in [39]. HOGs use the distribution of local intensity gradients to describe both object appearance and shape (Figure 2.13). HOGs require more supervision than interest point driven SIFT detectors: for successful learning it is extracted from a bounding box region around the object [39, 55]. Another difference between HOG and SIFT features is that SIFT chooses the dominant orientation of a feature, while HOGs keep information about all gradient orientations.

Building a Histogram of Oriented Gradients starts with calculation of gradients for each

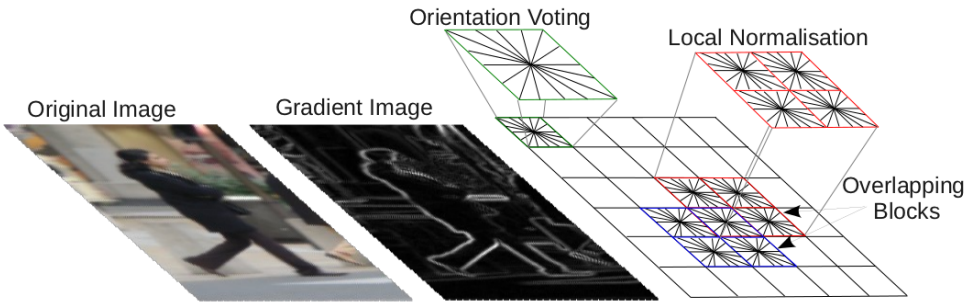


Figure 2.13: Illustration of HOG descriptor formulation.

pixel in the image. In color images, only the value of the channel with the highest norm of the gradient is chosen. This use of locally dominant color provides color invariance. Image window is then divided into small rectangular cells. A histogram of gradient orientations with 9 orientations is constructed for each cell. The gradient magnitudes of the pixels in the cell are used as votes in the orientation histogram (Orientation Voting in Figure 2.13). The final stage employs contrast normalization for the overlapping 2×2 blocks of cells. Each block is normalized separately. Moreover, as normalization is performed for overlapping blocks, each cell contributes to several blocks, and is normalized every time accordingly. Normalization introduces better invariance to illumination, shadowing and edge contrast. The normalized block descriptors are referred to as the Histogram of Oriented Gradients (HOG). A feature vector is constructed of HOG descriptors taken from all blocks of a dense overlapping grid of blocks covering the detection window. The most used implementations of HOG features are: VLFeat [175], OpenCV [23] and Pedro Felzeszwalb's implementation [55]. HOG features in combination with a deformable part-based model (DPM) provide state-of-the-art results in many applications such as tracking [196, 167] and object detection and classification [181, 180, 107, 202].

Texture features

In early works, wavelets, Gabor features and image patches were widely used texture features [100, 185, 128, 56]. Wavelets and Gabor features are multiresolution function representations that allow a hierarchical decomposition of a signal [118]. Wavelet and Gabor features allow a potentially lossless image representation and reconstruction (in contrast to e.g. HOGs [178] or SIFT [184] features), because wavelets or Gabor filters applied at different scales encode information about an image from the coarse approximation to any level of fine details [104]. As Gabor filters are the features of choice in this work, their construction and properties are described in detail in Section 3.1.

An industrially used face detector implemented in modern photo cameras for focusing on faces has been developed by Viola and Jones [177]. It is based on the simplified Haar wavelets. These Haar-like features, represented by two-, tree- and four-rectangle features (Figure 2.14 left), are extremely efficient in computation. The dark part in the image corresponds to a weight -1 and the white part to a weight +1, therefore,

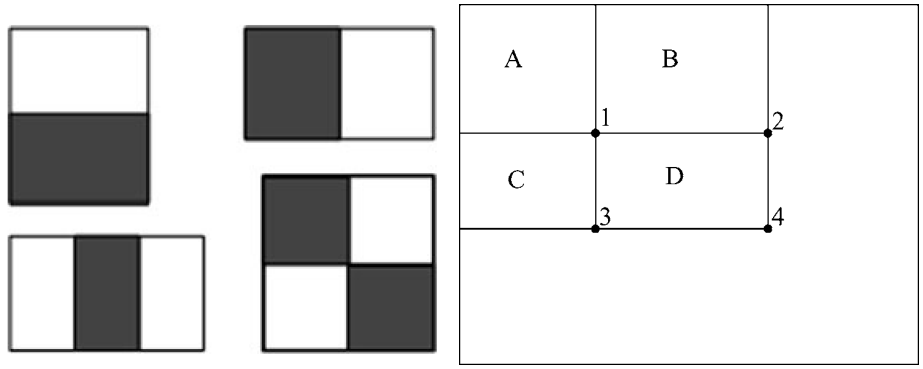


Figure 2.14: Haar-like simple features and an integral image.

simple Haar-like features are calculated as the difference between the sum of pixels within dark and white regions. These features capture the relationship between the average intensities of neighbouring regions and encode them along different orientations. Efficient feature extraction is achieved through the use of the integral image I_i (Figure 2.14 right), which allows calculation of the sum of elements in any arbitrary rectangle with only four references to I_i. Efficiently calculated simple features and classifiers, arranged as a cascade, made the Viola-Jones face detector one of the fastest detectors of the time, leading to its extensive use in industry.

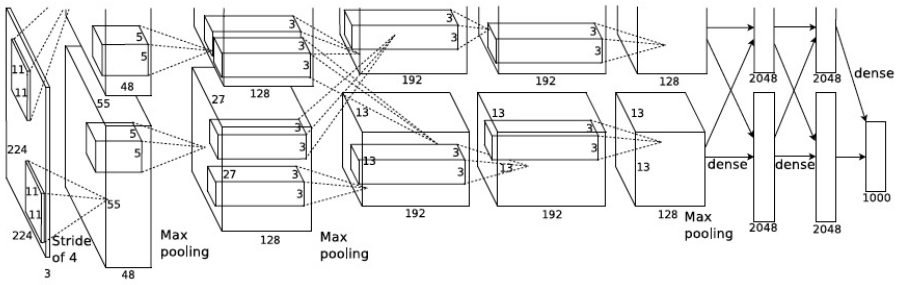


Figure 2.15: Illustration of the architecture of a Convolutional Neural Network from [97].

Recently a new generation of features has been introduced for object detection and classification. These are non-engineered features produced by deep Convolutional Neural Networks during the learning procedure [97], referred to as "deep features". Even though deep features are learned, the neural network's structure is manually engineered, inspired by the biological visual cortex, and contain a lot of parameters learned from data. The architecture of a Convolutional Neural Network is shown in Figure 2.15. Deep features are produced by alternating convolution and pooling procedures, where convolution can be thought of as actual feature extraction (filtering) and pooling as an invariance step. A max-pooling layers reduce feature dimensionality and computations for the following

layers, simultaneously enabling position invariance over larger local regions and improving generalization. Figure 2.16 shows the kernels of the first convolutional layer learned by the network. It can be seen that the network has learned a variety of frequency- and orientation-selective kernels, as well as various color blobs. Thus color information plays an important role in the excellent performance of neural networks in computer vision tasks [29].

As deep features are learned from the training data, the choice of the dataset affects feature formulation, i.e. features are data specific. In [200] Zhou et al. show the effect of training data on the results of image classification. In particular, deep features learned on object oriented data (ImageNet [148]) perform better than features trained on scene oriented data (Places database [200]) for the object oriented datasets and vice versa. Therefore in [200] authors propose to combine both training datasets and obtain results either better or similar to the best performing method on all datasets. Deep features extracted after the last pooling layer learned on the object oriented dataset look like object-blobs. Features learned on the Places dataset look like landscapes with more spatial structures, their visualization can be found in [200]. Interestingly, parameters of a DNN learned on a large dataset, like ImageNet, produce good results when applied to other smaller data, either as is or after fine-tuning of the final classification layer on the target data [69].

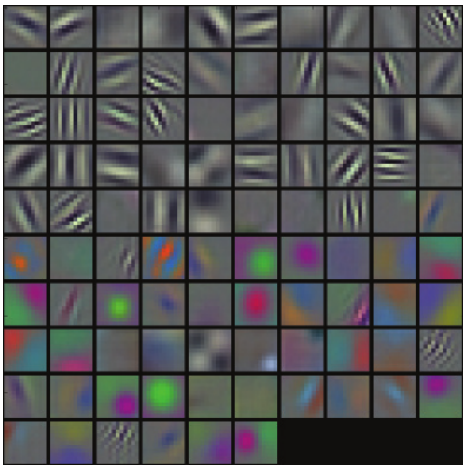


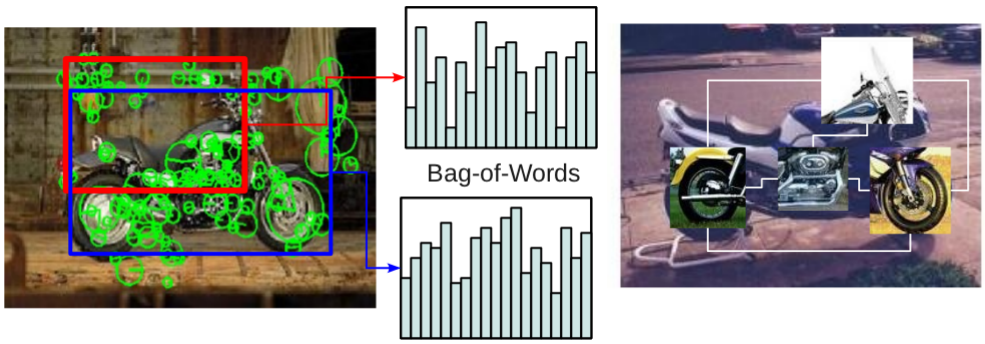
Figure 2.16: Deep features of the first convolutional layer.

DNNs are very powerful learning tools that can achieve excellent human-competitive results on visual and speech recognition tasks, however how and what they learn from the data is still unclear and results in some counter-intuitive properties. For example, invisible to human non-random perturbations of an image can change the category it is assigned to by a network [164] or specially generated artificial images meaningless to a human [123] can confuse the neural network and obtain a class label with high probability. The most common open-source CNN implementations are: Caffe [90] and OverFeat [154].

2.5 Object Representation

2.5.1 Model-free vs. Model-based

The problem of object detection, which is localization and classification of objects appearing in still images, is a hot topic in computer vision. Due to its large variations in scale, pose, appearance and lighting conditions, the problem has attracted a wide attention and a number of algorithms have been proposed. Existing object detection algorithms can be divided into two categories: model-free methods [17, 18, 37, 69, 159, 197] and model-based methods [1, 34, 54, 55, 56, 140, 141]. Specifically, the difference between model-free methods and model-based methods lies in the usage of the explicit object models with spatial constraints between object parts (Figure 2.17).



(a) model-free **(b)** model-based

Figure 2.17: Illustration of model-free and model-based object detection concepts. Sub-figure (a) demonstrates detection principle of Bag-of-Words model-free method which does not use spatial information in object model. Sub-figure (b) shows a part-based model of a motorbike, using which object detector is aware of both object part appearance and their relative spatial locations.

In the category of model-free methods, discrimination of feature representation plays a dominating role in mitigating large variations of pose, scale and appearance. The most well known model-free methods are Bag-of-Words [103] and more recent deep feature approaches [69, 197]. Deep learning architectures [97, 157, 69] learn a constellation model implicitly along the deep layers of processing. First visual bag-of-words (BoW) models [159, 37] omitted spatial constellation of parts and used shared codebook codes to describe the parts. However, the BoW model can be extended to include loose spatial information, for example, by dividing the image to spatial bins [103] or refining the codebook codes by their spatial co-occurrence and semantic information [105].

On the other hand, by introducing object models, both the appearance of local object parts and the geometric correlation between object parts (e.g. star model [57] or Implicit Shape Model [106]) can be simultaneously learned in a unique framework. Thus, part-based object model detection is based on two factors: detection of object parts and

verification of their spatial constellation. The first part-based approach to object detection was proposed by Fischler and Elschlager in 1973 [62]. In earlier works of generative part-based constellation algorithms [56], the location of the parts was limited and only a sparse set of candidates, selected by a saliency detector, was considered. In [34], the proposed pictorial structure model can tolerate changes of pose and geometric deformation of the object, but label annotation is required for each object part. The first attempts to learn full models of parts and their constellation were generative [183, 51], but due to the success of discriminative learning the generative approach has received less attention recently. Some object detectors (both model-free and model-based) are presented in Table 2.1. Methods are arranged chronologically in four groups (two for model-free methods, i.e. Bag-of-Words, and two for model-based methods, i.e part-based methods).

2.5.2 Generative vs. Discriminative

Object detection and classification methods can be divided into two major categories based on their learning principle: generative [56, 53, 8, 91] and discriminative [177, 147, 39, 55] approaches. The difference between these two approaches is that generative models capture the full distribution of an object class while discriminative models learn just a decision boundary between object class instances and the background or other classes.

Let $\boldsymbol{x}$ correspond to raw image pixels or some features extracted from the image and c is an object class that might be present in the image. Given training data consisting of N images with $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots \boldsymbol{x_N}\}$ and corresponding class labels $C = \{c_1, c_2, \dots c_N\}$, when images and their labels are drawn from the same distribution, the system should be able to predict a label $\hat{c}$ for a new input vector $\boldsymbol{x'}$. The best characteristic guaranteeing minimization of the expected loss, e.g. number of misclassifications, is a posterior probability $p(C|\boldsymbol{X})$. In discriminative approaches this posterior probability is learned directly from the data. Generative approaches, on the other hand, model the joint distribution over all variables $p(C, \boldsymbol{X})$ and posterior probabilities are calculated using Bayesian formula. Generative models are appealing for their completeness and often have higher generalization performance than discriminative models; however, they are redundant (as the system needs just posterior probabilities).

Generative methods can handle missing or partially labelled data, i.e. use both labelled and unlabelled data. New classes can be added incrementally independently from previous classes. As generative learning procedure learns a full data distribution, one can sample this learned model to 1) verify if it indeed represents provided training data or 2) artificially extend training set by generating new instances. In contrast to discriminative models, generative models can handle compositionality, i.e. they do not need to see all possible combinations of features during training (e.g. hat+glasses, no hat+glasses, no hat+no glasses, hat+ no glasses). Discriminative methods are generally faster and have better predictive performance as they are trained to predict class labels whereas generative methods learn a joint distribution of input data and output labels. Based on the differences in training and calculating generative and discriminative models one of the most important distinctions arises: to train the object model generative models do not need background data [140, 8], but discriminative models need both positive and negative examples to learn decision boundaries. The most common discriminative learning tools are SVMs [172, 173], neural networks [97] and decision trees [136, 24].

Complementary properties of discriminative and generative methods have inspired a number of efforts to combine the approaches and utilize the best of both paradigms. Hybrid approaches are used in a number of computer vision applications [20, 110, 108]. The version of generative-discriminative hybrid object detector in this work is presented in Section 5.1.

2.5.3 Examples

The deformable part-based model (DPM) [55] is a discriminative model-based method for visual class detection and classification. DPM is one of the most successful examples of using HOG features for visual class detection. The DPM has only a few tunable parameters, owing to the fact that selection of the parts, learning their descriptors and learning of the discriminative function for detection are all embedded in the latent support vector machine framework (see Figure 2.18). Intuitively, DPM alternates optimization of the learning weights and the relative locations of deformable part filters in order to achieve high response in the foreground and low response in the background. With the learned DPM model, the root filter and part filters are applied to scan the whole feature pyramid map to find regions with high response, which can finally determine locations of the object. In the final stage, the location of the bounding box is refined and re-scored based on the training statistics of bounding box corner positions relative to a root filter. A deformable part-based model [55] is used in the experiments in this work as the discriminative part of the hybrid method (Section 5.1).

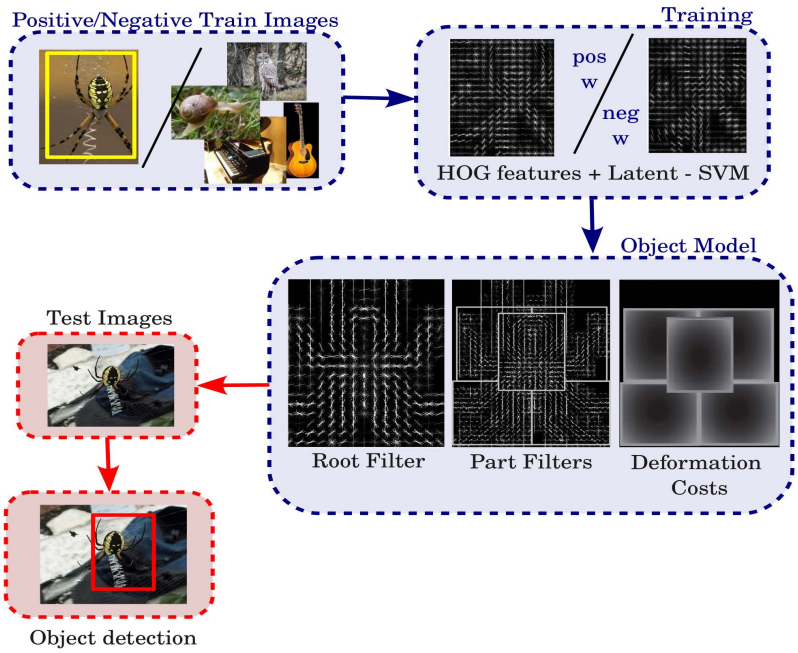


Figure 2.18: The deformable part-based model (DPM) [55] for learning and detecting visual classes.

Linear Discriminant Analysis of the DPM model [78] has resulted in WHO features (Whitened Histogram of Orientations), allowing expensive SVM training to be avoided. In [78] background class is estimated just once and reused with all object classes.

Bag-of-(visual-)Words (BoW) is a discriminative model-free method often applied to object detection and classification tasks [37, 44, 158]. The framework of BoW methods is very simple. First, local image features such as SIFTs are extracted. These features are then clustered to form an N-entry codebook characterized by N visual words. Images are represented by histograms showing how many features from each cluster occur in the image (cluster histograms). Histograms of training images are used to train an SVM classifier. During testing, cluster histograms are constructed for all test images (overlapping candidate detection windows in different scales and positions) and then scored by SVM. Figure 2.17 left shows an example of two candidate detection windows, each producing a histogram allowing to classify it as containing or not containing the object.

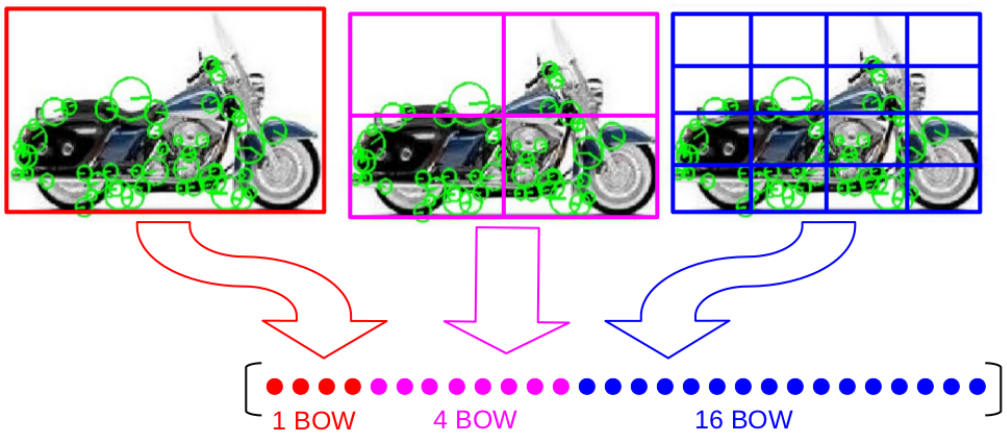


Figure 2.19: An example of a spatial pyramid with Bag-of-Words.

One of the most popular BOW modifications is Bag-of-Words with a spatial pyramid [103]. Original BoWs are incapable to capture shape or segment an object from its background; however, adding a spatial object model on top of a BoW representation is not straightforward. In [103], to include spatial information images were repeatedly subdivided (Figure 2.19) and histograms of local features were constructed for all obtained image regions with increasingly fine resolutions.

Bourdev et al. [21] propose a discriminative model-based method for body parts (poselets) detection. The method is based on data with annotated keypoints, in particular, joints of the human body. Poselets are very discriminative image patches that form a dense cluster in the appearance space. To find poselets, a lot of seed patches are first randomly generated from object regions of training images. When a seed window is chosen, patches with similar spatial configuration of keypoints are extracted from training images and aligned with the seed based on their keypoints. The most dissimilar candidate patches (with big residual error) are excluded from the set of positive examples. Negative examples are sampled randomly from images that do not contain the object.

Then HOG features are extracted from all patches (positive and negative) and used to train an SVM classifier. Finally a small set of poselets is selected based on the frequency of their occurrence in the training images. In [182] a model of a human pose was hierarchically constructed out of poselets and further used for person detection and tracking when a system knows the location of each separate object part.

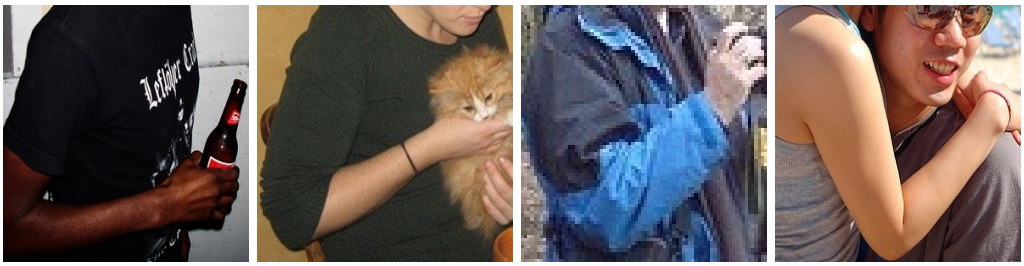


Figure 2.20: Examples of poselet image patches corresponding to a bent right arm from which HOG features are extracted.

A work of Ying Nain Wu et al. [186] on an object active basis (sketch) model is an example of a recent generative model-based method. The model proposed in [186] describes an object with a small number of representative strokes. Each oriented stroke of an object model is effectively described by a Gabor filter. Filters are allowed to shift their locations and orientations for a best description of the nearest edge. For the final object representation, those filters are chosen whose shifted versions sketch the most edge segments in the training images. Thus, the learning process resembles a simultaneous edge detection in multiple images.

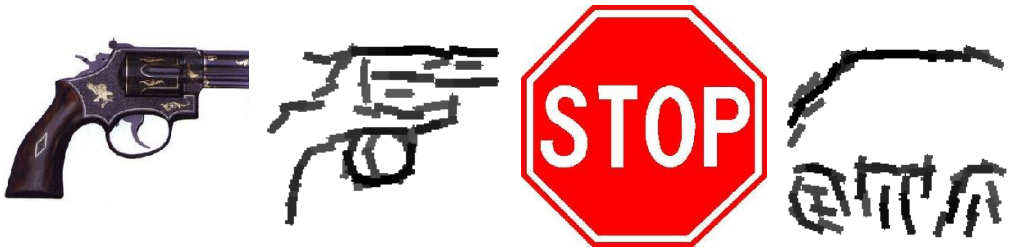


Figure 2.21: Examples of active basis models for *revolver* and *stopsign* Caltech-101 categories (original images on the left, corresponding models are on the right).

This work has been extended to form a hierarchical compositional object model [38]. The model in [38] is composed of object parts that are allowed to shift their location and orientation, which in turn are composed of Gabor filters (strokes) that are also allowed to shift their location and orientation. Another recent generative hierarchical object model is proposed in [59]. In the work low level features are represented by oriented Gabor filters learned in an unsupervised category-independent way, while high level object parts are constructed by using specific categories.

2.6 Summary

This chapter presented the parallel synergy and evolution of datasets and methods for visual class detection. Major challenges of computer vision have been solved to different extents and many different methods have been used to approach them. The different methods have their advantages and disadvantages, e.g. model-free methods are very flexible in object representation and can model several views/poses at the same time. However, a lack of spatial information makes model-free methods less precise than model-based ones, which are able to provide object location information and filter out hypotheses with high appearance scores but not consistent with the spatial model. Generative methods produce a complete model, can handle unlabelled data and the absence of negative examples. However discriminative methods, despite acting like a black box and being unable to explain obtained results, often outperform generative ones.

Table 2.1: Part-based methods for object class detection (chronological order, Cl.: the classifier type, Discriminative (D)/Generative (G)).

Ref	*Feature*	*Const. model*	*Cl.*	*Test data*
Bag of Words (omitted, see Huang et al. [83] for survey on state-of-the-art):				
Sivic 2003, [159]	Codebook histogram	-	D	Own
Lazebnik 2006, [103]	Spatial codebook histogram	-	D	Scene-15, Caltech-101, Graz-02
...	...	...	...	...
Cao 2010, [26]	Spatial codebook histogram	-	D	Oxford buildings
Bag of Words with spatial model:				
Weber 2000, [183]	Codebook	P parts in canonical space	G	Faces and cars
Agarval 2004, [1]	Codebook	Pair-wise relation	D	Own 170 car images
Leibe 2008, [105]	Codebook	Hough spatial voting	D	Own car images
Carbonetto 2008, [27]	Codebook	Overlap of spatial segments	G	Caltech-4, Corel, Graz-02
Allan 2009, [5]	Codebook (category specific)	Gen. model, search over pose parameters	G	VOC2005 (4 categories)
Ommer 2010, [127]	Codebook (category specific composition)	Compositions with respect to the object centre	G	Caltech-101, VOC2006
Early part-based constellation model (with interest point detectors):				
Fergus 2003, [56]	Patch from IPs	P parts in canonical space	G	Caltech-4
Fei-Fei 2006, [50]	Patch from IPs	P parts in canonical space	G	Caltech-101
Crandall 2007, [35]	Various features	Pair-wise relation	G	VOC2006
Holub 2008, [82]	Patch from IPs	P parts in canonical space	G+D	Caltech-4 and Graz
Bar-Hillel 2008, [12]	Patch from IPs	Star model	G	Caltech-4 + own
Todorovic 2008, [168]	Segments	Segmentation trees and sub-tree matching	G	3 cl. from Caltech-101, UIUC cars, horse and cow images
Chen 2009 and Zhu 2009, [31, 201]	Various features and detectors	Feature triplet based stochastical grammar	G	26 cl. from Caltech-101
Part-based constellation model:				
Rao 1995, [137]	Gaussian derivative features	Spatial voting	G	A few simple objects
Burl 1998, [121]	Sliding window detector	P parts in canonical space	G	Own face images
Crandall 2005, [34]	Edge features	K-fan representation	G	Caltech-4
Felzenszwalb 2005, [53]	Steerable filters + diagonal Gaussian pdf	Pair-wise energy model	G	20 from Yale face database, articulated torso images
Eichner 2009, [45]	General detector + part detector (color features)	Spat. prob. model on a "detected frame"	G	Torso images in "Buffy", VOC2008
Heitz 2009, [79]	Boosted set of various features on object boundary	Boundary model from learned parts on boundaries	G	Own "googled" (giraffe, cheetah, airplane etc.)
Kumar 2009, [98]	Color and HOG	Tree structure between "putative poses" of parts	D	Videos of human movement: sign language and Buffy
Lin 2009, [112]	HOG	No spatial model - sums votes of part detectors	D	Human detection (INRIA and MIT data sets)
Bergtholdt 2010, [15]	Sliding window detector	Graphical model, A^*-search	D+G	Caltech-4 face, torso images
Wu 2010, [186]	Gabor edge detectors	Learned "Gabor edge map" by matching pursuit	G	Own cars, bicycles and a few animals
Felzenszwalb 2010, [55]	HOG	Root filter and deformable parts	D	Pascal VOC 2006-2008
Wang 2011, [182]	Multiscale HOG features (poselets)	Multiscale hierarchy of parts	D	UIUC people dataset
Zhang 2014, [198]	Deep features	Based on Gaussian mixture model	D	Caltech-UCSD birds
Learned object model:				
Krizhevsky 2012, [97]	Deep features (original CNN)	-	D	ILSVRC10, ILSVRC12
Girshick 2014, [69]	Region proposals + deep features (R-CNN)	-	D	PASCAL VOC 07,10-12
Sermanet 2014, [154]	Deep features at multiple scales (OverFeat)	-	D	ILSVRC12, ILSVRC13
Szegedy 2014, [163]	Image sampling + deep features (GoogLeNet)	-	D	ILSVRC14

Chapter III

Gabor Local Part Detector

The assumption that an object can be described with a set of parts linked with each other by spring-like connections has lead to development of a variety of part-based object detection methods starting from Fischler and Elschlager in 1973 [62] and evolving into modern state-of-the-art object detectors like that presented in Felzenszwalb et al. [55]. Along with the standard computer vision task of object detection, where the location of an object is often coarsely defined by a bounding box, supplementary detection of specific semantically meaningful object parts allows additional information to be obtained, such as the object's pose [160, 10, 193] or the identity of a face [199, 30].

This chapter presents a learning and detection pipeline (Figure 3.1) first introduced in [141], explicitly providing locations of class specific landmarks, i.e. object parts. These manually annotated landmarks are the distinguishable parts of the objects with semantic meaning (Figure 3.5 top row). Object parts are learned in a generative manner from only a few positive examples. In [87] Gabor features showed superior performance compared to steerable filters [156] or local binary pattern (LBP) [126] approaches paired with the Gaussian mixture model classifier and are the features of choice in this work. Therefore, the object appearance model is represented by a Gaussian mixture model of a selected subset of complex-valued multi-resolution Gabor bank responses. During testing, the proposed method transforms an input image to a part conditional likelihood map of landmarks. From these likelihood maps, a desired number of the best candidates can be fetched and used in further processing stages for object detection and localization. Figure 3.1 illustrates the method output.

3.1 Gabor Features

Gabor filters were originally introduced in 1946 by Dennis Gabor. They are used to represent signals as a combination of elementary functions [65]. In [42] Daugman demonstrated the similarity of 2D Gabor filters to simple cells of mammalian visual systems, showing the biological relation of Gabor features. Moreover, invariance to translation,

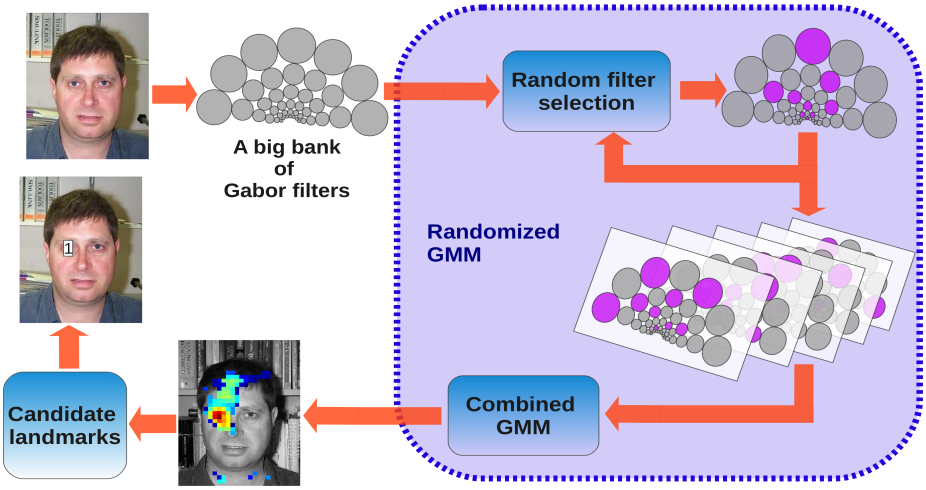


Figure 3.1: Generative part-detector workflow.

rotation, scale and illumination changes, which are desirable feature properties in computer vision, are all achievable with features constructed from the responses of Gabor filters. Due to these properties Gabor features have been successfully used in many computer vision applications, especially in biometrics (iris recognition [41], face recognition [155, 192], face expression recognition [74] and fingerprint matching [89]). Gabor features are considered as effective texture descriptors [22, 119, 77, 144], but encoding of local object parts was also among their first applications [100, 185].

3.1.1 Multi-resolution Gabor Features

In the spatial domain, the Gabor filter is a complex plane wave (a 2D Fourier basis function) multiplied by an origin-centred Gaussian (Equation 3.1), and in the frequency domain, it is a single real-valued Gaussian centred at f (Equation 3.2), as visualized in Figure 3.2. The multi-resolution forms and parametrization in (3.1) and (3.2) are used in this work. The multi-resolution Gabor filter is a restricted version of the general 2D Gabor function derived by Daugman [42] from Gabor's original 1D "elementary function" [65]. The restricted form enforces self-similarity, i.e. all filters are scaled and rotated versions of each other ("Gabor wavelets") regardless of the frequency f and orientation θ.

A core element of the 2D Gabor filter function is [93]:

$$\begin{aligned} \psi(x,y) &= \frac{f^2}{\pi\gamma\eta} e^{-(\frac{f^2}{\gamma^2}x'^2+\frac{f^2}{\eta^2}y'^2)} e^{j2\pi f x'} \\ x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \quad , \end{aligned} \tag{3.1}$$

where f is the central frequency of the filter, θ the rotation angle of the Gaussian major axis and the plane wave, γ the sharpness (bandwidth) along the major axis, and η the

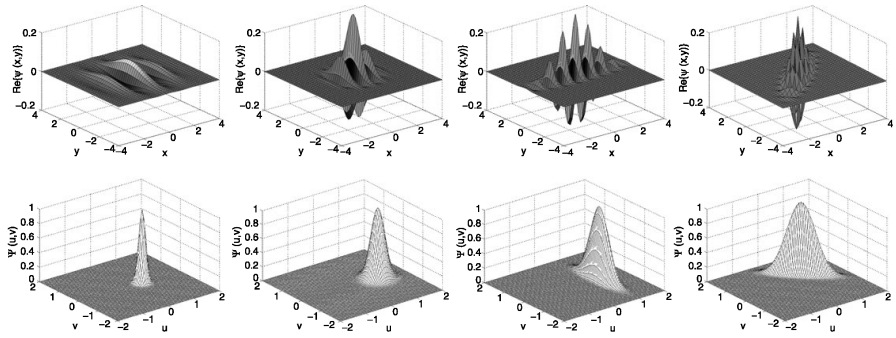


Figure 3.2: Example of a 2D Gabor filter in the spatial and frequency domain.

sharpness along the minor axis (perpendicular to the wave). In the given form, the aspect ratio of the Gaussian is η/γ.

The Gabor function in Equation 3.1 has the following analytical form in the frequency domain

$$\begin{aligned} \Psi(u,v) &= e^{-\frac{\pi^2}{f^2}(\gamma^2(u'-f)^2+\eta^2 v'^2)} \\ u' &= u\cos\theta + v\sin\theta \\ v' &= -u\sin\theta + v\cos\theta \ . \end{aligned} \tag{3.2}$$

Responses of the Gabor filters r for an image I are calculated as a convolution:

$$\begin{aligned} r(x,y;f,\theta) &= \psi(x,y,;f,\theta) * I(x,y) = \\ &= \int_{-\infty}^{\infty}\!\!\int \psi(x-x_\tau, y-y_\tau,;f,\theta) I(x_\tau,y_\tau) dx_\tau dy_\tau . \end{aligned} \tag{3.3}$$

The relationship between different filters provides the basis for distinguishing objects, therefore multi-resolution Gabor features are constructed from the responses of the filters in (3.1) or (3.2) on multiple frequencies and orientations. The detector assumes that local parts can be described by multi-resolution Gabor features computed at single locations, often in the middle of an object part. The assumption provides computational simplicity for processing and this single location features still provide a powerful representation of an object part as the Gabor filters are sensitive in the vicinity of their location. In this work, filter responses at the location (x_0, y_0) are arranged into a matrix form as [99]

$$\mathbf{G} = \begin{pmatrix} r(x_0,y_0;f_0,\theta_0) & r(x_0,y_0;f_0,\theta_1) & \cdots & r(x_0,y_0;f_0,\theta_{n-1}) \\ r(x_0,y_0;f_1,\theta_0) & r(x_0,y_0;f_1,\theta_1) & \cdots & r(x_0,y_0;f_1,\theta_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ r(x_0,y_0;f_{m-1},\theta_0) & r(x_0,y_0;f_{m-1},\theta_1) & \cdots & r(x_0,y_0;f_{m-1},\theta_{n-1}) \end{pmatrix} \tag{3.4}$$

where rows correspond to responses on the same frequency and columns correspond to responses at the same orientation. The first row is the highest frequency $f_0 = f_{max}$ and the first column $\theta_0 = 0°$. Despite computation in a single location, every filter "sees" in its vicinity defined by the filter's effective bandwidth (the Gaussian envelope controlled by γ and η).

3.1.2 Gabor Feature Properties

It is interesting that the actual physical structure in the striate cortex of the mammalian visual system has characteristics similar to a Gabor feature space $\boldsymbol{G}$ in (3.4). In the cortex the processing cells responsible for different orientations are layered on a columnar structure, which is repeated for different scales in the perpendicular direction [84].

As filters on the different frequencies with the same bandwidth are scaled versions of each other, to achieve a homogeneous spacing between the scales, frequencies f_m in 3.4 are drawn from the logarithmic distribution [87]:

$$f_m = k^{-m} f_{max}, \;\; m = \{0, \ldots, M-1\}, \tag{3.5}$$

where f_m is the mth frequency, $f_0 = f_{max}$ is the highest frequency, and $k > 1$ is the frequency scaling factor. The filter orientations are drawn from a uniform distribution:

$$\theta_n = \frac{n2\pi}{N}, \;\; n = \{0, \ldots, N-1\}, \tag{3.6}$$

where θ_n is the nth orientation and N is the total number of orientations.

An important property which makes multi-resolution Gabor features computationally attractive is the fact that simple row-wise and column-wise shifts of the response matrix correspond to scaling and rotating in the input space. Rotating an input image anti-clockwise by $\frac{\pi}{N}$ corresponds to the following column-wise shift of the feature matrix

$$\begin{pmatrix} r(x_0,y_0;f_0,\theta_{n-1})^* & r(x_0,y_0;f_0,\theta_0) & \Rightarrow & r(x_0,y_0;f_0,\theta_{n-2}) \\ r(x_0,y_0;f_1,\theta_{n-1})^* & r(x_0,y_0;f_1,\theta_0) & \Rightarrow & r(x_0,y_0;f_1,\theta_{n-2}) \\ \vdots & \vdots & \ddots & \vdots \\ r(x_0,y_0;f_{m-1},\theta_{n-1})^* & r(x_0,y_0;f_{m-1},\theta_0) & \Rightarrow & r(x_0,y_0;f_{m-1},\theta_{n-2}) \end{pmatrix} \tag{3.7}$$

where * denotes the complex conjugate. Downscaling an image by a factor $\frac{1}{k}$ corresponds to the following row-wise shift of the feature matrix

$$\begin{pmatrix} r(x_0,y_0;f_1,\theta_0) & r(x_0,y_0;f_1,\theta_1) & \cdots & r(x_0,y_0;f_1,\theta_{n-1}) \\ r(x_0,y_0;f_2,\theta_0) & r(x_0,y_0;f_2,\theta_1) & \cdots & r(x_0,y_0;f_2,\theta_{n-1}) \\ & & \ddots & \\ \Uparrow & \Uparrow & & \Uparrow \\ r(x_0,y_0;f_m,\theta_0) & r(x_0,y_0;f_m,\theta_1) & \cdots & r(x_0,y_0;f_m,\theta_{n-1}) \end{pmatrix} \tag{3.8}$$

It should be noted that responses on the new low frequencies f_m in (3.8) must be computed and stored in advance, while the highest frequency responses on f_0 are excluded from the feature matrix. The matrix shifts provide fast search of local parts over scaling and rotation.

Illumination invariance is achieved when feature matrix 3.4 is normalized as follow:

$$\mathbf{G}' = \frac{\mathbf{G}}{\sqrt{\sum_{i,j} |\mathbf{G}_{i,j}|^2}} \tag{3.9}$$

where $\boldsymbol{G}_{i,j}$ is an entry of the feature matrix on the i^{th} row and j^{th} column. Unfortunately, as illumination invariance is achieved, the number of false responses on the background often increases as areas with high and low energy are equalized.

As Gabor filters see till their vicinity, an object's visual appearance can be reconstructed up to an arbitrary accuracy by inverting the filtering transformation [135] (Fig. 3.3).

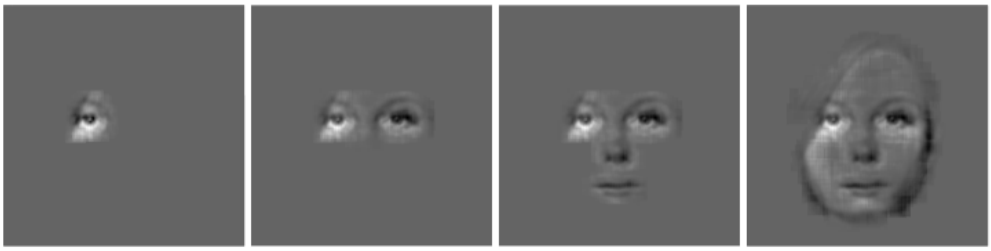


Figure 3.3: Reconstruction by inverting multi-resolution Gabor responses at an increasing number of locations.

3.1.3 Parameter Selection

In [86] authors established formulas defining relationtips between parameters of the multi-resolution Gabor features f_{max}, k, M, N, γ and η through a parameter p denoting the intersection point of two adjacent Gabor filters. The most intuitive parametrization was achieved by defining an effective envelope cross point at $p = 0.5$, which means that all Gaussian envelopes cross on the half magnitude with their neighbours. This value has been experimentally tested and shown to provide sufficient "shiftability" [156, 152], i.e. information loss is tolerable even if the signal's frequency content falls between bank filters. If the cross point parameter p remains fixed, then the adjustable parameters are the highest frequency f_{max}, the scaling factor of Gabor filter frequencies k, the number of frequencies M and the number of orientations N. The bandwidths γ and η are automatically set using formulas from [86]. See Figure 3.4 for illustrations.

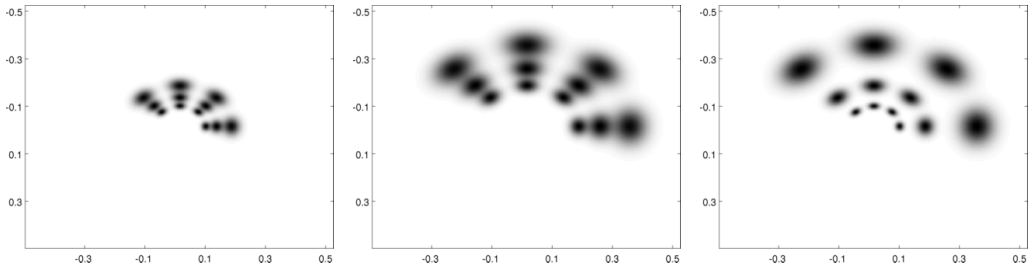


Figure 3.4: Examples of a multiresolution set of Gabor filters in the frequency domain: (a) $M = 4$ orientations and $N = 3$ frequencies, (b) the base frequency f_{max} increased, (c) the frequency scaling factor k increased.

A frequently overlooked problem in using multiresolution Gabor features to learn and detect local object parts, and using Gabor features in general, is how the feature parameters, the maximum frequency (finest scale) f_{max}, the frequency scaling factor (scale "jump") k, and the numbers of orientations M and frequencies N, should be selected. The optimal values always depend on the specific application, but for better understanding of their meaning, generic guidelines for the ideal case of unlimited resources can be defined as:

f_{max}: use the highest possible frequency in the Nyquist limit to capture even the finest details,

k: use very dense frequency spacing to notice even the smallest scale changes,

M: use all frequency bands from the highest to the lowest frequency that covers the whole image, and

N: use very dense orientation spacing to notice even the smallest rotation changes.

The above ideal case guidelines would provide ultimately traceable and accurate features but cannot be realized in practice due to two fundamental limitations: limited computational resources and limited amount of training data. Computational resources, especially space and time, set practical limits within which local features must be learned from training data and extracted from input images. Dense frequency and orientation spacing also affect computation time due to invariance shifts; matrix shifts (Equations 3.7 and 3.8) are fast operations, but pdf evaluations over all shifts using Gaussian Mixture Model (Section 3.3) become a bottleneck. However, the amount of training data is the main limitation as learning requires images and manually annotated landmarks.

In the conducted experiments the best set of Gabor bank adjustable parameters was exhaustively searched for each object category from the following set:

$$\begin{aligned} f_{max} &\in \{1/9, 3/40, 1/20, 1/30, 1/45\}\,, \;\; k \in \left\{\sqrt{2}, \sqrt{3}\right\} \\ M &\in \{6, 8\}\,, \;\; N \in \{4, 6, 8\} \;\; . \end{aligned} \tag{3.10}$$

f_{max} is defined in pixels, thus is related to the image size. These frequency values give good results for the original size of Caltech images and images from ImageNet scaled to have highest dimension equal to 300 pixels. The optimal parameters of a bank of Gabor filters for Caltech categories are:

$$f_{max} \in \{1/20\}\,, \;\; k \in \left\{\sqrt{3}\right\}\,, \;\; M \in \{6\}\,, \;\; N \in \{4\} \;\; . \tag{3.11}$$

3.2 Spatial Alignment

In natural images objects vary in their pose and scale, therefore if proposed multi-resolution Gabor features are directly extracted from the original images, they not only code visual appearance but are also perturbed by geometric distortion noise. Geometric distortions can be removed by geometrically normalizing the images, i.e. by registering all images to a common frame. Nevertheless, exact alignment is rarely possible due to natural variation of part constellation. This section presents the developed procedure for registering images to the so called "mean object space" (Figure 3.5).

In earlier works, Burl et al. [121] solved the alignment problem by fixing the locations of two points. In his solution, variation in locations of the fixed object parts is transformed to others while requiring a guaranteed detection of these fixed parts in all images. This disadvantage can be overcome if instead of just two, locations of all of the object parts are fixed during alignment, i.e. a seed image (object) is selected. Then, prior to feature

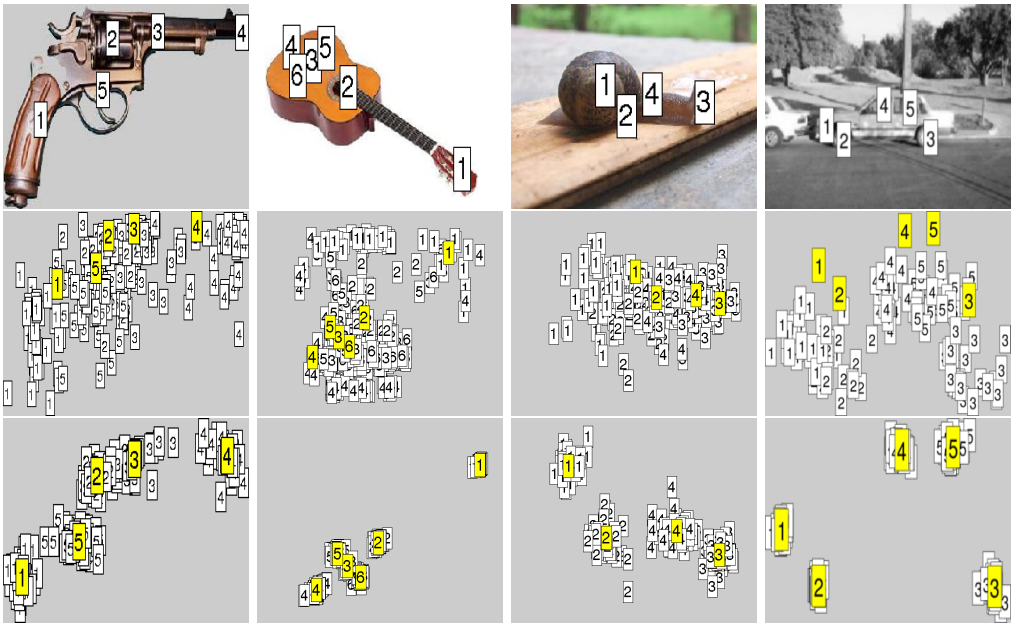


Figure 3.5: Top row: annotated object landmarks; middle: landmarks of 50 examples in the same global space; bottom: the same landmarks in the estimated aligned object space.

extraction, landmarks from all the images are transformed to this random seed using 2D homography estimation. However, in this approach the seed selection affects the final result, e.g. alignment performance degrades if a rare category example is chosen as a seed.

Algorithm 3.1 Aligned mean object space.

1: Select a random seed image and use its parts' locations as the initial object space.
2: **for all** images **do**
3: Estimate isometry/similarity transformation H to the object space using the part coordinates and the Umeyama's method [170]). //Store as H_{prior} for Algorithm 4.1.
4: Transform object's parts and bounding box coordinates to the object space.
5: Refine the object space by computing average of transformed points.
6: **end for**
7: Return the mean object space and transform all images, landmarks and bounding box corners.

Finally, to form an aligned object space, the work adopts the mean shape model by Cootes et al. [32]. Their iterative method is used, where the approximate similarity transformation is replaced by linear similarity estimation using the Umeyama method [170] (modification of a Procrusters algorithm). The method iterates through training images simultaneously updating the mean object space (see the pseudo algorithm in Algorithm 3.1 and Figure 3.5 for illustration), as a result providing the optimal pose for landmark/object detection if the training and test set are equivalent. Here, for efficient search among the

object detection hypothesises, a set of allowable transformations $\{\boldsymbol{H}_{\text{prior}}\}$ is estimated for Algorithm 4.1. $\{\boldsymbol{H}_{\text{prior}}\}$ is constructed from the set of similarity transformations $\boldsymbol{H}$ in Algorithm 3.1.

3.3 Appearance Model for Object Parts

Construction of a probabilistic model for object parts consist of two stages: 1) extraction of local image features $\boldsymbol{g}(x, y)$ and 2) their probabilistic representation $p(\boldsymbol{g}, F_i)$, where F_i denotes an object part. The main properties/requirements of an appearance model is the ability to classify and rank features of the new observations (test images) based on statistics of the training images. This kind of model is often realized in the form of a conditional density function. Therefore, the appearance model is defined by fitting a model to the observed features. The drawback is that many of the object parts should be represented with multi-modal distributions, e.g. opened/closed eyes vs glasses, neutral mouth vs a smile vs a beard, what implies a complex multi-modal nature of the probability density function. In [85] SVM and GMM were compared as probabilistic feature classifiers, and GMM was found to be superior to SVM. Moreover, GMM allows construction of a feature classifier without negative examples, resulting in the object detection method learning from positive examples only.

3.3.1 Gaussian Mixture Model (GMM)

For a proper generative model, the landmark specific conditional probability density functions (pdf), $p(\boldsymbol{g}|F_i)$, need to be estimated. F_i denotes a class specific landmark and $\boldsymbol{g}$ is a multi-resolution Gabor feature vector formed from the matrix $\mathbf{G}$ (Equation 3.4) by concatenating the responses row-wise

$$\boldsymbol{g} = (\mathbf{G}(1,1)\dots\mathbf{G}(1,N)\ \ \mathbf{G}(2,1)\dots\mathbf{G}(2,N)\dots\dots\mathbf{G}(M,1)\dots\mathbf{G}(M,N))^T\ . \tag{3.12}$$

The feature dimension of the probability density function (pdf) is the number of frequencies times the number of orientations, $D = M \times N$. The elements of $\boldsymbol{g}$ are complex-valued $\boldsymbol{g} \in \mathbb{C}^D$ because it was found in [85] that the popular magnitude representation destroys important information for detection. The Gaussian distribution for complex random vectors ($\boldsymbol{x} \in \mathbb{C}^D$) is defined as [71]:

$$\mathcal{N}^{\mathbb{C}}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\pi^D |\boldsymbol{\Sigma}|} \exp\left[-(\boldsymbol{x}-\boldsymbol{\mu})^H \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{3.13}$$

where $\boldsymbol{\mu}$ is the mean, $\boldsymbol{\Sigma}$ the covariance matrix and ^{H} denotes the adjoint matrix (transpose and complex conjugate). The complex-valued Gaussian mixture model (GMM) probability density function can be defined as a weighted sum of Gaussians:

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{c=1}^{C} \alpha_c \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}) \tag{3.14}$$

where α_c is the weight of the cth component and $\sum_{c=1}^{C} \alpha_c = 1$ (note that the complex superscript in the normal distribution is omitted for clarity). Now, a Gaussian mixture model probability density function can be completely defined by the parameter list [49]

$$\boldsymbol{\theta} = \{\alpha_1, \boldsymbol{\mu_1}, \boldsymbol{\Sigma_1}, \dots, \alpha_C, \boldsymbol{\mu_C}, \boldsymbol{\Sigma_C}\}\ . \tag{3.15}$$

The motivation to use the complex space $\mathbb{C}^D$ lies in the fact that this is the natural representation of the complex-valued Gabor responses and it reduces the number of free parameters from $C(2D^2 + 3D) + C - 1$, where the complex values are separated to the real and complex part or phase and magnitude ($\boldsymbol{x} \in \mathbb{C}^D \rightarrow \mathbb{R}^{2D}$), to

$$C(D^2 + 2D) + C - 1 \ . \tag{3.16}$$

In [85] several algorithms for GMM pdf (Eq. 3.15) parameter estimation were compared. It was found that the maximum-likelihood estimation by the standard expectation maximization (EM) algorithm [16] is best if the number of components C is known. This, however, is not possible for unknown feature densities and therefore unsupervised methods need to be used. In [85] two popular methods were tested, Figueiredo and Jain (FJ) [60] and greedy EM (GEM) by Verbeek et al. [176], and in the experiments the FJ algorithm performed better and was more stable with limited data. In the utilized implementation of the FJ algorithm, steps are replaced with complex number equations and on every iteration the Gaussian covariance matrices are tested and enforced to proper Hermitians, stabilizing the estimation. For more details on unsupervised Gaussian mixture model estimation of complex valued variables, see [129].

3.3.2 Randomized GMM

Virtually all works using Gabor features use a small bank of Gabor filters [179, 75, 191], typically 4-6 orientations and 3-5 frequencies, and all responses form the feature vector. This structure of filter bank may cause nearby overlapping filters to correlate strongly and a signal can often be detected already by a subset of the filters, making many filters redundant. The estimation becomes difficult and a large number of training examples is required. To circumvent these problems, this work proposes to use a large Gabor bank to make sure that different frequency and orientation characteristics of each part are available and a randomization procedure picks the most important filter combinations for each part (Figure 3.6).

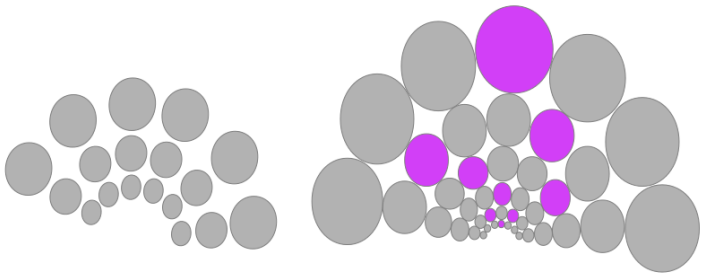


Figure 3.6: Examples of a fixed size traditional bank of Gabor filters and a bank used in a novel randomized GMM approach. Left: The traditional approach – a small Gabor filter bank from where responses of all filters form the feature vector (e.g., [179, 75, 191, 87]); Right: a large bank from where the part characteristic frequencies are selected by the random optimization process. [141]

As already mentioned in Section 3.1.3, the bigger the bank of filters the more precisely the object parts are represented. However, a straightforward augmentation of the filter

bank would require an increased amount of the training data, which is not always available, and heavier computations, making the detector less efficient. Eventually, this kind of expansion does not improve description of the object parts and consequently their detection. The performance gain is not achieved as a lot of uninformative features are introduced into the feature pool along with the useful features. In contrast to simple expansion, the proposed novel randomized GMM allows each object part to be represented with a unique set of Gabor features (a subset of the full bank of filters). The randomization procedure solves the problem of data deficiency and provides a very discriminative representation of the object parts.

For a small number of training examples, even a subset of features may overfit, leading to the low bias high variance problem. In machine learning literature decision trees are known to have a similar problem, and therefore inspired by a meta-method, random forests [24], for decision trees, it was decided to similarly adopt bagging and randomization to unsupervised Gaussian mixture models in this work: *randomized Gaussian mixture models*. A pseudo code for the randomized GMM estimation procedure is given in Algorithm 3.2.

Algorithm 3.2 Randomized Gaussian mixture model pdf estimation.

```
Perform 4-fold separation of the training images to a training set and a validation set
Compute filter responses with a large Gabor bank at all part locations.
for T iterations do
    Randomly select K filter responses from all training set images
    Run F-J GMM estimator using the selected features
    Evaluate landmark detection accuracy for the validation set
    Store the filter set and its performance
end for
Choose the B best sub-banks of filters and use their combination as the GMM pdf
```

The randomization procedure from Algorithm 3.2 chooses K filters from the full bank of Gabors at random and evaluates the performance of this randomized descriptor based on the training data (according to the 4-fold procedure). For better generalization, the whole process is repeated T times (e.g. $T = 50$). The final descriptor is composed of B best performing sub-banks of K filters. The chosen sub-banks of Gabor filters are applied to each pixel of an image. The received responses are then converted into B likelihood maps using the corresponding GMM. In all experiments, K was equal to 9. This value is the highest number of random filters, which guaranteed successful training of an FJ GMM algorithm with less than 30 training images.

The part detection becomes slightly trickier with B random Gaussian mixture model pdfs per one object part. Thus a procedure to combine the likelihood maps of each GMM and non-maximum suppression to select the best candidates from an input image is needed. The pseudo code of the detection procedure is in Algorithm 3.3.
The Gabor object part descriptor, composed only from a small portion of the original Gabor bank filters, yields a lot of false positives. Nevertheless, each of the B sub-sets has a different set of false positives while having high responses at the same true locations. In order to suppress the false positives, the most strict rule for combining classifiers is used, the product rule [96]. To further prune false positives and simplify calculations,

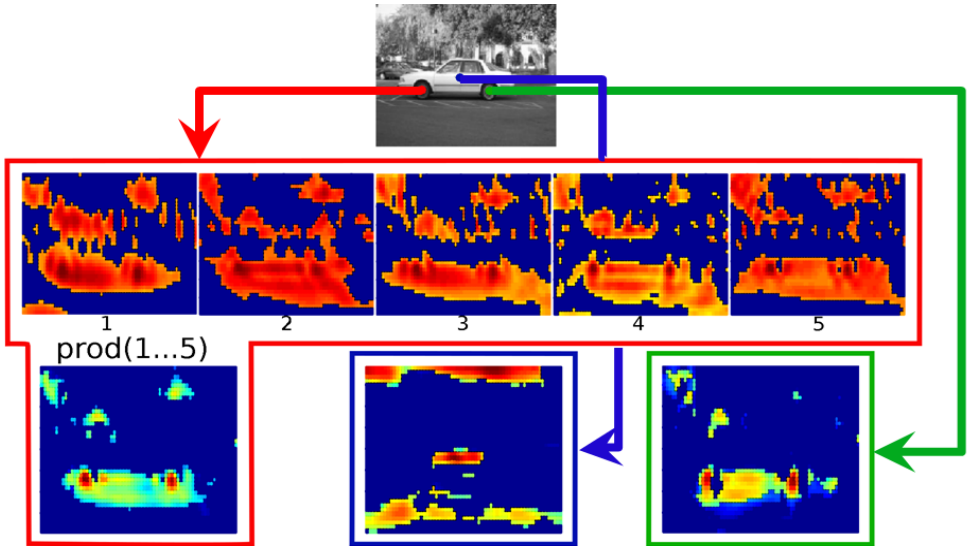


Figure 3.7: Illustration of object part likelihood map formation. At the top is a car image from which three different landmarks should be detected. The second row shows the 5 thresholded likelihood maps of the $B = 5$ Gaussian mixture models corresponding to a single landmark (a front tyre). The third row shows the combined likelihood maps of all three landmarks. Warm colors denote high likelihood.

Algorithm 3.3 Detection using rand-GMM & Gabor features.

1: Apply B sets of K Gabor filters
2: Compute B likelihood maps using the estimated GMMs
3: Threshold each likelihood map to retain the proportion of P_1 highest likelihoods
4: Compute the product likelihood of the B thresholded maps
5: Apply recursive global maximum search with suppression

likelihood maps prior to multiplication are thresholded so that only 40% ($P_1 = 0.4$) of the highest values of each likelihood map (Fig. 3.7 2nd row) contribute to the final likelihood map (Fig. 3.7 3rd row). Pixels with the highest product likelihoods are the best candidates for true object parts locations (Fig. 3.7 3rd row red areas). To find these locations (local maxima) a simple yet effective method of global maximum search with consecutive suppression is used. During this process, after a global maximum is found, likelihoods in the area around it are assigned to zero. The size of the suppression region is defined by the discrimination ability of the Gabor features, i.e. the lowest frequency of the Gabor bank. Suppressing likelihoods in the neighbourhood of a maximum forces the part detector to find candidates from different peaks exploring the whole image area. Consecutive maximum search and suppression are continued until a desired number of candidates is provided or the whole likelihood map has been explored (i.e. there are no longer any non-zero elements). Thus, the proposed part detector provides a varying number of landmark candidates by adapting to the difficulty (stability of representation) of a specific landmark. Parameter B in the experiments was set to 5.

3.4 Experiments

3.4.1 Data and Parameter Settings

Natural and divergent categories were selected from the Caltech-101 dataset for experimental testing of the object part detection. Images of each category were randomly assigned into approximately equal sized training and testing parts. The developed part detector tolerates small amounts of training data, thus categories with 28 to 406 training images were used in the experiments. Categories selected from Caltech-101 are: *airplanes* (406 train/394 test images), *car side* (58/65), *dollar bill* (28/24), $faces\ easy$ (206/229), *motorbikes* (377/412), *revolver* (41/41), *stop sign* (30/34), *watch* (118/121), *yin yang* (30/30), *menorah* (43/44), *grand piano* (49/50) and $dragonfly$ (34/34). For each selected Caltech-101 category from 3 to 5 semantically meaningful object parts were manually annotated. The developed part detector was also tested on the BioID face database containing 1521 images of human faces (507/1024) recorded under natural conditions, i.e. with varying illumination and complex background. Faces in the images vary in size, have different facial expressions, facial hair and glasses, belong to different gender and racial groups. The experiments used the FGnet Markup Scheme of the BioID Face Database, which has 20 landmarks useful for facial analysis and recognition [138, 43, 189].

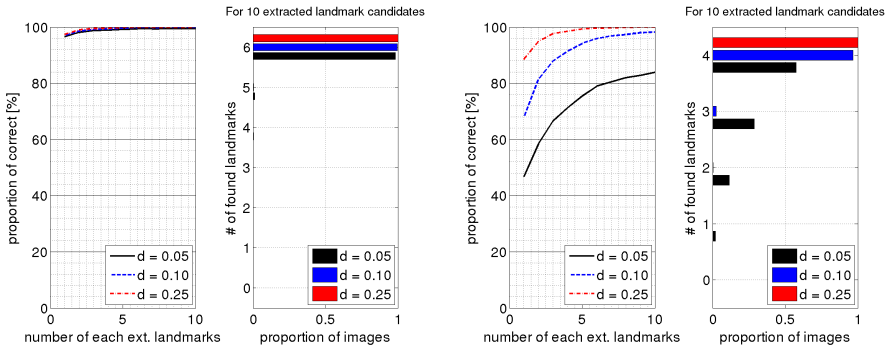
A quantitative comparison of the proposed part detector and state-of-the-art results is presented for facial landmark detection with the BioID dataset. Moreover, although the developed part detector is general (can be applied to any object class) no tailoring or parameter tweaking for facial landmark detection was applied.

3.4.2 Performance Evaluation

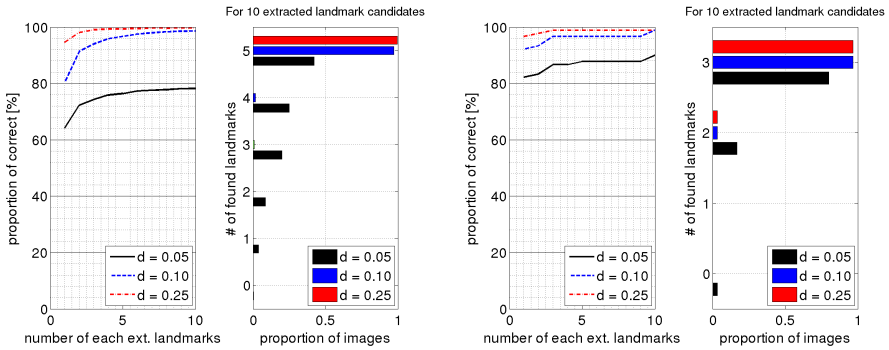
To the author's best knowledge no performance evaluation method for local part detection has been proposed. Thus the evaluation metric is derived from the d_{eye}-measure used for measuring face localization accuracy [76, 145]. This metric computes distances from predicted coordinates of the object parts to the groundtruth coordinates that are normalized by the diagonal of the corresponding bounding box. For object detection purposes, the normalized distance ≤ 0.05 is considered as excellent, ≤ 0.10 good and > 0.25 as a failure (thresholds are adopted from face detection [76, 145]). Using these fixed thresholds, a number of the best landmark candidates falling within the thresholds can be suggested. The sooner the correct landmarks are fetched, the better. As a result, either a cumulative detection curve or detection bars can be plotted. The curve denotes how fast the correct landmarks are found by increasing the number of fetched landmarks. While the graph reports at landmark level, detection bars report at image level, i.e. for how many images at least some number of landmarks are detected for a fixed maximum error (0.05, 0.10, 0.15, 0.25). The latter information is important because the sampling method in the object detection pipeline (Algorithm 4.1) is expected to perform poorly for images with less than three landmarks detected correctly (two landmarks used to estimate a transformation and at least one to verify it). These performance graphics are illustrated in Figure 3.8.

3.4.3 Visual Class Landmarks (Caltech/ImageNet) Detection

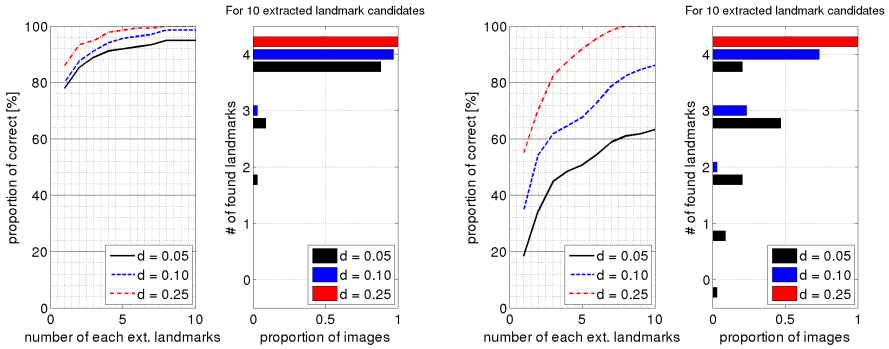
Evaluation of landmark detection for a visual class introduces a quantitative detection measure similar to the facial landmarks by dividing the pixel detection distances by the bounding box diagonals. The detection graphs and bars for several image categories are



(a) Faces_easy **(b)** airplanes



(c) Motorbikes **(d)** yin_yang



(e) stop_sign **(f)** dragonfly

Figure 3.8: Caltech-101 landmark detection (left: cumulative detection graph, right: detection bars for 10 best candidates).

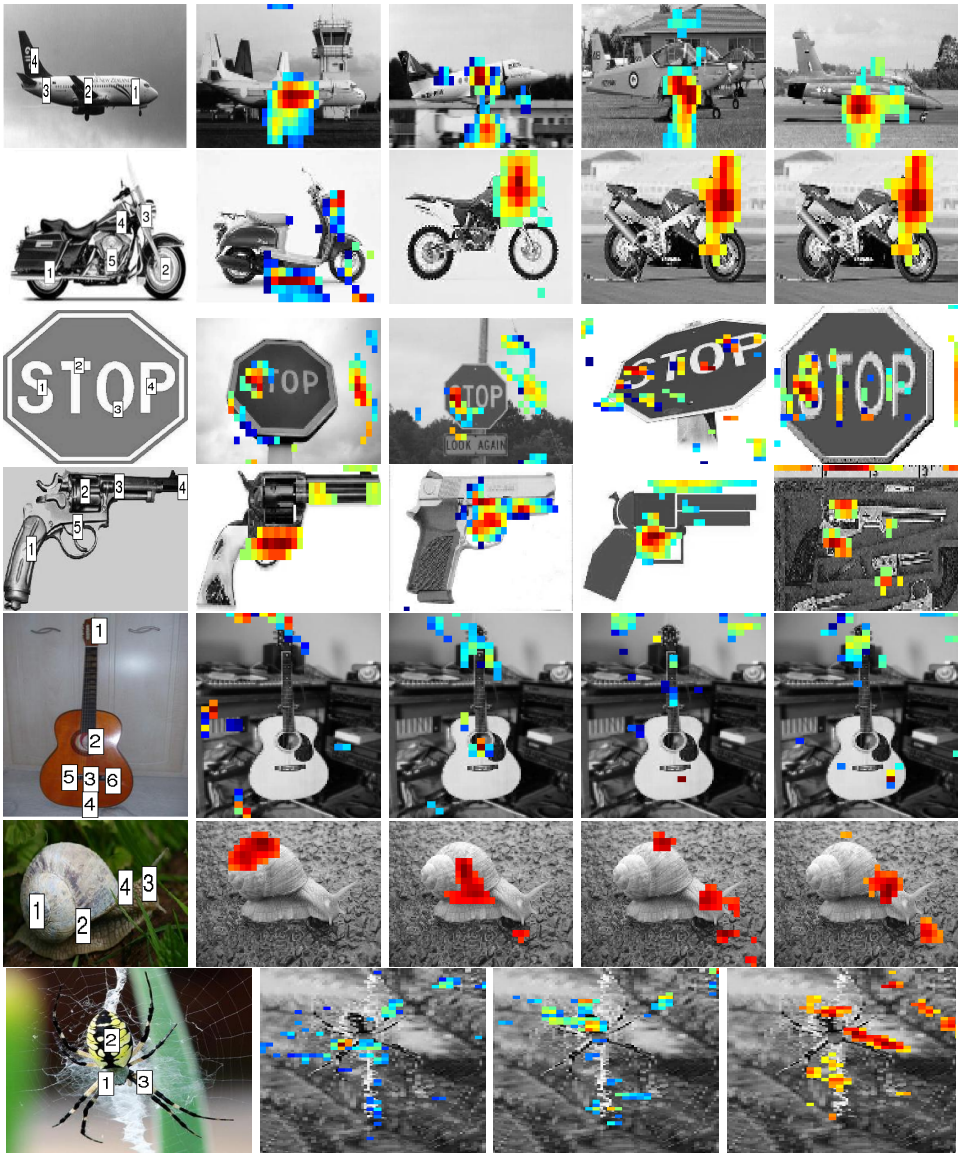


Figure 3.9: Example likelihood maps of different object classes. First four rows illustrate probabilistic detection of landmarks in Caltech-101 images. On the left are all annotated landmarks. Images on the right demonstrate detection of the same landmark (*airplanes*: #2, *motorbikes*: #3, *revolvers*: #5, *stopsign*: #1) in different images. The last three rows show probabilistic detection of landmarks in ImageNet images. Examples of detection of different landmarks in the same image (*acousticguitar*: #1,#2,#3,#6; *snail* and *gardenspider* all landmarks).

shown in Figure 3.8. The detection accuracies as the function of the best candidates

were studied for the visual classes and the results verified that 5-10 best candidates are needed to fetch a sufficient number of correctly detected landmarks (at least 80% of the landmarks detected within an accuracy of 0.10). The results demonstrate that the method has almost perfect performance for the *faces* class. The second best class is *motorbikes* and the third *airplanes*. These three are the classes with the largest amount of training images. The difference between *faces* (for 0.10 almost all landmarks were detected in all images), *motorbikes* (97%) and *airplanes* (95%) can be explained by the fact that the *faces* class is much easier compared to *motorbikes* and *airplanes*, which contain many sub-classes. Results for the best performing classes are followed by *yin yang* and *stop sign*, i.e. classes with the most simple structure. The worst performing class is *dragonfly*, which has the biggest appearance variation and only few training examples. Detection likelihoods are illustrated in Figure 3.9. Detection graphs and bars for the remaining Caltech-101 categories are presented in Appendix I.

Effect of Feature Normalization

This experiment was implemented to show the importance of feature normalization for illumination invariance (see Equation 3.9) and to demonstrate the improvement in object part representation achieved by randomization of the Gaussian mixture model (Section 3.3.2). Experiments were conducted with the full bank of Gabor filters and its randomized version. With the full Gabor bank (parameters from 3.11) the Gaussian mixture model converges with approximately 200 training images, thus two Caltech-101 classes satisfying this condition (*airplanes* and *motorbikes* with 406 and 377 training images correspondingly), were chosen for evaluation. The effect of feature normalization is most evident for the *airplanes* category, the more difficult of the two. Part detection results for *motorbikes* remain almost unchanged with feature normalization, however randomization of GMM affects both categories, ensuring better results with a smaller number of detection candidates.

Table 3.1: Object part detection results (proportion of correctly found object parts within 5 best candidates for the 0.10 accuracy margin) using different types of feature normalization.

	airplanes(%)	motorbikes(%)
no normalization	22	84
zero mean	13	80
zero mean std-one	9	75
L1	76	89
L2	69	86

Several types of feature normalization were investigated in this work: zero mean, zero mean with standard deviation equal to one, L_1 and L_2 norm. Detection results using these normalizations applied to *airplane* and *motorbike* images are presented in the Table 3.1. The table contains results obtained during object part detection on the test images. Numbers correspond to the proportion of correctly found object parts within 5 best candidates for the 0.10 accuracy margin (blue curves in the Figure 3.10). The experiments were conducted using the full bank of Gabor filters with parameters from

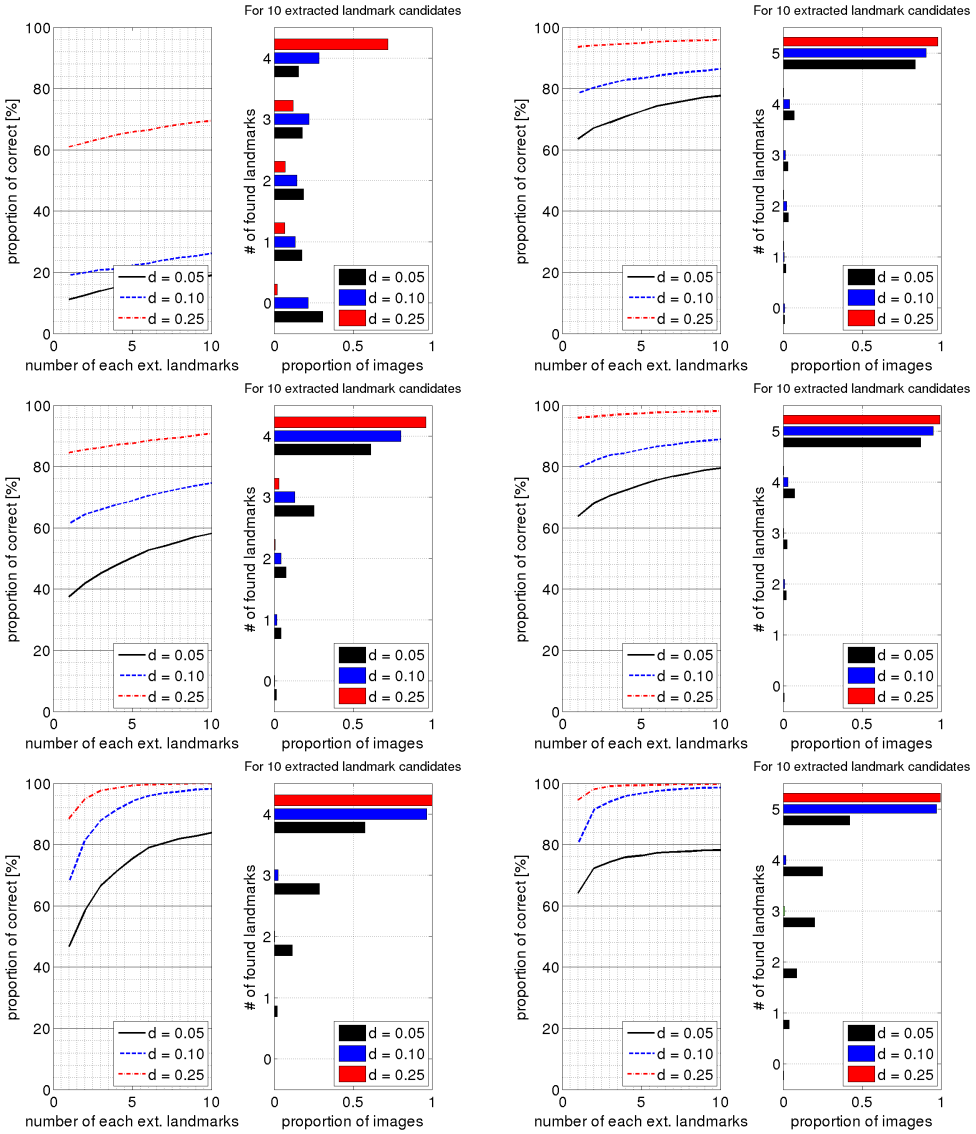


Figure 3.10: Effect of feature normalization and GMM randomization on object part detection of *airplanes* (left) and *motorbikes* (right) categories from Caltech-101. The top row corresponds to results obtained with the full Gabor bank without feature normalization. The middle row gives results with the normalization (Equation 3.9) enabled. The bottom row shows part detection results using randomized GMM (from the same bank of Gabor filters) with feature normalization.

Equation 3.11. Table 3.1 shows that the part detector achieves best performance with the L_1 norm. However, in experiments with the randomized Gaussian mixture model

the L_2 norm performed favourably compared to the L_1 norm, and thus L_2 normalization (Equation 3.9) was used in all other experiments of this work.

3.4.4 BioID Facial Landmarks Detection

Here generative part detector is applied to the well-known problem of facial landmark detection. The results are reported for the BioID database according to the BioID evaluation protocol. Obtained results are compared to the state-of-the-art detectors: LEAR by Martinez et al. [120], MK-SVM by Rapp et al. [138] and CLM by Cristinacce et al. [36]. Figure 3.11 shows dependency of a proportion of correctly detected object parts as a function of distance from their true locations. The Gabor bank parameters used in this experiment are: $f_{max} = \sqrt{3}/20$, $k = \sqrt{3}$, $M = 7$, $N = 4$ with 1 scale shift. For each object part only one best candidate detection is considered in performance evaluation.

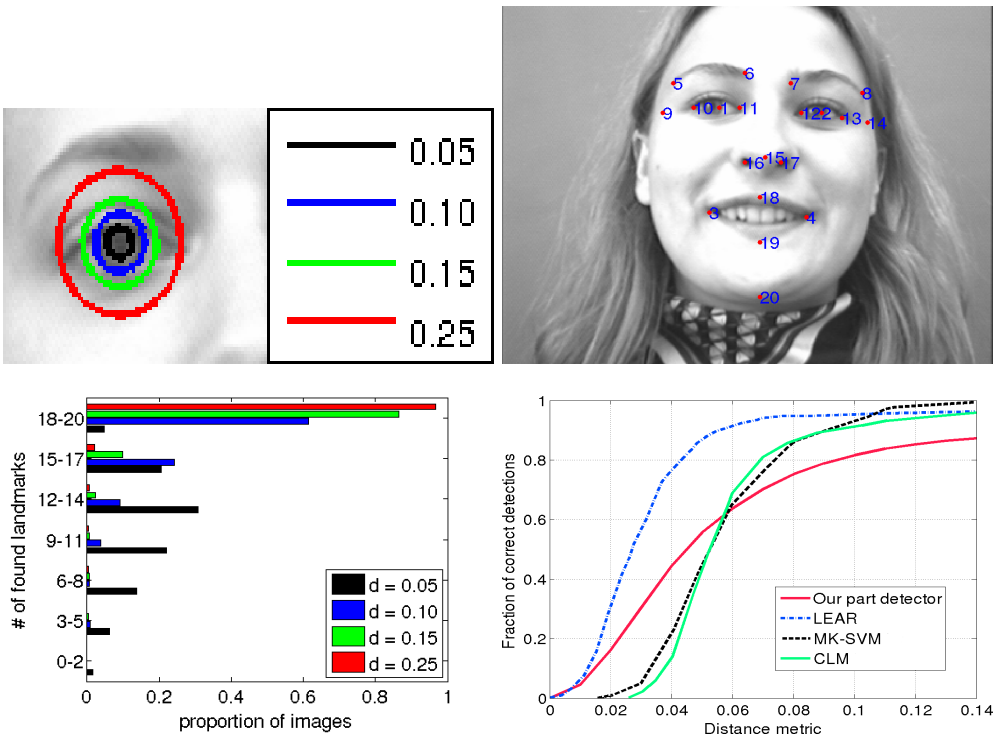


Figure 3.11: Object part detection results for the BioID dataset. Top: illustration of the detection thresholds and example of BioID landmarks. Bottom: detection bars and cumulative error graph.

It is noteworthy that the developed part-detector without any special processing for facial parts performs comparably to very dedicated facial landmark detection methods from the recent literature. The proposed detector misses about 10% of the most difficult landmarks. On the other hand, more than half of the landmarks are correctly found in 73% of the images, even for the most strict metric (≤ 0.05). For the less strict metrics

more than 10 landmarks per image are found in $90 - 98\%$ of the images, while for successful object class detection already 3 correctly detected landmarks are sufficient. It should be noted that all other methods are discriminative and include special processing and a full facial landmark model, whereas the developed method just returns the one best candidate of each landmark with no spatial regulation. Examples of facial part detections are given in Figure 3.12.

Figure 3.12: Example detections for BioID images in different scale and illumination conditions.

3.5 Summary

This chapter began with an extensive description of Gabor features, their properties and parameters. Gabor features together with a Gaussian mixture model form an appearance model for object parts but during the course of experiments it was found that about 200 training images are required for convergence of the GMM with the selected parameters of the Gabor bank. Therefore a randomization procedure was developed. Gabor bank randomization allows to relax the limitation of the required number of training images and improves object part representation by making it part specific, i.e. by avoiding uninformative frequencies and orientations of multidimensional Gabor features. The Gaussian mixture model, which outperformed the one-class support vector machine classifier [153] in [85], enables learning from positive examples only. This ability to learn without negative examples sets apart the proposed part detector and based on it object class detector from the mainstream where the background class is modelled either for every category

separately or only once, being the same for all categories. The experiments show that Gabor features and a Gaussian mixture model are a good combination for object part description and detection. Results obtained with Caltech-101 categories demonstrated good performance for all chosen categories. Moreover, comparison of the developed general part detector with specialized state-of-the-art facial landmark detectors confirmed the effectiveness of the proposed method. Therefore, it is logical to adopt the introduced part-detector in the more popular task of object class detection, which is described in the following chapter.

Chapter IV

Part-Based Gabor Object Detector

Object detection is a widely investigated task in computer vision. Many approaches exist and many challenges need to be tackled, e.g., large variations in scale, pose, appearance and lighting conditions. Part-based object representation [1, 34, 54, 55, 56, 140, 141] is one of the popular approaches to object detection. However, part-detector based methods [15, 79, 166, 53, 185] often require manually annotated landmarks in the training images. In [53], similar to the concept of this work, Gaussian derivatives (steerable filters) are adopted and part pdf's estimated by a single diagonal Gaussian. Manually annotated object parts are used in the method by Bergtholdt et al. [15]. The landmark selection was partially automated in [79], where landmarks were constructed from the object outlines, but they can be distributed uniformly within a bounding box (see experiments in section 4.5.5) if the training images are aligned with recent unsupervised alignment procedures [33, 188]. However, the information provided by object parts in part-based models with automated part detection, e.g. [55], cannot explicitly localize the object parts having semantic meaning (e.g. eyes, tires, handle).

The detector proposed in this work is strongly supervised and, unlike weakly supervised methods based on bounding box information, represents objects with manually annotated class specific landmarks. Given this strong label information, i.e. learning using privileged information [174], the developed framework can get benefits from additional sources of useful information related to the object detection task and significantly boost the training step. When provided with annotated images, the detector learns probabilistic models for both class landmarks and their spatial variation, i.e. the constellation of the class parts. The constellation model is based on the mixture of Gaussians, while the appearance of the parts is described with Gabor features and the Gaussian mixture model in Chapter 3. The full pipeline of the proposed generative part-base Gabor object class detector is shown in Figure 4.1.

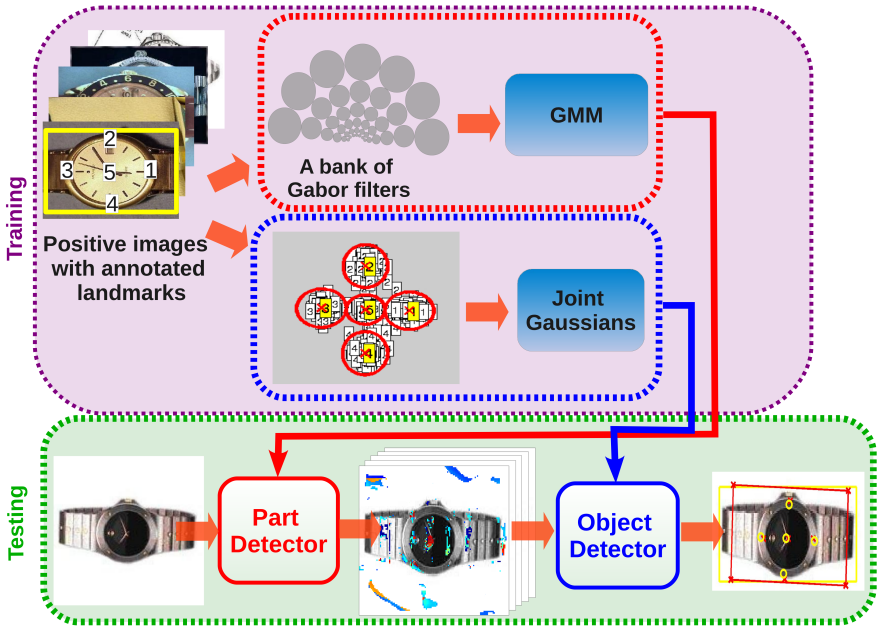


Figure 4.1: Workflow of the developed generative part-base Gabor object class detector.

4.1 Object Pose Clustering

Analysis shows that many objects in popular datasets (Caltech-101, Caltech-256, Pascal VOC and ImageNet) are captured from a limited set of viewpoints. This fact is easy to explain with the laws of physics, scene structure, and the way people capture images; pictures of sofas are usually frontal as their backs are turned towards walls; humans are almost always photographed vertically while being awake. The local part detector in this work can find parts in any scale or rotation via the rotation and scale shifts described in Section 3.1, but this is effective and efficient only if the pose distribution is uniform. Otherwise the experimental results will be inferior to methods that are not necessarily invariant but exploit the quantized pose property of the datasets. For example, the Deformable Part-Based Model [55] clusters training image bounding boxes and trains a separate detector for all clusters. It is noteworthy that their heuristic method is effective only if a class has different dimensions in different views (guitar, car). The invariance shifts (Equations 3.7 and 3.8) would be inefficient with sparse clusters in orientation and/or scale since most of the shifts would not provide any detections. To improve the method the quantized inhomogeneous properties of the datasets are exploited and several (3 in the experiments) different object models used during the training phase. To solve the pose quantization task standard K-means clustering is used to find the dense regions. However, instead of bounding boxes, which easily fail with objects of almost equal dimensions, the transformations H of the images to an arbitrary mean object space (see Algorithm 3.1) are used, which are shown as 2D points in the scale

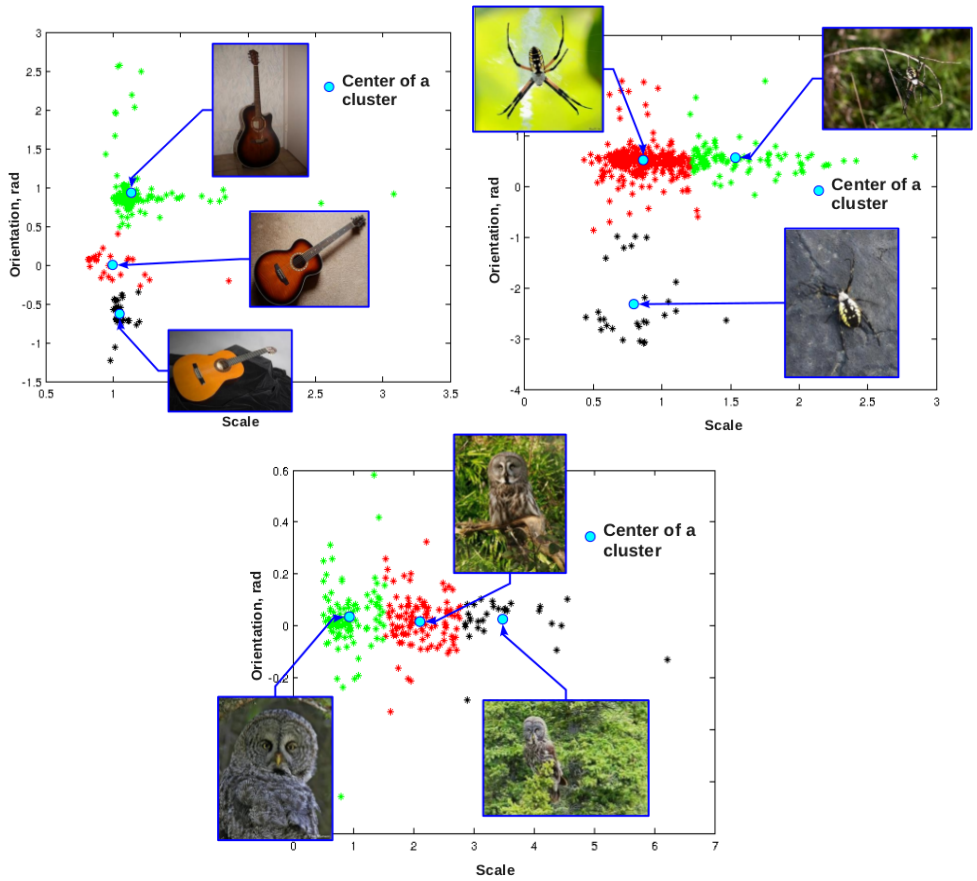


Figure 4.2: Examples of ImageNet classes that are clustered in their poses. Colors denote the discovered pose clusters. For example, *guitars* are mainly quantized in orientation while *owls* in scale.

- orientation space in Figure 4.2. To form pose clusters the images are aligned to an arbitrary mean space, i.e. a random seed image is used, because seed selection does not affect the results of the clustering as the absolute values of the scales and angles are not important for grouping. Examples of discovered clusters with their representatives are presented in Figure 4.2. More importantly, if the viewpoint changes are close to in-plane (2D), the full training data can still be used to train a model for each cluster. For this purpose, all training images are transformed into the new cluster specific canonical space, i.e. aligned using the cluster center as a seed.

4.2 Constellation Model

The starting point of the probabilistic constellation model representation is the aligned object space defined in Section 3.2. In the canonical space location variance, of the ob-

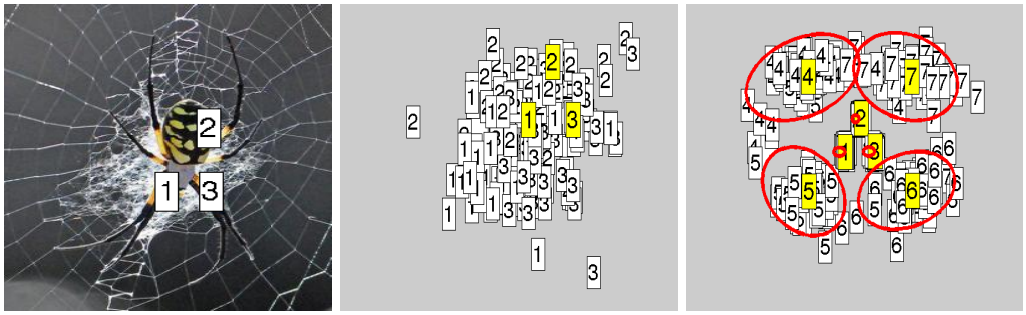


Figure 4.3: Illustration of constellation model generation using spatial alignment. Left-to-right: annotated object parts; 50 training examples plotted in the original space; the same examples plotted in the object space with their ImageNet bounding boxes corners (see Algorithm 3.1). Yellow labels denote the mean locations and red ellipses the area of two standard deviations around the mean locations. Parts from 4 to 7 correspond to the box corners, which have much higher uncertainty (bigger ellipses) compared to aligned object parts.

ject landmarks is minimized with respect to the selected transformation type (similarity transformation in the experiments). A likelihood density representation is introduced in the canonical space by capturing each landmark location using a 2D Gaussian (see Figure 4.3). The joint pdf of the spatial constellation is $p(\boldsymbol{x}'_1, \boldsymbol{x}'_2 \dots \boldsymbol{x}'_N | \boldsymbol{\theta}_C)$, where $\boldsymbol{\theta}_C$ is the set of parameters of N 2D Gaussians representing $i = 1 \dots N$ parts $\boldsymbol{x}'_i$ in the aligned space for the class C. Assuming parts' locations independence, the constellation pdf is (the object space notation $'$ and indexing with C are omitted for clarity):

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2 \dots \boldsymbol{x}_N | \boldsymbol{\theta}_C) = p(\boldsymbol{x}_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \dots p(\boldsymbol{x}_N | \boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N). \tag{4.1}$$

This constellation probability (4.1) is the prior in the developed generative method – without any detected observations, it gives a probability for a configuration of the parts. Note that the number of parts, N, is class specific and varies from 3 to 6 in the experiments.

4.3 Object Detection by Search

In the proposed generative probabilistic object model, the prior probability is the spatial constellation model in an estimated canonical object space and observation probabilities are found from the likelihood map of each local part provided by the part detector from Chapter 3. The desired output of the object detector is a set of the most likely hypotheses of object locations $\{hyp_{BEST}\}$ - the nominator (likelihood) in the Bayes' theorem. In order to find these most likely object locations, the detector exhaustively samples the best part candidates from part-specific likelihood maps, estimates their transformation to the canonical object space, transforms all parts and selects the ones that agree with spatial constellation model the most (i.e. have highest constellation likelihoods). The sampling procedure is simple to implement and is effective in the experiments. In the case of a large number of candidates, an efficient random sampling procedure can be used instead of the exhaustive search. Analysis of landmark detection curves (e.g. Figure 3.8)

shows that curves for most of the categories rich their saturation region when 5-10 highest scoring candidates of each part is considered in the evaluation. Thus to find a correct object hypothesis during sampling, it is sufficient to use as few as five best candidates of each part and then the exhaustive search takes only several (approx. 300) hundred iterations. Moreover, for most of the iterations only the operations up to line 3 will be executed due to the transformation prior omitting too awkward transformations. The spatial prior H_{prior} is estimated from training examples (Algorithm 3.1). H_{prior} defines the allowed intervals of scales and orientations characterising object position in the image. This allowed intervals are constructed by extending scale ($\times 1.5$) and orientation (± 15 *degrees*) intervals obtained from the training images. The pseudo code is given in Algorithm 4.1 where N_{bestLM} are the best local part candidates (see Figure 3.9). The most important factor is the detection probability computed in line 10 of Algorithm 4.1. Line 17 in the algorithm reflects prior knowledge that all bounding boxes in all images are annotated within the image boundaries. Thus, even though the detection bounding box can predict the location of truncated object parts outside the image, the final bounding box boundaries should be inside of image.

Algorithm 4.1 Object detection using the probabilistic part-based Gabor object model

1: Initialize the set of best hypotheses $\{hyp_{BEST}\}$ to null and set score values to zero
2: **for** all minimum combinations of the detected parts (2 for isometry/similarity) **do**
3: Estimate the transformation H from the image space to the object space
4: **if** $H \notin \{H_{\text{prior}}\}$ **then**
5: Skip this hypothesis.
6: **end if**
7: Transform all detected parts to the object space using H
8: For each transformed part compute the spatial likelihood (the single terms in (4.1)).
9: Select the parts with the highest likelihoods (omit if below $P_{landmark}$).
10: Compute the detection probability p (see Equation 4.2) (detection score)
11: **if** p is better than for any in $\{H_{best}\}$ **then**
12: Using all selected parts, estimate H^{-1} from the object space to the image space
13: Transform the selected parts to the image space (replace omitted parts and bounding box corners with ones from the mean shape model).
14: Add hypothesis (parts' coordinates) to $\{hyp_{BEST}\}$ (remove the worst if the max number exceeded).
15: **end if**
16: **end for**
17: Enforce all bounding box corners from $\{hyp_{BEST}\}$ to be inside of the image
18: Return the best hypotheses in $\{hyp_{BEST}\}$

4.4 Detection Score Formulation

The detection probability consists of two elements: the probability of the spatial locations of the parts in the object space and the likelihoods of the parts' appearance. The traditional approach is to use the Bayesian rule of posteriors similar to [121], but that has two problems. Firstly, to be able to apply it a good estimate of the $X\_not\_at\_that\_location$ density is needed, i.e. a number of negative examples. Secondly, object detection, a task when knowing object class its most probable location in the image is to be found, should

be based on likelihood values (the nominator in the Bayes' theorem) and is not a classification problem. Detection must be based on the highest likelihoods found for each class. Therefore, detection precedes the classification stage, which is a Bayesian task. Pruning false positives is also a Bayesian task that should be different from detection. This idea is further developed in section 5.1, where the generative detector is paired with discriminative classifiers in order to prune false positive detections.

The full likelihood consists of two elements, appearance and pose, and assuming independence of the appearance and spatial location, can be written as (C denotes class number, $\boldsymbol{\theta}^a$ and $\boldsymbol{\theta}^c$ are parameters of correspondingly appearance and constellation GMMs)

$$p(\boldsymbol{x}, \boldsymbol{g}|\boldsymbol{\theta}^a_C, \boldsymbol{\theta}^a_C) = p(\boldsymbol{x}|\boldsymbol{\theta}^c_C) \times p(\boldsymbol{g}|\boldsymbol{\theta}^a_C) == \underbrace{p(\boldsymbol{x}_1|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \cdot \ldots \cdot p(\boldsymbol{x}_N|\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)}_{\text{constellation}} \underbrace{p(\boldsymbol{g}|\boldsymbol{\theta}^a_C)}_{\text{appearance}} . \tag{4.2}$$

In this work, conditional independence for the appearance of different object parts is assumed. To solve the problem of occlusion or simply detection failure all possible combinations of 1, 2, ..., N-1, N parts visible out of N possible are used. This leads to 1st, 2nd, ..., Nth order probability terms:

$$p(\boldsymbol{g}|\boldsymbol{\theta}^a_C) = \begin{cases} p(\boldsymbol{g}_1|\boldsymbol{\theta}_{C,1}) + p(\boldsymbol{g}_2|\boldsymbol{\theta}_{C,2}) + \ldots + p(\boldsymbol{g}_N|\boldsymbol{\theta}_{C,N}) + \ldots \\ p(\boldsymbol{g}_1|\boldsymbol{\theta}_{C,1})p(\boldsymbol{g}_2|\boldsymbol{\theta}_{C,2}) + p(\boldsymbol{g_1}|\boldsymbol{\theta}_{C,1})p(\boldsymbol{g}_3|\boldsymbol{\theta}_{C,3}) + \\ p(\boldsymbol{g_2}|\boldsymbol{\theta}_{C,2})p(\boldsymbol{g}_3|\boldsymbol{\theta}_{C,3}) + \ldots \\ p(\boldsymbol{g}_1|\boldsymbol{\theta}_{C,1})p(\boldsymbol{g}_2|\boldsymbol{\theta}_{C,2})p(\boldsymbol{g}_3|\boldsymbol{\theta}_{C,3}) + \ldots \\ \ldots \end{cases} \tag{4.3}$$

which enforces to perform a summation of $\sum_{k=1\ldots N} \binom{N}{k}$ product terms. Good iterative implementations perform well up to 10-15 parts after which approximations must be utilized. In the performed experiments the number of parts was sufficiently small for exact computation.

In Equation 4.2, a strong product constraint is used for the spatial locations of the parts but a rather weak sum constraint for their appearance. The reason is that the part detection is less reliable due to detection failures (background clutter) or occlusion. The appearance part in Equation 4.2 consists of low, mid and high order probability terms, exemplified in Equation 4.3, that cannot be reduced since the high order terms dominate if almost all local parts are detected correctly and the low order terms dominate if only a small number of the parts are detected. It should be noted that Algorithm 4.1 recovers from occlusion (lines between 11 and 15) since the parts with too low constellation likelihoods, which are likely to be false alarms, are replaced with the mean parts locations and mapped back to the query image.

4.5 Experiments

4.5.1 Data

For visual class detection, several classes from the popular Caltech and ImageNet datasets were selected. Caltech, and in particular Caltech-101, are considered easy datasets as

many classes contain examples in almost the same pose. Therefore, the Caltech classes measure the method's capability to model visual appearance variation. ImageNet is a more recent large scale dataset, where examples are captured from more challenging viewpoints and many classes are difficult by their visual appearance. Images from the ImageNet categories were selected manually in order to remove images where only a small part of an object is visible. Since the training data is limited for Caltech-101, the results for Caltech-101 and ImageNet are presented using the feature randomization method with GMM-based pdf estimation (Section 3.3.2). The 12 categories with the biggest number of images were chosen from Caltech-101: *airplanes, car side, dollar bill, faces easy, motorbikes, revolver, stop sign, watch, yin yang, menorah, grand piano, dragonfly.* From ImageNet, the following categories were selected: *acoustic guitar, garden spider, grey owl, piano* and *snail*. For computational speed-up and for ease of Gabor bank parameter selection, all ImageNet images were pre-scaled so that their maximal dimension is equal to 300 pixels (original aspect ratios were preserved). All classes were randomly divided into approximately equal size training and test sets. Object parts in all images of selected Caltech and ImageNet categories are annotated manually prior to experiments. As proposed in this work, detection and classification stages are separated in the experiments, thus test sets for the detection (localization) task are composed of only positive examples, i.e. representatives of the category being tested. For the classification task, on the other hand, the test sets are formed with test images of all classes from which the system should discriminate.

4.5.2 Performance Measures

Detection evaluation in this work is based on a PASCAL VOC evaluation procedure that includes precision-recall curves and average precision. Classification results are represented with confusion matrices.

In [47] precision and recall are defined for images ranked according to their scores (the higher a score is, the more confident the system is in the result). The evaluation procedure penalizes the object class detector for: i) missing object instances, ii) multiple detections of the same instance, and iii) false positive detections. For each image, a detector is expected to return hypotheses of object locations, defined by corners of the bounding box, and corresponding confidence scores. A successful detection is defined by the overlap ratio [47]:

$$A = \frac{BB_{gt} \cap BB_{pred}}{BB_{gt} \cup BB_{pred}},$$

where BB_{gt} is the area of a ground truth bounding box and BB_{pred} - of a predicted bounding box. Detection is accepted if the overlap ratio is greater than 0.5. The detection rate corresponds to the proportion of correctly located objects ($A \geq 0.5$) in the images. Precision p is equal to the proportion of correct detections out of all detections made by the algorithm. Recall r is equal to the proportion of true-class objects that are detected by the algorithm. Average precision is also used in evaluation, it is defined as the mean value of precision at 11 levels of recall (evenly distributed from 0 to 1) and describes the shape of a curve:

$$AP = \frac{1}{11} \cdot \sum(p(r)).$$

Even though the proposed method is designed for the detection task, there is an interest in its performance for classification. In the classification framework a label corresponding to the class having the highest detection score in the image is assigned to this image. Decisions made by the classifier are presented in the form of a confusion matrix, where each value V_{ij} shows the proportion of images of class i that have been classified to the class j. Thus, the values of each raw sum to one. These accuracy measures are used to compare obtained results to other methods.

In order to use the detection score in classification it should be transformed to be non-depend on the number of object parts by averaging the score over the object parts. This transformation is necessary to make the detection scores of different object classes compatible, because objects from different categories have different numbers of object parts and the final detection score is formed as a product depending on the number of object parts. The transformation is performed as follows:

$$score_{clsn} = \frac{\log_{10}(score_{detn})}{N_{parts}}, \tag{4.4}$$

where $score_{detn} = p(\boldsymbol{x}, \boldsymbol{g}|\boldsymbol{\theta}^a_C, \boldsymbol{\theta}^a_C)$ from Equation 4.2. A logarithmic scale is used for its property $\log(x^N) = N\log(x)$, thus allowing formation of a score that is not dependent on the number of object parts.

4.5.3 Caltech-4 Object Classification

The main interest of this experiment section is comparison of the developed generative object detector with other generative methods, therefore a common benchmark, the Caltech-4 dataset, is used for evaluation. Caltech-4 contains the categories *faces*, *airplanes*, *motorbikes* and *cars rear* with 435, 800, 798 and 1155 images, respectively. The Bergtholdt et al. [15] results cannot be compared to since they report results only for the *faces* category and their own background images. Other similar work by Crandall et al. [34] omits the most difficult class, *cars rear*, which was replaced by a separate background class. However, their results are reported in Table 4.1 where the diagonal elements denote the proportions of correctly classified images. Table 4.1 also contains a classification confusion matrix for a rival method - a state-of-the-art discriminative object detector [55].

Table 4.1: Comparison of classification confusion matrices for Caltech-4.

	Our method				Crandall et al. [34]				Felzenzswalb et al. [55]			
	faces	airplanes	mbikes	cars rear	faces	airplanes	mbikes	bg	faces	airplanes	mbikes	cars rear
faces	92,11	2,63	0,44	4,82	94,90	1,40	3,70	0,00	99,12	0,00	0,88	0,00
airplanes	0,00	100,00	0,00	0,00	0,00	90,50	1,25	8,25	3,30	93,40	2,54	0,76
motorbikes	0,00	0,24	99,29	0,48	0,00	1,00	96,00	3,00	0,00	0,00	99,76	0,24
cars rear	0,00	0,69	0,00	99,31	-	-	-	-	0,00	0,52	0,35	99,13

Table 4.1 reveals the importance of landmark choice. It can be noticed that most of the misclassified images are confused with *cars rear*. The errors happen because of too general nature of *cars rear* landmarks, i.e. simple corners, which can be found everywhere with high probability, showing the necessity of more careful landmark choice.

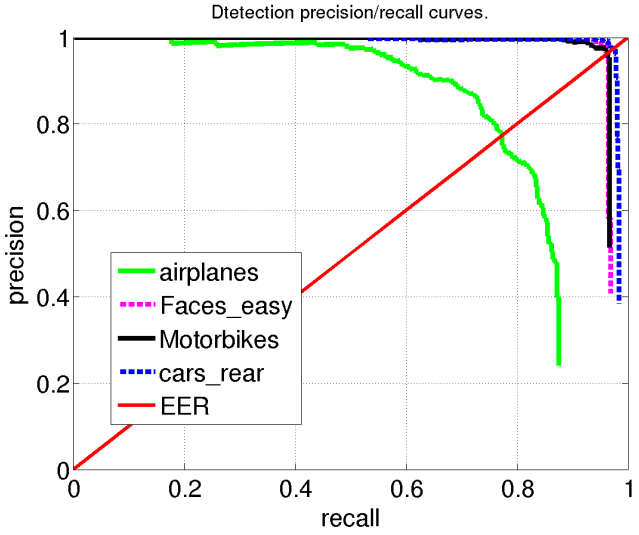


Figure 4.4: Detection precision-recall curves for Caltech-4 classes.

The developed method, utilizing only positive examples in the training phase performs favourably with Caltech-4 compared to other methods (Table 4.1). Moreover, the classification results are based on detection scores that are not optimal for the classification task. Figure 4.4 shows detection precision-recall curves for the Caltech-4 classes. A noticeable difference between classification and detection results for the *airplanes* category is most likely explained by the fact that even though hypothesis for the *airplane* category had higher scores than other categories (*cars rear*, *motorbikes* and *faces easy*) the overlap with the groundtruth bounding box was less than 0.5.

4.5.4 Caltech-101 with Manually Annotated Landmarks

Object Class Detection

In this experiment the developed generative object class detector is compared to the state-of-the-art discriminative deformable part-based model (DPM) by Felzenszwalb et al. [55]. Precision-recall curves are reported for several Caltech-101 categories that show the performance highs and lows of the both methods. The experiment is done with Caltech-101 classes that have minor 2D pose changes and therefore mainly evaluate the method's ability to capture appearance variation.

The Felzenszwalb et al. method was executed in two modes: 1) with a sufficient number of images from the background class as negative examples, exploiting its full discriminative power, and 2) with only a single randomly selected background image (-no-neg). The latter is intended to approximate the setting of positive examples only. The result graphs in Figure 4.5 and Appendix II demonstrate that the proposed generative detector performs comparably to the state-of-the-art method.

In general, the standard DPM method is superior, except for two classes, *yin yang* and

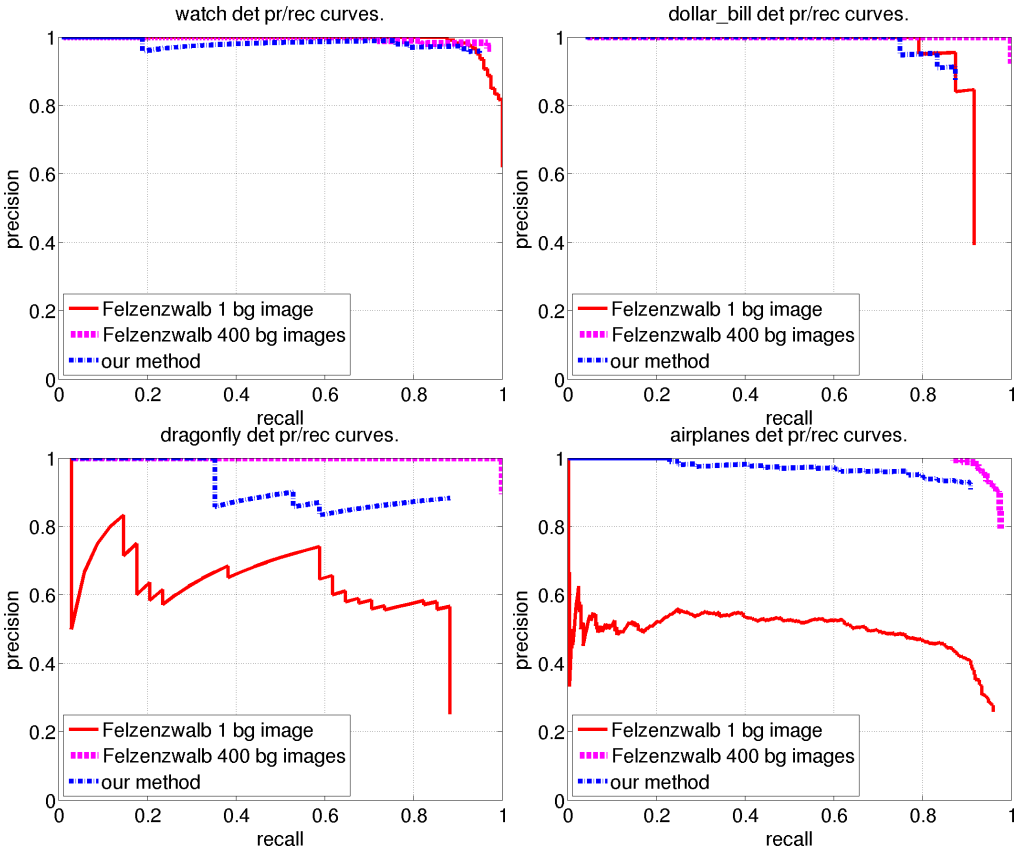


Figure 4.5: Comparison of the generative positive examples only method and a state-of-the-art discriminative method (Felzenszwalb et al. [55]) in an object detection task for Caltech-101 categories: *watch, dollar bill, dragonfly* and *airplanes* (from left-to-right and top-down).

watch, for which the negative set yields worse parts than without negative examples at all. The proposed generative learning performs comparably to the DPM method without negative examples, except for the classes dragonfly and airplanes for which the DPM-no-neg fails to learn the classes properly. Note that both DPM methods utilize the bounding box optimization procedure as post-processing (for more details see section 2.5.3 and [55]). Average precision results are compared in Table 4.2

Table 4.2: Detection results (average precision) for the selected Caltech-101 categories.

	Classes											
												
Our	95,2	95,8	94,1	88,7	95,7	92,3	81,7	93,1	99,6	97,2	86,7	96,9
DPM-no-neg [55]	99,6	89,1	99,7	97,4	60,2	87,4	87,3	92,7	97,0	98,2	51,3	99,8
DPM [55]	100,0	100,0	100,0	99,8	100,0	100,0	90,1	89,7	100,0	90,7	90,7	100,0

Object Classification

In Section 4.5.3 detection based scores (Equation 4.4) were successfully used for the classification task of Caltech-4. Table 4.3 shows that the use of detection based scores for classification becomes unsuccessful when the number of categories is increased. However, numbers on the main diagonal show that the majority of the images were classified correctly. The worst classification results were obtained for the *dollar bill* category (only $41,67\%$ of true positives). The category was often confused with *motorbikes* and *watches*. This behaviour can be explained by the fact that the *dollar bill* is represented with simple and general features, i.e. corners. Moreover, many *watches* and *motorbikes* in the dataset are shown with a dark frame around the edges, thus making the brighter area inside easily confused with a *dollar bill*. The best classification results, over 90% of true positives, were achieved for the three most plentiful Caltech-101 categories (*airplanes, faces easy* and *motorbikes*) and *grand pianos*. However, *grand piano* and *watch* are categories with which most of the other categories are confused. It is noteworthy that none of the images were confused with categories having simple and stable spatial and appearance models, such as *yin yang, stop sign, faces* and *dollar bill*. The confusion matrix entries were calculated with detection scores transformed for classification according to Equation 4.4.

Table 4.3: Classification results (confusion matrix) for the selected Caltech-101 categories.

												
airplanes	91,37	2,79	0,00	1,01	0,00	3,05	0,00	0,25	0,51	0,00	1,01	0,00
car side	0,00	86,15	0,00	0,00	0.00	1,54	0,00	4,61	3,08	0,00	4,62	0,00
dollar bill	0,00	4,17	41,67	0,00	0,00	4,17	0,00	16,67	4,17	0,00	29,17	0,00
dragonfly	0,00	0,00	0,00	61,76	0,00	11,76	0,00	0,00	5,88	0,00	20,59	0,00
faces	0,00	0,00	0,00	0,00	92,54	0,88	0,00	0,00	0,44	0,00	6,14	0,00
piano	0,00	0,00	0,00	0,00	0,00	94,00	0,00	2,00	0,00	0,00	4,00	0,00
menorah	2,27	0,00	0,00	0,00	0,00	13,64	72,73	0,00	0,00	0,00	11,36	0,00
motorbikes	0,24	0,48	0,24	0,00	0,00	0,71	0,00	97,62	0,00	0,00	0,71	0,00
revolver	2,44	4,88	0,00	4,88	0,00	2,44	0,00	2,44	75,61	0,00	7,32	0,00
stop sign	0,00	0,00	0,00	0,00	0,00	5,88	2,94	2,94	0,00	67,65	20,59	0,00
watch	1,65	0,00	0,00	0,00	0,00	12,39	1,65	2,48	2,48	0,00	79,34	0,00
yin yang	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	3,33	0,00	16,67	80,00

Effect of Prior Knowledge

Prior information about allowable transformations for object structure verification (Algorithm 4.1) was used during testing. This not only speeds up the process of hypothesis verification but also filters out hypotheses not consistent with training data properties, e.g. if all people in the training set are vertically oriented the algorithm would not allow a horizontally oriented hypothesis to score highly or be considered at all. The experiment also investigates the effect of reinforcing hypothesis bounding box corners to be inside the image boundaries. The reason for this bounding box post-processing comes

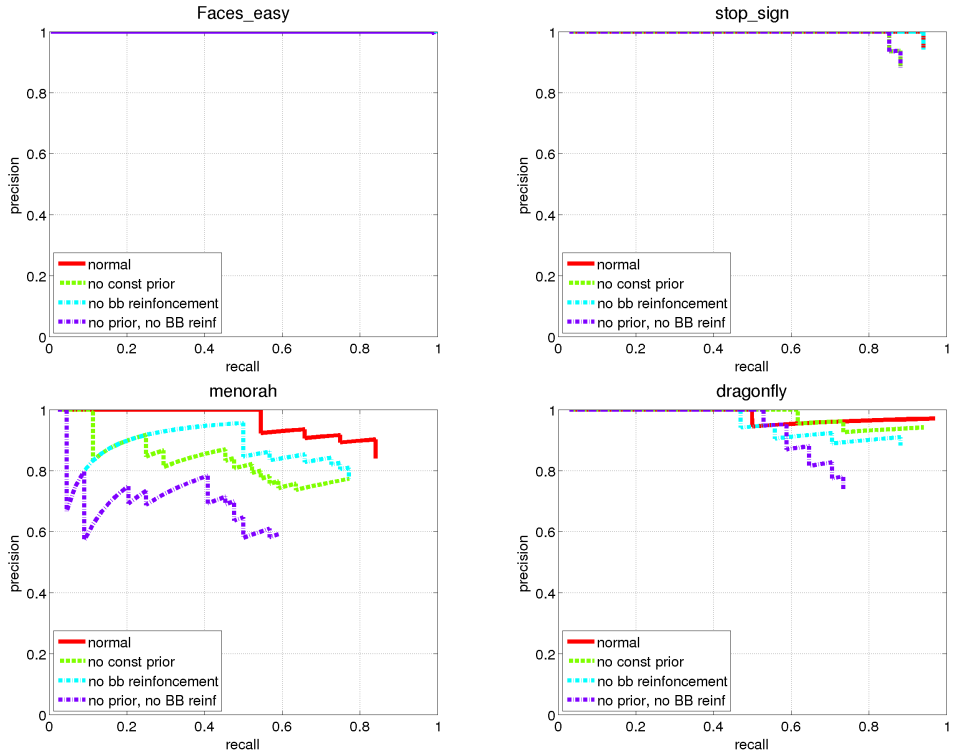


Figure 4.6: Effect of prior knowledge about objects' spatial statistics (orientation and scale, H_{prior}) and reinforcement of all bounding box corner locations to be inside the image boundaries.

from the empirical knowledge that all bounding boxes in all images of all datasets are marked inside the image boundaries even if the objects are truncated. From Figure 4.6 it can be seen that the detection results of well-performing categories (e.g. *faces* and *stop signs*) are almost not affected by the prior knowledge factors, but the use of prior information about object pose and bounding box correction improves the results of the more challenging (big intra-class variation) and scarce categories like *menorah* and *dragonfly*.

4.5.5 Caltech-101 with Automatically Generated Landmarks

Since manual annotation of object parts is very time consuming, it was decided to develop a procedure for automatic object part (landmark) selection. Another reason to renounce the use of manual landmarks is that intuitive manual selections do not guarantee good discriminative qualities of the landmarks from a computational point of view.

As many modern image databases provide object bounding boxes for the training images, the most straightforward and general way for automatic landmark generation is dense $P \times P$ sampling within the object's bounding box [19, 103, 52]. Nevertheless, many of the generated landmarks would not be object specific (e.g. they would appear on

the background) thereby unnecessarily increasing computational workload. To increase the speed of computation and decrease uncertainty in object description a procedure for landmark selection is needed. The selection procedure is described in Algorithm 4.2.

Algorithm 4.2 Automatic landmark selection.

1: Generate dense grid of landmarks *gtLms* for all train images and apply the part detector to them.
2: **for all** images **do**
3: Calculate the average location for the predicted landmarks *predLms*.
4: Find euclidean distances *errd* between the groundtruth landmarks *gtLms* and average locations of the predicted ones *predLms*.
5: **end for**
6: Set the threshold *thld* equal to the weighted mean of all *errd*.
7: Select those categories of landmarks for which the error distance *errd* is lower than the threshold *thld*.

The minimum number of the landmarks allowed to be chosen is 3, as similarity transformations to the mean object space in Algorithm 3.1 require at least 2 landmarks. In general, the number of selected landmarks is class dependent in order to optimally reflect the structural characteristics of each object class. Several object class models formed with automatically selected landmarks with Algorithm 4.2 are shown in Fig. 4.7.

The landmark selection procedure described in Algorithm 4.2 imposes a restriction on the object's pose variation within the bounding box: the more dense sampling used (the higher is P) the less variation in the object is allowed. This restriction ensures correspondence of dense samples to the same parts of the objects in different images.

Despite the restriction mentioned above, automatic landmark selection improves the results of part detection for most of the tested classes; several examples of this improvement, corresponding to parts shown in Figure 4.7, are shown in the Fig. 4.8. For fair comparison no randomization was used here but a full bank of Gabor filters with parameters from 3.11.

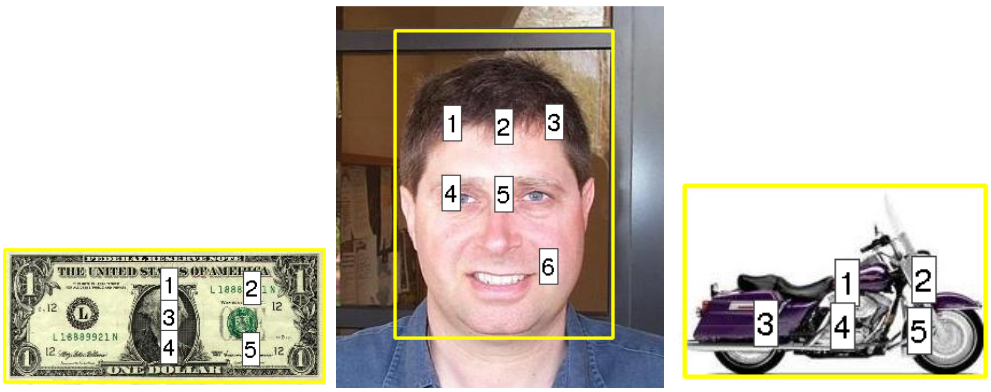
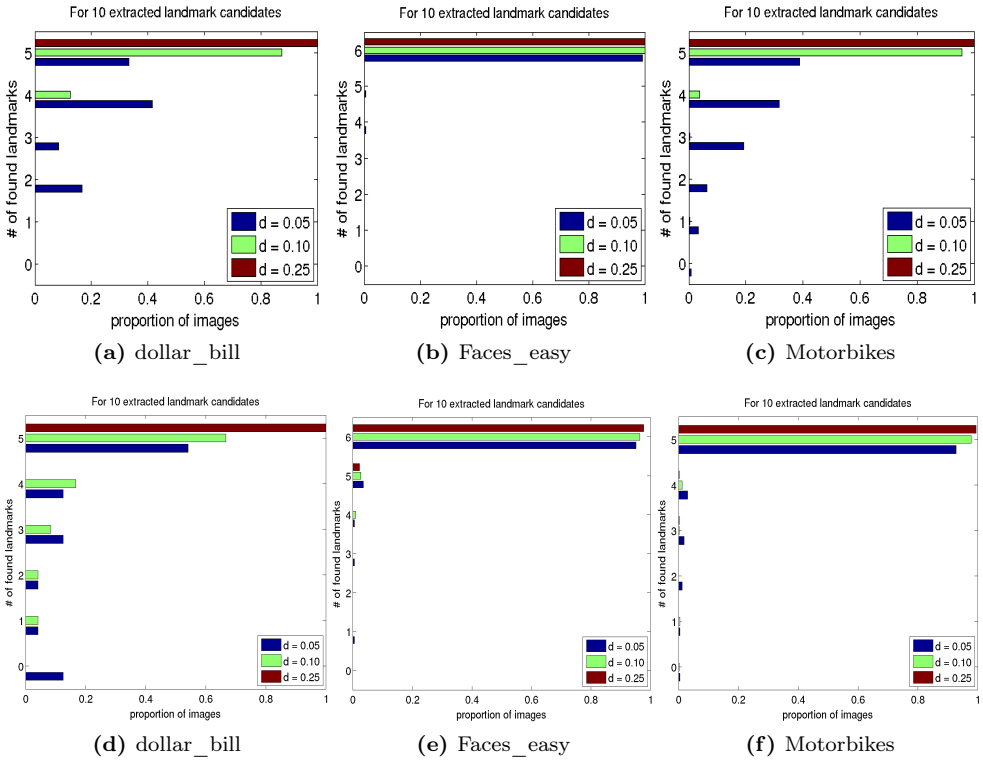


Figure 4.7: Automatically selected landmarks used for the results in Fig. 4.8.



(a) dollar_bill **(b)** Faces_easy **(c)** Motorbikes

(d) dollar_bill **(e)** Faces_easy **(f)** Motorbikes

Figure 4.8: Top: part detection results for Caltech-101 classes (dollar bill, faces easy and motorbikes) using full bank of Gabor filters 3.11 with manually annotated landmarks. Bottom: results using the same Gabor bank with automatically selected landmarks.

Two experiments with Caltech-101 classes exploring the applicability of automatically selected landmarks object class detection are described below. Both experiments are based on dense grid generated landmarks. The first set of landmarks represents object parts, so a grid of 3×3 points is placed inside the bounding box (Figure 4.9 top left). The second set of points corresponds to the object contours, thus evenly spaced points are generated along bounding box edges plus one point in the centre (Figure 4.9 top right). The experiments show that object detection results with points inside bounding box outperform those on its edges, though are still worse than manually annotated semantically meaningful object parts (Figure 3.5 top). It can be seen from Table 4.4 that motorbikes can be described with their contours as well as with manually selected parts. The *yin yang* category has better detection results with the dense grid generated inside the bounding box than with manual landmarks. These results show the possibility of discarding the exhaustive annotation step by substituting it with a dense grid of points after prior image alignment, for example with [101, 188]. However, in general, representation with manually annotated parts outperforms both types of dense grid, leaving the unsupervised automatic landmark selection as an open question for the future.

Table 4.4: Detection average precision for Caltech-101 categories with different landmarks.

	car side	dollar bill	stop sign	revolver	dragonfly	grand piano
manual ann	95,21	95,84	94,12	88,65	95,72	92,29
grid inside BB	55,97	91,67	81,69	80,26	83,17	82,94
grid on BB border	50,13	70,40	47,03	73,20	65,46	67,81
	menorah	yin yang	faces easy	watch	airplanes	motorbikes
manual ann	81,69	93,07	99,56	97,19	86,68	96,92
grid inside BB	78,40	100,00	98,42	82,16	67,69	87,44
grid on BB border	51,74	96,33	97,76	54,68	63,81	95,02

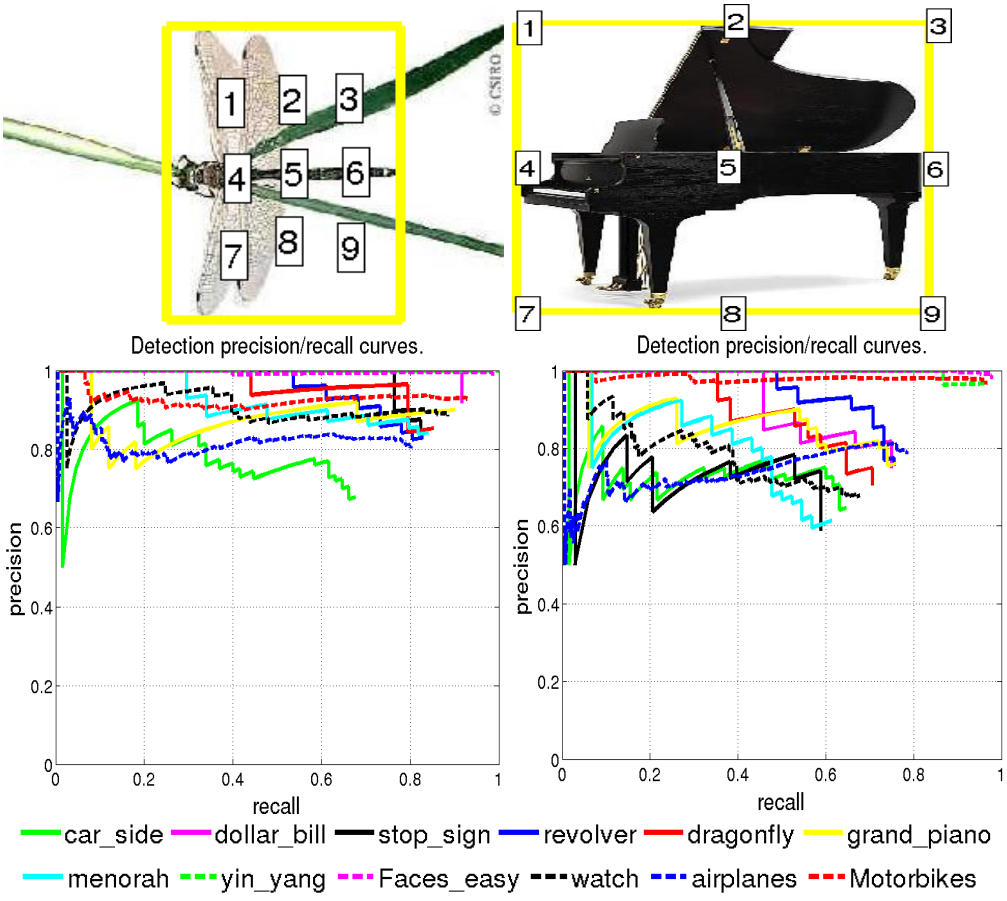


Figure 4.9: Object detection results for Caltech-101 classes using automatically generated landmarks. Top: Automatic landmarks inside the bounding box (left) and on the box contour plus centre (right). Bottom: object detection results, correspondingly.

4.5.6 ImageNet Object Class Detection

Pose and appearance variation in the ImageNet classes is much more complex and represents more realistic test data. For ImageNet classes the novel pose quantization procedure was tested with three pose models ("our-3"). Table 4.5 also shows the detection results using only the most frequent of the three pose models ("our-1"). An experiment with test images in the canonical space ("our-canonic"), where pose variation is removed and no quantization is needed, is also presented in this section. The idea behind the canonical

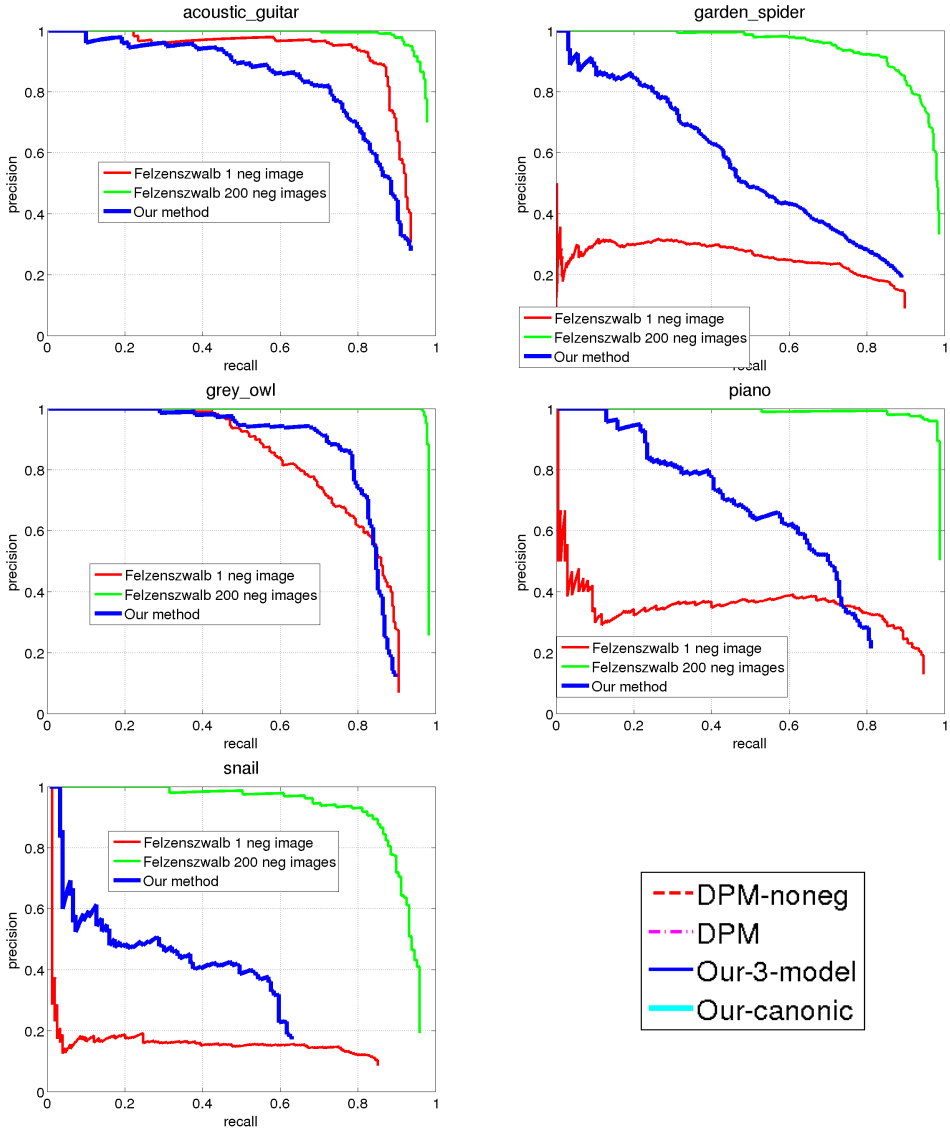


Figure 4.10: Precision-recall curves for the selected ImageNet categories.

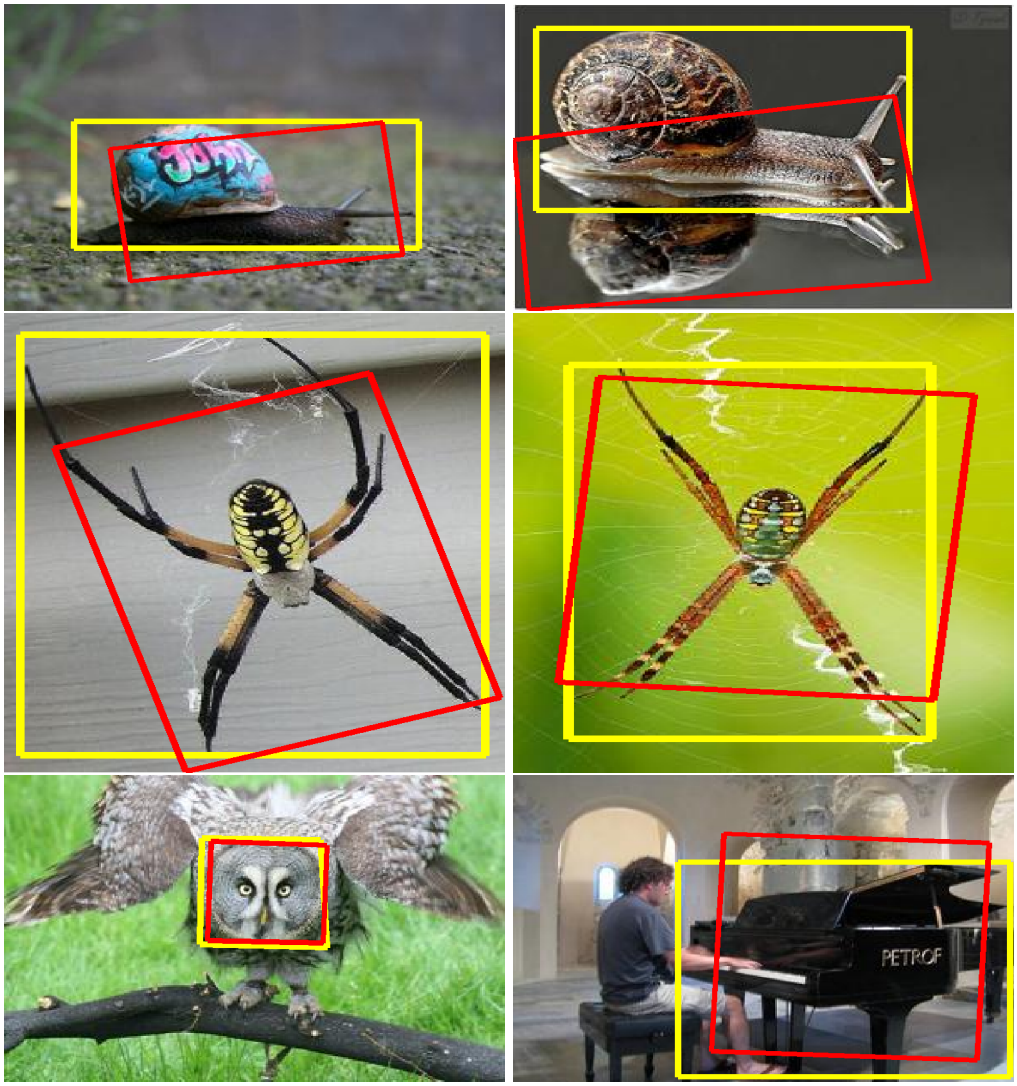


Figure 4.11: Example detections in the images from ImageNet. Note that bounding boxes provided by the developed object detector are not limited to a rectangular shape, but show the object's pose, hence sometimes they are not counted as correct (with overlap ≥ 0.5). Yellow boxes - groundtruth, red boxes - obtained detections.

space experiment was to study the contribution of appearance without pose variation. These experiments verified the previous results with the Caltech images: DPM without negatives often fails to learn the proper object class model, while in other cases the DPM-no-neg and the method presented in this work provide comparable accuracy (Table 4.5). However, the developed object detector does not suffer from problems with the negative

examples since they are not used. The biggest problem with the proposed detector occurs with the *snails* class where there is a significant gap between the performance in the canonical space and the original test images. This difference can be explained by the presence of multiple sub-classes, i.e. different types of snails, and 3D pose changes, which cannot be modelled properly by the 2D quantization procedure (Section 4.1). It was also found that the object parts that were manually selected have a dramatic effect on object detection and parts intuitive to humans are not necessarily easy to detect with local part detectors. The full precision-recall curves are shown in Figure 4.10. Several example detections in the ImageNet images are presented in Figure 4.11.

Table 4.5: Detection results (average precision) for the selected ImageNet categories.

	ImageNet categories				
	grey owl	acoustic guitar	garden spider	piano	snail
our-canonic	96,3	88,4	71,5	66,8	58,2
our-1 (best of 3)	42,9	60,4	60,9	53,2	29,8
our-3-model	81,7	80,2	52,5	59,5	31,4
DPM-no-neg	76,4	86,2	24,6	39,0	20,8
DPM	90,9	90,7	88,0	90,5	86,8

4.5.7 Making the DPM [55] Fail

In the previous experiments, the developed object detector performed comparably or even superior to the DPM method trained with a single negative example among the positive examples. However, the standard DPM was clearly superior to both leaving little space to the proposed detector or the DPM with a single (or only a few) negative examples. This work nevertheless postulates that detection is essentially a generative machine learning problem and other classes should not affect selection of the best parts to detect an object class. Even if the best parts are easily confused with parts of another class, the pruning of false positives should happen in the stages following detection. Some evidence that this problem could occur with DPM appeared in the Caltech experiment, where the no-negative version of the DPM outperformed the standard DPM with the two classes, *yin yang* and *watches*. Further investigation of this finding was conducted by introducing "hard" negative examples, i.e. images from a visually similar class (violin vs. cello, etc.) into the training images. For all tested classes there was a clear drop in the results, but there was also a striking finding that sometimes even a small number of hard negatives, down to 1%, can make the strong discriminative latent support vector machine learning of DPM fail. For example, Figure 4.12 shows the DPM results for the *garden spider* class trained with 200 random negative examples and the same 200 negative examples with two hard examples of the *black widow* spider. Presence of hard negatives caused the accuracy to collapse from $AP = 88.0$ to $AP = 14.4$. This result further justifies research on the generative approach to object detection or hybrids adopting both generative and discriminative principles of learning.

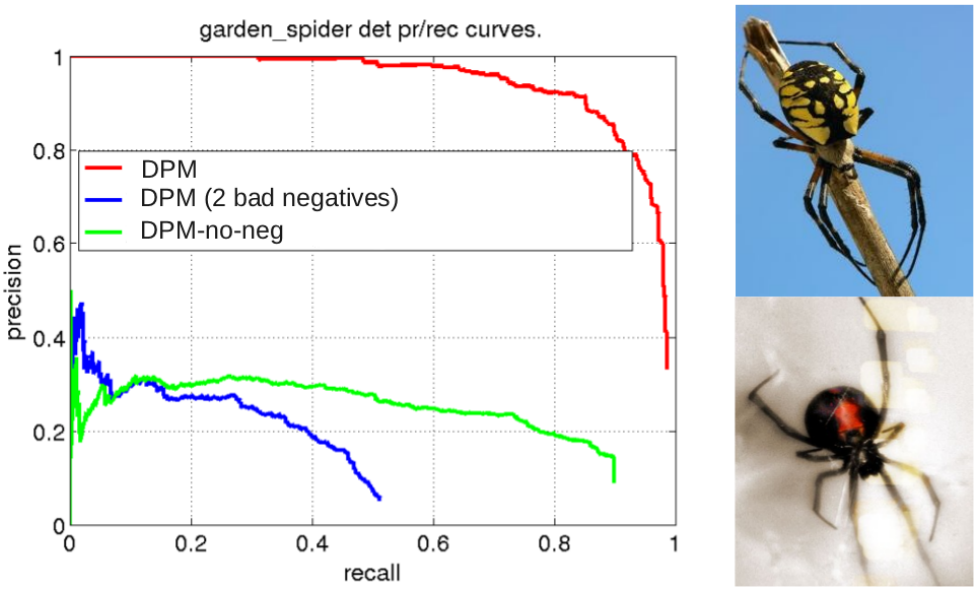


Figure 4.12: The DPM method by Felzenszwalb et al. [55] fails to learn an object detector for *garden spiders* (top right) if two examples from a similar class, *black widow* spider (bottom right), are introduced into the training set of 200 negative examples.

4.6 Summary

In this chapter, a generative part-based object class detector was described and its performance on several Caltech and ImageNet categories evaluated. An interesting property of various databases, object pose quantization, was also investigated in the beginning of the chapter. During training, the detector aligns the images in order to learn their appearance without geometric distortions (see Section 3.2) simultaneously revealing the spatial structure of the objects. With pose clustering, separate models can be learned by aligning training images belonging to a cluster, i.e. using just part of the training images, or aligning all training images, forcing cluster centres to act as seeds. The resulting object's spatial structures, i.e. the constellation models, are described with a mixture of 2D Gaussians. During object hypothesis retrieval the constellation and appearance scores are combined in Algorithm 4.1, which is robust to occlusions and miss-detections. Experiments showed that the proposed generative object detector is capable of good object representation when only small 3D pose variation is present (Caltech-101 results). The detector's performance drops when 3D pose changes are introduced (ImageNet results). In most cases, the developed generative object detector performs as well as a state-of-the-art discriminative detector in generative mode, significantly outperforming it for a few categories, e.g. *airplanes* (Figure 4.5). However, the discriminative detector with full discriminative power stably gave better results than other two approaches. An attempt to solve the problem of unsupervised object part selection was also made in this chapter. The experiments showed that object detection with dense grid landmarks

generated within a bounding box of aligned images provides results comparable to those with manually annotated landmarks, but the problem of automatic landmark selection remains an open question.

Despite one of suggestions made in this work that object detection and classification should be separated, as detection is a maximum likelihood task while classification is a Bayesian one, an attempt to use detection scores for classification was made for Caltech-4 and Caltech-101 categories. It is evident from the results (Caltech-101 classification) that the detection score does not have enough discriminative power to perform well in classification for a large number of categories. Moreover, features/object parts that are good for detection can perform poorly for classification.

The generative nature of the developed object detector allows learning from positive examples only, but at the same time the lack of discriminative power causes an excessive amount of false positive detections. Another reason for the large number of false positives is that the appearance score used in this work is essentially a likelihood not a probability, which makes the resulting detection scores not readily comparable between the images. In other words, the developed object detector tries to find an object from everywhere, hence it has high recall but low precision. This problem can be solved by adding a discriminative classifier after the generative detector. Thus the next chapter presents generative-discriminative hybrid combinations applied to the object detection task.

Chapter V

Advanced Processing for Object Detection

The previous chapter described a generative part-based object detector. Even though the detector achieves high levels of recall with challenging object classes, its average precision level is relatively low due to a large number of false positive detections. This problem arises from the way the task for object detection is formulated: find a place/places in the image that most likely contain an object of a certain class. Thus, the system has a predisposition to produce a lot of false positive detections and "see" objects even if they are not present. In this chapter, a 2-stage generative-discriminative hybrid is proposed to overcome the problem of excessive false positive detections. In the hybrid method, candidate detections of the generative method are re-scored by a discriminative stage, pruning false positives.

Another possible extension to the object detector is related to the use of color. It has been reported that use of raw color brings only several per-cent improvement to the performance of current systems [29]. However, proper color normalization could possibly increase the impact of color on the detectors' performance. This chapter also describes a color normalization technique in which true colors are not important per se but examples of the same class have photometrically consistent appearance. The color normalization is achieved by supervised estimation of a class specific canonical color space where the examples have minimal variation in their colors.

5.1 Hybrid Generative-Discriminative Method

Hybrid generative-discriminative methods are widely used in different applications of computer vision such as scene classification [20], tracking [110] and image classification [108]. Hybrid methods for visual object recognition can be divided into two categories: feature encoding based [108, 133] and learning based [64, 94, 102, 194] approaches. The framework in [64] shares a similar structure to the proposed algorithm, but in the developed framework the generative and discriminative stages are based on different and more generic features (Gabors and HOGs), while in [64] the same codebook representation is used by both stages of the hybrid system. Another mechanism similar to the proposed

method was introduced in Regions with Convolutional Neural Networks (RCNN) in [69], where a general objectness detector from [4] generates a large number of bounding box candidates and then a discriminative classifier is applied to obtain true positives in the images. Different from the RCNN method, the generative object detector in this framework provides control over the number of proposals generated, which can vary from max 1 per class to max N per class. Thus the following discriminative detector in the proposed framework can focus on coping with the variance between the background and object classes instead of both inter-class and intra-class variations as in the RCNN framework.

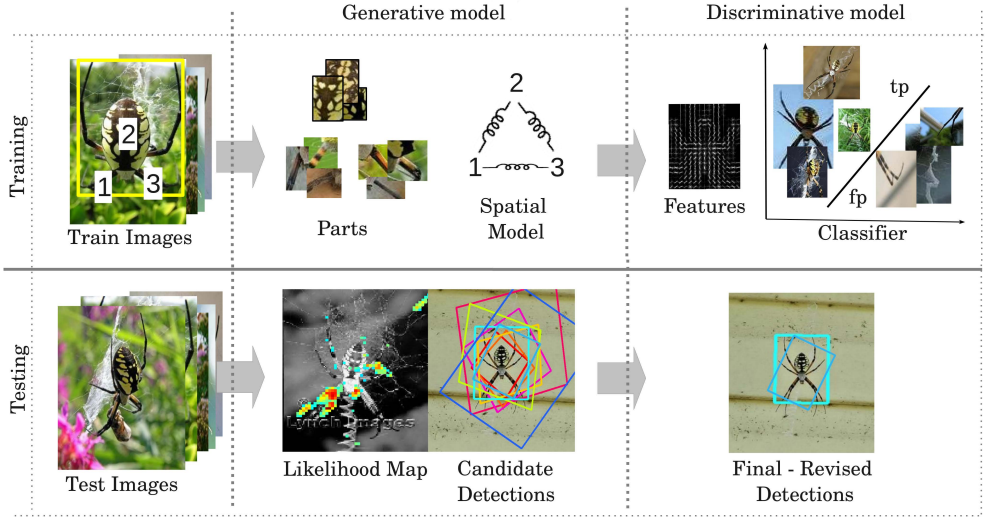


Figure 5.1: Workflow of the developed hybrid generative-discriminative method. In the generative detector only positive instances of each object class with annotated bounding boxes and semantic object parts are employed. The true positive and false positive detections, obtained from the generative model with training images, are used as positive and negative input examples for the discriminative object detector, which learns to discover their dissimilarities. During testing, candidate object detections of the generative method are re-scored with the discriminative method, leading to a reduction in false positives and increasing precision. Here *tp* denotes true positives and *fp* - false positives.

In this chapter, a hybrid 2-stage method for object detection is proposed. The method exploits the complementary properties of generative and discriminative approaches. Generative models capture the appearance distribution of a class and produce compact intra-class variance, while discriminative models learn the decision boundary between correct and false positive detections, producing large inter-class variance. By separating these stages, unlike in existing monolithic systems, a hybrid generative-discriminative model for visual class detection (Figure 5.1) can be established. The proposed framework can be viewed as a coarse-to-fine cascade, i.e., first localize the candidate locations with the generative object detector and then find true objects among those candidates with the discriminative classifier. In the experiments, the hybrid method was composed of the fully probabilistic Generative Object Detector (GOD) (presented in Chapters 3 and 4) and various state-of-the-art discriminative methods (deformable part-based model (DPM) [55],

histogram of oriented gradients (HOG) [175] or deep features (DF) [154] combined with the support vector machine (SVM) [28] or random forest (RF) [88] classifiers).

As shown in Figure 5.1, the pipeline of the proposed framework can be divided into a generative part (see Chapter 3 and 4) and a discriminative part (see Section 5.1.1). Section 5.1.2 presents details of the generative-discriminative hybrid formulation.

5.1.1 Discriminative Learning

Discriminative learning efficiently establishes mapping between input and output parameters (e.g. between features and class labels). This section presents two popular discriminative classifiers: the Support Vector Machine (based on the generalized portrait algorithm [172, 173]) and the random forest [24].

Support Vector Machines (SVM)

For easier understanding, a support vector machine applied to a 2D linearly separable classification problem (Figure 5.2) is described in this section. However, SVMs can handle linearly non-separable cases by using kernel functions to transform features into higher dimensional spaces where classes are separable. Ultimately good separation is achieved with a hyperplane that has the largest margin to the nearest instances of the different classes, i.e. maximum margin classification. The larger the margin the better the classifier is expected to perform on unseen data.

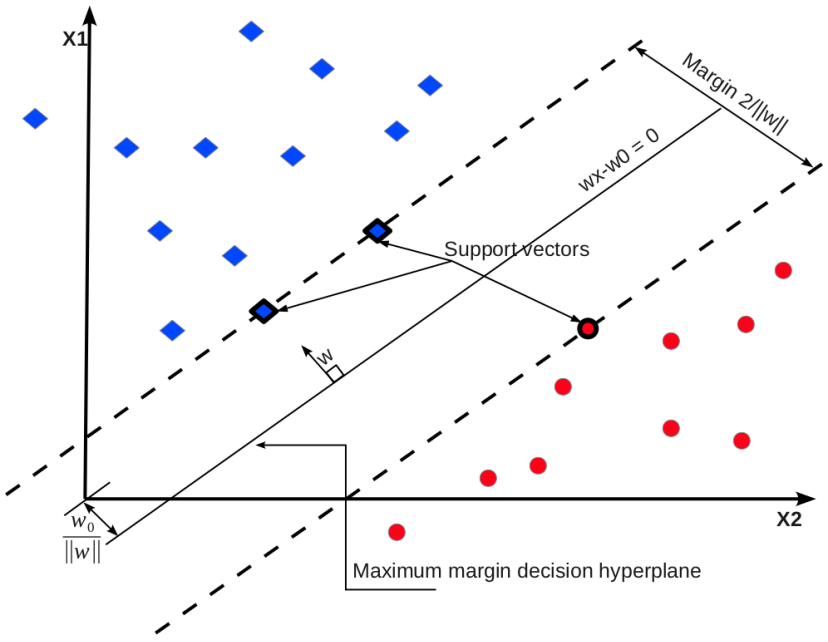


Figure 5.2: Linear SVM maximum margin separation.

Any hyperplane can be defined by its perpendicular $\boldsymbol{w}$ and offset from the origin $\boldsymbol{w_0}/\|\boldsymbol{w}\|$, thus the equation of a hyperplane is $\boldsymbol{w} \cdot \boldsymbol{x} - \boldsymbol{w_0} = 0$, where $\cdot$ is a dot product. In this example $\boldsymbol{x_i} \in \mathbb{R}^2$. Given the points $\boldsymbol{x_i}$, $i = 1 \dots N$ and the corresponding class labels $y_i = \{-1, 1\}$, SVM finds such a hyperplane that samples from different classes are on the different sides of the hyperplane. The hyperplane also provides the largest margin between the points of class 1 and -1. Points closest to the hyperplane, defining the margin, are called support vectors. The corresponding optimization problem is defined as follows: $max \frac{2}{\|\boldsymbol{w}\|}$, subject to $y_i(\boldsymbol{w} \cdot \boldsymbol{x} - \boldsymbol{w_0}) \geq 1$.

There exist a variety of SVMs, such as latent-SVM used in [55], Least Squares Support Vector Machines [161] or SVM for multiple-instance learning defined in [9]. In the experiments SVM implementation from the libsvm library [28] was used.

Random Forests

To understand the concept of random forests it is better to start with a single tree example. As in the case of SVM the input data is multidimensional features $\boldsymbol{x_i} \in \mathbb{R}^p$, $i = 1 \dots L$ and the corresponding class labels $y_i = 1 \dots C$. Each node in a tree corresponds to a split of a feature into two regions, e.g., splitting of feature j with a splitting point s results in $R_1(i, s) = x|x_i > s, R_2(i, s) = x|x_i < s$. Both of these regions are also split into several regions and this process is repeated on all of the resulting regions until some stopping rule is applied. The splitting feature and splitting point are defined so that error between predicted and ground truth classes is minimized.

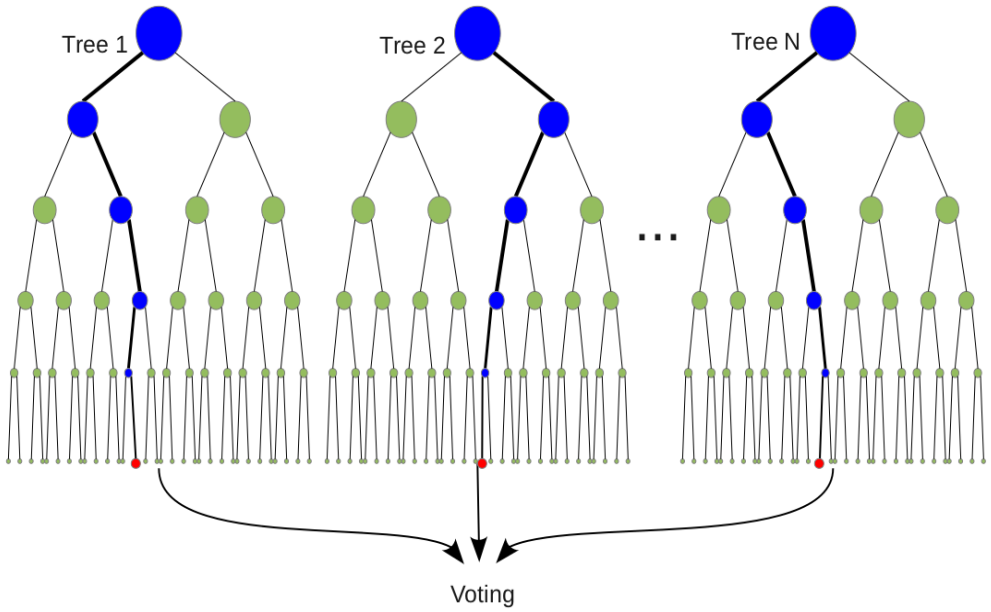


Figure 5.3: Visualization of a random forest. Darker blue circles with thicker branches highlight the decision path, which ends with red circle - the terminal node that gives the class label.

One of the major problems of single tree classifiers is their high variance, i.e. a small change in the data may result in a very different series of splits. Random forest, a forest composed of de-correlated trees, solves the problem of high variance. Random forest (Figure 5.3) is composed of a big number of trees each of which is trained on a random subset of training examples and each split is based on a random subset of features. The experiments used a Google implementation of random forests [88].

5.1.2 Generative-Discriminative Hybrid

The limitations of using exclusively either a generative or discriminative approach motivate the development of a hybrid detectors. The 2-stage pipeline where a discriminative classifier follows a generative object detector in order to improve the detection performance is inspired by state-of-the-art hybrid methods. In the developed hybrid method the discriminative classifier can be viewed as a post-processing stage of the generative object detector. Figure 5.5 illustrates the pipeline of the proposed hybrid method.

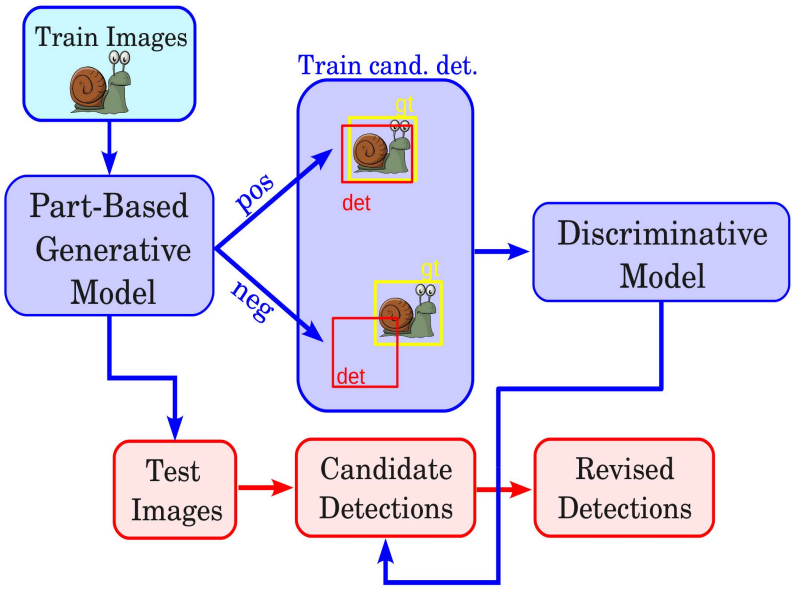


Figure 5.4: Generative-discriminative hybrid method. Blue parts correspond to the training stage, red parts to testing.

The generative method is trained by positive examples of the query image category with annotated object parts and bounding boxes. The discriminative method, on the other hand, uses the training stage outputs of the generative method as its inputs. Candidate detections of the generative method (training data) are transformed to the aligned space using the detected part locations. After alignment, detections are scaled to the size 64×64 in pixels and subsequently fed to the discriminative part for re-scoring. Generative output candidates having bounding box overlap ratio $A > 0.7$ with the ground truth are used as positive examples in discriminative training and outputs with $A < 0.2$

as negatives. This representation of positive and negative data allows the discriminative method to learn, exploit and emphasize the difference in appearance of the true positives and false positives that the generative part produces but is blind to. The re-scored detections of the discriminative stage are further processed by non-maximum suppression, which removes very similar and overlapping candidates. Non-maximum suppression in the experiments removes candidates with lower scores if their overlap ratio is greater than 0.5 (i.e. $BB_{hyp1} \cap BB_{hyp2}/BB_{hyp1} \cup BB_{hyp2} \geq 0.5$). The non-maximum suppression procedure is applied to all hybrid pipelines studied (GOD+DPM, G-DPM+DPM, GOD+HOG+SVM, GOD+HOG+RF, GOD+DF+SVM and GOD+DF+RF).

5.1.3 Experiments

Settings

Five challenging categories from the ImageNet database were used in the experiments: *acoustic guitar, piano, snail, garden spider* and *grey owl*. The images contain objects appearing in different scales, orientations, lighting conditions, with limited 3D pose changes and moderate intra-class variation. The images for each class were randomly divided into training and testing groups of approximately the same size. In these experiments the test set was composed of the test images from all categories selected from ImageNet, same way as in the classification task (e.g. subsection 4.5.3). Therefore result curves in this section demonstrate methods ability to detect and classify objects.

Performance Metrics

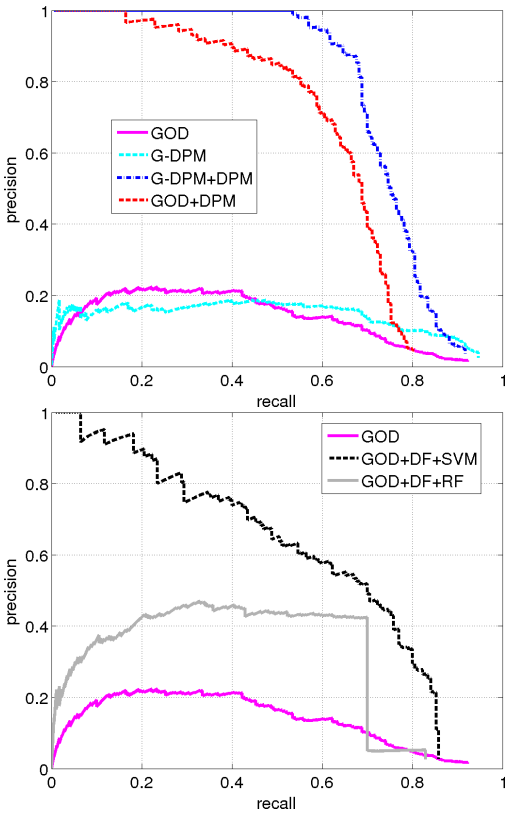
The detection hypothesis is considered correct if the overlap ratio A (see Section 4.5.2) is greater than 0.5 and detection is not duplicate, as in the experiments in Chapter 4. The general performance of the investigated methods is described with the precision-recall curves used in major computer vision competitions (e.g., PascalVOC [48] and ImageNet [148]). Precision and recall are defined through the concept of true positives, *tp*, and false positives, *fp*, where *tp* is the proportion of instances correctly labelled as positive, while *fp* is the number of negative examples incorrectly labelled as positive. Precision and recall are computed in the following way:

$$Precision = \frac{tp}{tp + fp}, \quad Recall = \frac{tp}{\text{Total number of positives}}.$$

In general, a generative method can produce a large number of hypotheses to guarantee that at least one passes the test in (4.5.2). This would result in high recall, but poor precision which, is the problem of generative methods. The discriminative part of the proposed pipeline aims to reduce the number of *false positives* (fp) by keeping the number of *true positives* (tp) high.

Results

The results of the various implementations of the proposed hybrid method are shown in Table 5.1. The implementations are based on publicly available code: the proposed



Acronym	*Description*
HOG	histogram of oriented gradients features [175]
DF	deep features produced by deep convolutional neural network [154]
GOD	generative Gabor part detector [141] and canonical space constellation model [140]
DPM	HOG feature based deformable part model [55]
G-DPM	"almost" generative version of DPM using only a single negative example.
RF	discriminative random forest classifier [88]
SVM	discriminative support vector machine classifier [28]

Figure 5.5: Precision-recall curves for the Imagenet category *piano*.

Table 5.1: Detection results (average precision and maximum recall) for the selected ImageNet categories.

	grey owl AP/max(rec)	acoustic guitar AP/max(rec)	garden spider AP/max(rec)	piano AP/max(rec)	snail AP/max(rec)
GOD	71,1/89,8	37,6/96,2	34,5/93,9	14,5/92,4	12,8/69,1
G-DPM	73,1/90,6	76,5/93,3	30,3/85,2	14,5/94,7	42,9/81,9
G-DPM+DPM	89,9/90,2	79,2/90,8	69,4/80,2	75,2/91,8	50,4/77,2
GOD+DPM	89,2/89,3	87,6/91,2	73,0/81,6	63,6/80,6	55,2/69,1
GOD+HOG+SVM	89,5/89,7	84,5/92,0	66,1/79,5	54,2/78,8	48,1/65,8
GOD+HOG+RF	84,5/89,3	75,5/88,7	64,0/80,9	49,5/74,1	41,2/61,7
GOD+DF+SVM	89,4/89,8	83,4/89,5	67,8/80,5	59,6/85,9	40,3/64,4
GOD+DF+RF	79,3/88,9	49,2/86,1	48,4/79,1	32,6/82,9	21,5/59,1

Gabor object detector (GOD) [140, 141], histogram of oriented gradients (HOG) [175], deep features (DF) produced by a deep convolutional neural network [154], and a state-of-the-art discriminative part-based model (DPM) by Felzenszwalb et. al. [55]. In addition to the standard DPM a generative version (G-DPM) was constructed by allowing only a single negative example. From the results in Figure 5.5, Appendix III and Table 5.1, it is obvious that the plain generative methods (GOD, G-DPM) achieve high recall but poor precision; they detect the correct class but are also triggered by many other things. The tested hybrid generative-discriminative methods (GOD+DPM, G-DPM+DPM, GOD+HOG+SVM, GOD+HOG+RF, GOD+DF+SVM, GOD+DF+RF) achieve almost the same recall as the generative methods, but with significantly better precision. The two strongest combinations are GOD+DPM and G-DPM+DPM, indicating the superiority of part-based methods over model-free ones.

5.2 Supervised Class Color Normalization

Color is an important cue in many applications of computer vision and image processing [29, 68]. Colors are determined by the intrinsic properties of the objects and surfaces as well as the color of the light source. For robustness, the effect of the light source must be filtered out, known as *color normalization* or *color constancy*. Previously proposed methods for color constancy aim to estimate the color of the light source, and then transform the image colors to the "world color space" (white illuminant) [67, 61]. In the world color space, the photometric variations are only due to the natural variation of the scene objects. A survey of existing methods can be found from Gijsenij et al. [68]. The human visual system is remarkably robust to change of colors and even to completely abnormal colors [81], but it is unclear how color normalization or constancy should be performed for computer vision tasks.

In this chapter, a novel computational approach for color normalization is proposed. The proposed object class specific color normalization produces colors that minimize the difference within a provided set of images instead of giving physically correct colors. This approach is supervised utilizing common object landmarks. Using the landmarks, an optimal "canonical object color space" is estimated and all the examples transformed to the space. New examples can be transformed to the space by detecting the same landmarks.

5.2.1 Estimation of Canonical Object Color Space

The goal of the proposed color normalization is to construct a class specific "canonical object color space" where color variance of the transferred objects is minimized. This canonical space is based on alignment of class specific object colors in a 3D RGB color space. Class specific colors are defined by manually annotated landmarks (see Figure 5.6) corresponding to object regions, whose colors are expected to be photometrically (chroma and brightness) consistent. The minimum required number of annotated landmarks is 3 for 3D similarity transformation.

The canonical object color space estimation algorithm is outlined in Algorithm 5.1. The algorithm is based on the same principle as spatial alignment Algorithm 3.1 (Chapter 3).

Figure 5.6: Caltech-101 examples with annotated landmarks (denoted by the green circles and numbers). Good landmarks are: petals of a flower, green leaves, skin and fur patches.

Geometric transformation is estimated using the Umeyama method [170]. The estimation procedure is further illustrated in Fig. 5.7.

Algorithm 5.1 Canonical color space.

1: Select a random seed image r and use N landmark colors $\{\boldsymbol{col}_{r,1}, \ldots, \boldsymbol{col}_{r,N}\}$ as the initial color space S.
2: **for all** images i **do**
3: Estimate the geometric transformation ${}^{S}\boldsymbol{T}_i$ (3D similarity) from the colors of the ith image landmarks to the current color space S.
4: Transform the ith landmarks $\{\boldsymbol{col}_{i,1}, \ldots, \boldsymbol{col}_{i,N}\}$ to the current color space using ${}^{S}\boldsymbol{T}_i$.
5: Refine the current canonical color space by taking the average of the all transformed landmarks $S \leftarrow avg(\{\boldsymbol{col}_n\}_i)$.
6: **end for**
7: Return S as the canonical color space.

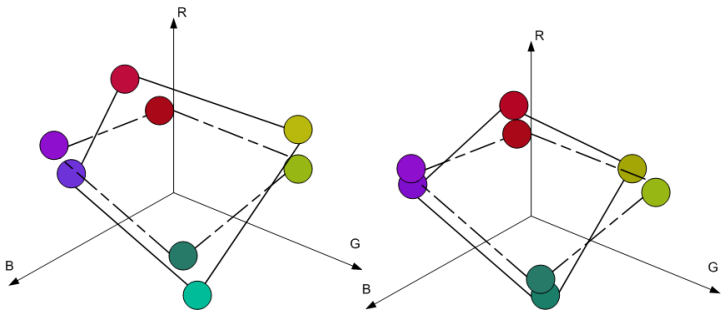


Figure 5.7: Example of the Umeyama estimated similarity; original (left) and the transformed (right). Note that the transform is not exact for four points.

Seed Selection

One issue that affects the final result of Algorithm 3.1 is seed selection. It is noteworthy that the seed does not particularly affect color variance but the mean, i.e. average colors of each landmark (Figure 5.8). Therefore, for the computational methods the result is seed-independent but for a human viewer it can be undesired that colors change after each run of the algorithm due to a random seed. However, there is a simple procedure to

Figure 5.8: Original (left) and color normalized images using three different random seeds shown in bottom right corners. Note that in all the images the skin colors look natural but biased toward the skin color of a person in each seed image.

fix seed selection: compute the mean colors of each landmark and then select the image whose landmark colors are closest to the mean values. This simple procedure is adopted in the experiments.

5.2.2 Experiments

In the following quantitative and qualitative results for class specific color normalization are reported. Experiments are conducted on the popular classification dataset Caltech-101 [51]. Additionally, the success of the canonical object color space is confirmed with a real application where the proposed method is used to photometrically normalize images prior to object pose estimation for robot grasping (see [25] for more details).

Caltech-101

For testing, the following Caltech-101 classes were selected: *Garfield*, *water lily*, *strawberry*, *sunflower*, *panda* and *faces* containing 11-28 images. Each class is represented by 3-4 manually annotated landmarks. Examples of original and processed images are shown in Figure 2.2, Figure 5.9 and Appendix IV.

The quantitative performance metric used in the experiments is the proportional changes in landmark color variances, provided that the mean color values are left almost unchanged during color normalization procedure:

$$\frac{var(\boldsymbol{c}_{orig}) - var(\boldsymbol{c}_{canonical})}{var(\boldsymbol{c}_{orig})} \quad . \tag{5.1}$$

Figure 5.9: *Sunflowers* (originals on the left, images after color normalization on the right).

The computed performance values are given in Table 5.2. The variance reduction for all classes was between 0.28-0.68, indicating significant improvement in color similarity (see also Figure 5.10 for illustration).

Table 5.2: Relative variances of the landmark colors after color normalization.

Cat.	Relative variance				
	LM-1	LM-2	LM-3	LM-4	Avg.
Faces	0.61	0.62	0.15	0.56	0.55
Water lily	0.63	0.57	0.60	0.14	0.55
Garfield	1.08	0.45	0.40	0.50	0.32
Sunflower	0.34	0.47	0.59	0.13	0.44
Panda	0.78	0.42	0.54		0,61
Strawberry	0.72	0.65	0.76		0.72

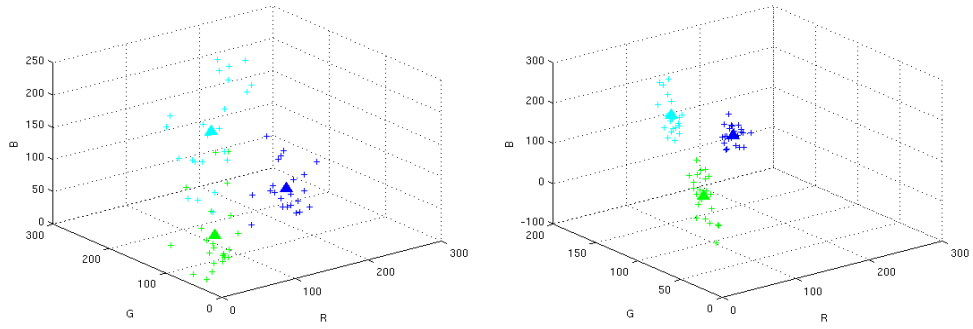


Figure 5.10: All *sunflower* landmark colors as points in the RGB space: original (left) and processed (right).

Color Feature Based Pose Estimation for Robot Grasping

This experiment demonstrates the use of the developed color normalization approach in a practical vision application in a robotic grasping work cell, which requires accurate pose estimation of objects. A Kinect sensor was used as the visual scene input, and the pose estimation system recently proposed in [25] was applied. The task was to find the pose of a real object in a captured scene using the KIT 3D model database [95]. The KIT database contains colorful richly textured objects for which color is an important cue for relating model points to corresponding scene points.

Figure 5.11: An input Kinect scene (top left), two textured KIT object models (top middle), color normalized KIT models (top right) and pose estimation results (color features projected to the scene) (bottom).

The very different illumination conditions between the experiment setup and the setup used for capturing the model textures had significant negative impact on the calculation of the color correspondences between the model textures and the observed scene data,

making the pose estimation to fail. To overcome this problem, a small set of landmarks was used between the textured models and frontal views of the objects in the setup for estimating the color transformation. After processing the models, pose estimation was successfully carried out with a great degree of robustness and accuracy. More details can be found in the paper where the pose estimation method and full results are published [25]. An example scenario for two objects is given in Figure 5.11.

5.3 Summary

This chapter presented two possible extensions of a developed generative part-based object detector. The first part of the chapter was devoted to a generative-discriminative hybrid object detector, where the discriminative part is used for re-scoring detections of the preceding generative detector and thus pruning undesired false positive detections. Experiments showed that all combinations of generative-discriminative detectors performed better than pure generative methods, supporting the author's contention that detection and classification tasks should be separated.

The second part of the chapter investigated supervised object class color normalization. Even though traditionally color is considered an important cue in object detection or classification tasks, in reality color is often not consistent within a visual class, especially with man-made objects. The good performance of the proposed color normalization scheme for the popular classification dataset Caltech-101 and, more importantly, in a practical application of pose estimation for robot grasping suggests that the use of color cues should be studied further in future work.

Chapter VI

Conclusions and Future Work

This work was devoted to the development of a generative part-based object detector based on Gabor features and learning from positive examples only. The proposed visual class object detector originated from a face detector described in [85], however a number of major contributions have been made. The biggest contribution of this work is undoubtedly the introduction of a randomized Gaussian mixture model enabling learning from tens of training images in contrast to the hundreds required by a regular GMM. The randomized GMM also improves object part representation, keeping only descriptive filters in a Gabor filter bank. Another contribution is raising awareness of the importance of learning in the aligned object space to avoid learning of spatial distortions along with the objects appearance. The aligned object space also provided important statistics about object positions in the training images, used as prior knowledge to prune hypotheses with objects in odd poses (e.g. face or a car upside down). A spatial model (constellation score) robust to occlusions and parts misdetections was developed. A property of object classification datasets, object pose quantization, was investigated, leading to learning of object pose clusters instead of searching over all possible combinations of object scales and rotations. Finally, the generative Gabor object detector was combined with discriminative classifiers, which allowed the number of false positive detections of the generative object detector to be decreased, especially for images in which the searched object is not present. Moreover, the proposed method is generic, so its parts can be replaced by other methods, for example, the Gabor features and Gaussian mixture model in the part detector can be changed to a SIFT descriptor and SVM classifier.

This work suggested to separate detection, based on likelihood values, and classification, a Bayesian problem. Figure 6.1 illustrates difference in detection and classification approaches in object part selection. The picture shows two classes of objects: class one defined by corners and a rectangle in the middle and class 2 defined by corners and an ellipse in the middle. The corners are very clearly visible, but the figures in the middle of the objects are not. The best features for detection are object corners as they would allow reliable object detection. However, using corner features it is impossible to differentiate between the objects, i.e. classify them. The features most suitable for classification

are the figures in the middle of the objects, but due to their poor visibility they would give much worse results for detection than corners. This simple example shows the need to separate the detection and classification tasks in such a way that detection precedes classification. Selection of too discriminative parts unsuitable for detection can explain the DPM failure in Section 4.5.7 (Figure 4.12) when two similar classes were used as positive and negative examples.

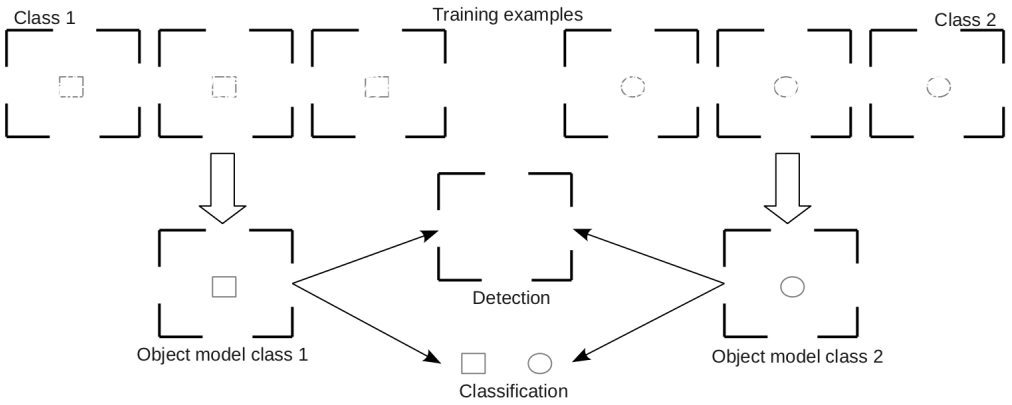


Figure 6.1: Illustration of good features for detection and classification.

Experiments done in this work show that the developed part and object detectors have a performance comparable to state-of-the-art methods; however, there is room for improvement. For example, it was demonstrated that combining the proposed generative object detector with discriminative classifiers significantly reduces the number of false positive detections, improving the average precision of the method. Extensive experiments on adding color information have not been conducted yet, but preliminary results on color normalization indicate that transforming objects to a canonical object color space could boost the performance of the proposed object detector if combined with color features. Both of the aforementioned topics (discriminative postprocessing and color normalization) are possible areas of future research.

This work emphasizes the importance of object part choice for the performance of an object detector. Experiments with landmarks selected from a dense grid within a bounding box showed that semantically meaningful object parts might not be optimal for object detection. Thus, one direction for further research is unsupervised selection of optimal object parts. In Caltech-101 the majority of the classes have only minor pose variations, which explains the success of dense grid object parts. From image to image, the generated points should correspond to approximately the same region of the object, which is impossible in most modern datasets containing big pose variations. A possible solution to this problem can be found from interest point driven alignment of objects prior to dense grid generation. Based on the alignment results (similarity graph), images can be divided into groups by object pose similarity, thus assuring only minor pose changes within a group. This idea is based on a recent unsupervised object alignment method [188].

Another area for extending the developed object detector is detection of objects in 3D. The current 2D constellation model can be extended to 3D by using 3D similarity trans-

formation instead of 2D transformation. A PASCAL3D+ dataset [187] with annotated object parts and 3D object models provides a suitable benchmark to proceed in this direction.

[1] Agarwal, S., Awan, A., and Roth, D. Learning to detect objects in image via a sparse, part-based representation. *Transactions on Pattern Analysis and Machine Intelligence 26*, 11 (2004), 1475–1490.

[2] Agarwal, S., and Roth, D. Learning a sparse representation for object detection. In *European Conference on Computer Vision (ECCV)* (2002).

[3] Alexe, B., Deselaers, T., and Ferrari, V. What is an object? In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2010).

[4] Alexe, B., Deselaers, T., and Ferrari, V. Measuring the objectness of image windows. *Transactions on Pattern Analysis and Machine Intelligence 34*, 11 (2012), 2189–2202.

[5] Allan, M., and Williams, C. Object localisation using the generative template of features. *Computer Vision and Image Understanding 113* (2009), 824–838.

[6] Alvarez, S., Sotelo, M., Parra, I., Llorca, D., and Gavilán, M. Vehicle and pedestrian detection in esafety applications. In *Proceedings of the World Congress on Engineering and Computer Science* (2009), vol. 2.

[7] Alvira, M., and Rifkin, R. An empirical comparison of snow and svms for face detection. A.I. memo 2001-004, Center for Biological and Computational Learning, MIT, Cambridge, MA, 2001.

[8] Amit, Y., and Trouvé, A. Pop: Patchwork of parts models for object recognition. *International Journal of Computer Vision 75*, 2 (2007), 267–282.

[9] Andrews, S., Tsochantaridis, I., and Hofmann, T. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems (NIPS)* (2002).

[10] Andriluka, M., Roth, S., and Schiele, B. Pictorial structures revisited: People detection and articulated pose estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).

[11] Aytekin, C., Kiranyaz, S., and Gabbouj, M. Automatic object segmentation by quantum cuts. In *International Conference on Pattern Recognition (ICPR)* (2014).

[12] Bar-Hillel, A., and Weinshall, D. Efficient learning of relational object class models. *International Journal of Computer Vision 77* (2008), 175–198.

[13] Bay, H., Tuytelaars, T., and Van Gool, L. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)* (2006).

[14] Berg, A. C., Berg, T. L., and Malik, J. Shape matching and object recognition using low distortion correspondences. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2005).

[15] Bergtholdt, M., Kappes, J., Schmidt, S., and Schnör, C. A study of parts-based object class detection using complete graphs. *International Journal of Computer Vision 87* (2010), 93–117.

[16] Bilmes, J. A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models, 1997.

[17] Blaschko, M. B., and Lampert, C. H. Learning to localize objects with structured output regression. In *European Conference on Computer Vision (ECCV)* (2008).

[18] Bosch, A., Zisserman, A., and Munoz, X. Representing shape with a spatial pyramid kernel. In *International Conference on Image and Video Retrieval (CIVR)* (2007).

[19] Bosch, A., Zisserman, A., and Muoz, X. Image classification using random forests and ferns. In *International Conference on Computer Vision (ICCV)* (2007).

[20] Bosch, A., Zisserman, A., and Muoz, X. Scene classification using a hybrid generative/discriminative approach. *Transactions on Pattern Analysis and Machine Intelligence 30*, 4 (April 2008), 712–727.

[21] Bourdev, L., and Malik, J. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision (ICCV)* (2009).

[22] Bovik, A. C., Clark, M., and Geisler, W. S. Multichannel texture analysis using localized spatial filters. *Transactions on Pattern Analysis and Machine Intelligence 12*, 1 (January 1990), 55–73.

[23] Bradski, G. Open source computer vision library, opencv. *Dr. Dobb's Journal of Software Tools* (2000).

[24] Breiman, L. Random forests. *Machine Learning 45*, 1 (2001), 5–32.

[25] Buch, A. G., Kraft, D., Kämäräinen, J.-K., Petersen, H. G., and Krüger, N. Pose estimation using local structure-specific shape and appearance context. In *International Conference on Robotics and Automation(ICRA)* (2013).

[26] Cao, Y., Wang, C., Li, Z., Zhang, L., and Zhang, L. Spatial bag-of-features. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2010).

[27] Carbonetto, P., Dorko, G., Schmid, C., Kuck, H., and de Freitas, N. Learning to recognize objects with little supervision. *International Journal of Computer Vision 77* (2008), 219–237.

[28] Chang, C.-C., and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2* (2011), 27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[29] Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. Return of the devil in the details: Delving deep into convolutional nets. *arXiv.org* (2014).

[30] Chen, K., Gong, S., Xiang, T., and Loy, C. C. Cumulative attribute space for age and crowd density estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2013).

[31] Chen, Y., Zhu, L., Yuille, A., and Zhang, H. Unsupervised learning of probabilistic object models (POMs) for object classification, segmentation, and recognition using knowledge propagation. *Transactions on Pattern Analysis and Machine Intelligence 31*, 10 (2009), 1747–1761.

[32] Cootes, T., Taylor, C., Cooper, D., and Graham, J. Active shape models – their training and application. *Computer Vision and Image Understanding 61*, 1 (1995), 38–59.

[33] Cox, M., Sridharan, S., Lucey, S., and Cohn, J. Least squares congealing for unsupervised alignment of images. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2008).

[34] Crandall, D., Felzenszwalb, P., and Huttenlocher, D. Spatial priors for part-based recognition using statistical models. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2005).

[35] Crandall, D., and Huttenlocher, D. Composite models of objects and scenes for category recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2007).

[36] Cristinacce, D., and Cootes, T. Automatic feature localisation with constrained local models. *Pattern Recognition 41* (2008), 3054–3067.

[37] Csurka, G., Dance, C., Willamowski, J., Fan, L., and Bray, C. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, European Conference on Computer Vision (ECCV)* (2004).

[38] Dai, J., Hong, Y., Hu, W., Zhu, S.-C., and Wu, Y. N. Unsupervised learning of dictionaries of hierarchical compositional models. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2014).

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

[40] Dalal, N., and Triggs, B. Inria person dataset, 2005.

[41] Daugman, J. High confidence visual recognition of persons by a test of statistical independence. *Transactions on Pattern Analysis and Machine Intelligence 15*, 11 (1993), 1148–1161.

[42] Daugman, J. G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *A Journal of the Optical Society of America 2*, 7 (1985), 1160–1169.

[43] Dibeklioglu, H., Salah, A., and Gevers, T. A statistical method for 2-d facial landmarking. *IEEE Transactions on Image Processing 21*, 2 (2012), 844–858.

[44] Dorkó, G., and Schmid, C. Selection of scale-invariant parts for object class recognition. In *International Conference on Computer Vision (ICCV)* (2003).

[45] Eichner, M., and Ferrari, V. Better appearance models for pictorial structures. In *British Machine Vision Conference (BMVC)* (2009).

[46] Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision 111*, 1 (2014), 98–136.

[47] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision 88*, 2 (2010), 303–338.

[48] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[49] Everitt, B., and Hand, D. *Finite Mixture Distributions*. Monographs on Applied Probability and Statistics. Chapman and Hall, 1981.

[50] Fei-Fei, L., Fergus, R., and Perona, P. One-shot learning of object categories. *Transactions on Pattern Analysis and Machine Intelligence 28*, 4 (2006), 594.

[51] Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding 106*, 1 (2007), 59–70.

[52] Fei-Fei, L., and Perona, P. A bayesian hierarchical model for learning natural scene categories. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2005), vol. 2, IEEE, pp. 524–531.

[53] Felzenszwalb, P., and Huttenlockher, D. Pictorial structures for object recognition. *International Journal of Computer Vision 61*, 1 (2005), 55–79.

[54] Felzenszwalb, P. F., Girshick, R. B., and McAllester, D. Cascade object detection with deformable part models. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2010).

[55] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. Object detection with discriminatively trained part-based models. *Transactions on Pattern Analysis and Machine Intelligence 32*, 9 (2010), 1627–1645.

[56] Fergus, R., Perona, P., and Zisserman, A. Object class recognition by unsupervised scale-invariant learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2003).

[57] Fergus, R., Perona, P., and Zisserman, A. A sparse object category model for efficient learning and exhaustive recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2005).

[58] Ferrari, V., Tuytelaars, T., and Van Gool, L. Simultaneous object recognition and segmentation by image exploration. In *European Conference on Computer Vision (ECCV)* (2004).

[59] Fidler, S., and Leonardis, A. Towards scalable representations of object categories: Learning a hierarchy of parts. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2007).

[60] Figueiredo, M., and Jain, A. Unsupervised learning of finite mixture models. *Transactions on Pattern Analysis and Machine Intelligence 24*, 3 (2002).

[61] Finlayson, G. D., and Schaefer, G. Solving for colour constancy using a constrained dichromatic reflection model. *International Journal of Computer Vision 42*, 3 (2001), 127–144.

[62] Fischler, M. A., and Elschlager, R. A. The representation and matching of pictorial structures. *IEEE Transactions on Computers 22*, 1 (1973), 67–92.

[63] Forsyth, D. A. A novel algorithm for color constancy. *International Journal of Computer Vision 5*, 1 (1990), 5–35.

[64] Fritz, M., Leibe, B., Caputo, B., and Schiele, B. Integrating representative and discriminant models for object category detection. In *International Conference on Computer Vision (ICCV)* (2005).

[65] Gabor, D. Theory of communication. *Journal of Institution of Electrical Engineers 93* (1946), 429–457.

[66] Gavrila, D. M., and Munder, S. Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision 73*, 1 (2007), 41–59.

[67] Gershon, R., Jepson, A. D., and Tsotsos, J. K. From [r, g, b] to surface reflectance: Computing color constant descriptors in images. In *International Joint Conference on Artificial Intelligence (IJCAI)* (1987).

[68] Gijsenij, A., Gevers, T., and van de Weijer, J. Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing 20*, 9 (2011), 2475–2489.

[69] Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *arXiv.org* (2013).

[70] Girshick, R. B., Iandola, F. N., Darrell, T., and Malik, J. Deformable part models are convolutional neural networks. *arXiv.org* (2014).

[71] Goodman, N. Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction). *The Annals of Mathematical Statistics 34*, 1 (1963), 152–177.

[72] Gorkani, M. M., and Picard, R. W. Texture orientation for sorting photos" at a glance". In *International Conference on Pattern Recognition (ICPR)* (1994).

[73] Griffin, G., Holub, A., and Perona, P. Caltech-256 object category dataset.

[74] Gu, W., Xiang, C., Venkatesh, Y., Huang, D., and Lin, H. Facial expression recognition using radial encoding of local gabor features and classifier synthesis. *Pattern Recognition 45*, 1 (2012), 80–91.

[75] Guo, G., Mu, G., Fu, Y., and Huang, T. S. Human age estimation using bio-inspired features. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).

[76] Hamouz, M., Kittler, J., Kamarainen, J.-K., Paalanen, P., Kalviainen, H., and Matas, J. Feature-based affine-invariant localization of faces. *Transactions on Pattern Analysis and Machine Intelligence 27*, 9 (2005), 1490–1495.

[77] Han, J., and Ma, K.-K. Rotation-invariant and scale-invariant gabor features for texture image retrieval. *Image and Vision Computing 25*, 9 (2007), 1474–1481.

[78] Hariharan, B., Malik, J., and Ramanan, D. Discriminative decorrelation for clustering and classification. In *European Conference on Computer Vision (ECCV)* (2012).

[79] Heitz, G., Elidan, G., Packer, B., and Koller, D. Shape-based object localization for descriptive classification. *International Journal of Computer Vision 84* (2009), 40–62.

[80] Hietanen, A., Lankinen, J., Buch, A. G., Kamarainen, J.-j., and Kruger, N. A comparison of feature detectors and descriptors for object class matching. *Neurocomputing (in press)* (2015).

[81] Ho-Phuoc, T., Guyader, N., Landragin, F., and Guerin-Dugue, A. When viewing natural scenes, do abnormal colors impact on spatial or temporal parameters of eye movements? *Journal of Vision 12*, (2):4 (2012).

[82] Holub, A., Welling, M., and Perona, P. Hybrid generative-discriminative visual categorization. *International Journal of Computer Vision 77* (2008), 239–258.

[83] Huang, Y., Wu, Z., Wang, L., and Tan, T. Feature coding in image classification: A comprehensive study. *Transactions on Pattern Analysis and Machine Intelligence 36*, 3 (2014), 493–506.

[84] Hubel, D., and Wiesel, T. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology 195* (1968), 215–243.

[85] Ilonen, J. *Supervised Local Image Feature Detection.* PhD thesis, Lappeenranta University of Technology, 2007.

[86] Ilonen, J., Kamarainen, J.-K., and Kälviäinen, H. Fast extraction of multi-resolution gabor features. In *International Conference on Image Analysis and Processing (ICIAP)* (2007).

[87] Ilonen, J., Kamarainen, J.-K., Paalanen, P., Hamouz, M., Kittler, J., and Kälviäinen, H. Image feature localization by multiple hypothesis testing of Gabor features. *IEEE Transactions on Image Processing 17*, 3 (2008), 311–325.

[88] Jaiantilal, A. Classification and regression by randomforest-matlab. Available at `https://code.google.com/p/randomforest-matlab`, 2009.

[89] Jain, A., Chen, Y., and Demirkus, M. Pores and ridges: Fingerprint matching using level 3 features. *Transactions on Pattern Analysis and Machine Intelligence 29*, 1 (2007), 15–27.

[90] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. *arXiv.org* (2014).

[91] Jin, Y., and Geman, S. Context and hierarchy in a probabilistic image model. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2006).

[92] Joulin, A., Bach, F., and Ponce, J. Discriminative clustering for image co-segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2010).

[93] Kamarainen, J.-K., Kyrki, V., and Kälviäinen, H. Invariance properties of Gabor filter based features - overview and applications. *IEEE Transactions on Image Processing 15*, 5 (2006), 1088–1099.

[94] Kapoor, A., and Winn, J. Located hidden random fields: Learning discriminative parts for object detection. In *European Conference on Computer Vision (ECCV)* (2006).

[95] Kasper, A., Xue, Z., and Dillmann, R. The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. In *The International Journal of Robotics Research* (2012), vol. 31, pp. 927–934.

[96] Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. On combining classifiers. *Transactions on Pattern Analysis and Machine Intelligence 20*, 3 (1998), 226–239.

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

[98] Kumar, M., Zisserman, A., and Torr, P. Efficient discriminative learning of parts-based models. In *International Conference on Computer Vision (ICCV)* (2009).

[99] Kyrki, V., Kamarainen, J.-K., and Kälviäinen, H. Simple Gabor feature space for invariant object recognition. *Pattern Recognition Letters 25*, 3 (2003), 311–318.

[100] Lades, M., Vorbrüggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R. P., and Konen, W. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers 42* (1993), 300–311.

[101] Lankinen, J., and Kamarainen, J.-K. Local feature based unsupervised alignment of object class images. In *British Machine Vision Conference (BMVC)* (2011).

[102] Lasserre, J. A., Bishop, C. M., and Minka, T. P. Principled hybrids of generative and discriminative models. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2006).

[103] Lazebnik, S., Schmid, C., and Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2006).

[104] Lee, T. S. Image representation using 2d gabor wavelets. *Transactions on Pattern Analysis and Machine Intelligence 18*, 10 (1996), 959–971.

[105] Leibe, B., Ettlin, A., and Schiele, B. Learning semantic object parts for object categorization. *Image and Vision Computing 26*, 1 (2008), 15–26.

[106] Leibe, B., Leonardis, A., and Schiele, B. Combined object categorization and segmentation with an implicit shape model. In *Workshop on statistical learning in computer vision, European Conference on Computer Vision (ECCV)* (2004).

[107] Li, L., Su, H., Lim, Y., and Fei-Fei, L. Object bank: An object-level image representation for high-level visual recognition. *International Journal of Computer Vision 107* (2014), 20–39.

[108] Li, Y., Shapiro, L. G., and Bilmes, J. A. A generative/discriminative learning algorithm for image classification. In *International Conference on Computer Vision (ICCV)* (2005).

[109] Lin, M., C. Q., and Yan, S. Network in network. In *International Conference on Learning Representations (ICLR)* (2014).

[110] Lin, R.-S., Ross, D., Lim, J., and Yang, M.-H. Adaptive discriminative generative model and its applications. In *Advances in Neural Information Processing Systems (NIPS)* (2004).

[111] Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)* (2014).

[112] Lin, Z., Hua, G., and Davis, L. Multiple instance feature for robust part-based object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).

[113] Lowe, D. Demo software: Sift keypoint detector. http://www.cs.ubc.ca/ lowe/keypoints/.

[114] Lowe, D. G. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)* (1999).

[115] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.

[116] Mahamud, S., and Hebert, M. Iterative projective reconstruction from multiple views. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2000).

[117] Malisiewicz, T., Gupta, A., and Efros, A. A. Ensemble of exemplar-svms for object detection and beyond. In *International Conference on Computer Vision (ICCV)* (2011).

[118] Mallat, S. G. A theory for multiresolution signal decomposition: the wavelet representation. *Transactions on Pattern Analysis and Machine Intelligence 11*, 7 (1989), 674–693.

[119] Manjunath, B., and Ma, W. Texture features for browsing and retrieval of image data. *Transactions on Pattern Analysis and Machine Intelligence 18*, 8 (1996), 837–842.

[120] Martinez, B., Valstar, M., Binefa, X., and Pantic, M. Local evidence aggregation for regression based facial point detection. *Transactions on Pattern Analysis and Machine Intelligence 35*, 5 (2013), 1149–1163.

[121] M.C.Burl, M.Weber, and P.Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *European Conference on Computer Vision (ECCV)* (1998).

[122] Mutch, J., and Lowe, D. G. Multiclass object recognition with sparse, localized features. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2006).

[123] Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *arXiv.org* (2014).

[124] Niblack, C. W., Barber, R., Equitz, W., Flickner, M. D., Glasman, E. H., Petkovic, D., Yanker, P., Faloutsos, C., and Taubin, G. Qbic project: querying images by content, using color, texture, and shape. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology* (1993), International Society for Optics and Photonics, pp. 173–187.

[125] Novak, C. L., and Shafer, S. A. Supervised color constancy using a color chart. *Technical Repot CMU-CS-90-140. Carnegie Mellon University, School Of Comprter Science* (1990).

[126] Ojala, T., Pietikäinen, M., and Mäenpää, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Transactions on Pattern Analysis and Machine Intelligence 24*, 7 (2002), 971–987.

[127] Ommer, B., and Buhmann, J. Learning the compositional nature of visual object categories for recognition. *Transactions on Pattern Analysis and Machine Intelligence 32*, 3 (2010), 501–516.

[128] Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., and Poggio, T. Pedestrian detection using wavelet templates. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (1997).

[129] Paalanen, P., Kamarainen, J.-K., Ilonen, J., and Kälviäinen, H. Feature representation and discrimination based on Gaussian mixture model probability densities - practices and algorithms. *Pattern Recognition 39*, 7 (2006), 1346–1358.

[130] Papageorgiou, C., and Poggio, T. A trainable object detection system: Car detection in static images. Tech. Rep. 1673, October 1999. (CBCL Memo 180).

[131] Parkhi, O. M., Vedaldi, A., Jawahar, C., and Zisserman, A. The truth about cats and dogs. In *International Conference on Computer Vision (ICCV)* (2011).

[132] Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. Cats and dogs. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2012).

[133] Perina, A., Cristani, M., Castellani, U., Murino, V., and Jojic, N. A hybrid generative/discriminative classification framework based on free-energy terms. In *International Conference on Computer Vision (ICCV)* (2009).

[134] Ponce, J., Berg, T. L., Everingham, M., Forsyth, D. A., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, B. C., Torralba, A., Williams, C. K. I., Zhang, J., and Zisserman, A. Dataset issues in object recognition. In *Toward Category-Level Object Recognition*. Springer, 2006, pp. 29–48.

[135] Pötzsch, M., Maurer, T., Wiskott, L., and vd Malsburg, C. Reconstruction from graphs labeled with responses of gabor filters. In *International Conference Artificial Neural Networks (ICANN)* (1996).

[136] Quinlan, J. R. Induction of decision trees. *Machine learning 1*, 1 (1986), 81–106.

[137] Rao, R., and Ballard, D. An active vision architecture based on iconic representations. *Artificial Intelligence Journal 78* (1995), 461–505.

[138] Rapp, V., Senechal, T., Bailly, K., and Prevost, L. Multiple kernel learning SVM and statistical validation for facial landmark detection. In *International Conference on Automatic Face and Gesture Recognition (AFGR)* (2011).

[139] Riabchenko, E., and Kämäräinen, J.-K. Generative part-based gabor object detector. *Pattern Recognition Letters* (2015).

[140] Riabchenko, E., Kämäräinen, J.-K., and Chen, K. Density-aware part-based object detection with positive examples. In *International Conference on Pattern Recognition (ICPR)* (2014).

[141] Riabchenko, E., Kämäräinen, J.-K., and Chen, K. Learning generative models of object parts from a few positive examples. In *International Conference on Pattern Recognition (ICPR)* (2014).

[142] Riabchenko, E., Kämäräinen, J.-K., and Chen, K. Progressive visual object detection with positive training examples only. In *Scandinavian Conference on Image Analysis (SCIA)* (2015).

[143] Riabchenko, E., Lankinen, J., Buch, A., Kämäräinen, J., and Kruger, N. Supervised object class colour normalisation. In *Scandinavian Conference on Image Analysis (SCIA)* (2013).

[144] Riaz, F., Hassan, A., Rehman, S., and Qamar, U. Texture classification using rotation-and scale-invariant gabor texture features. *Signal Processing Letters 20*, 6 (2013), 607–610.

[145] Rodriguez, Y., Cardinaux, F., Bengio, S., and Mariéthoz, J. Measuring the performance of face localization systems. *Image and Vision Computing 24* (2006), 882–893.

[146] Rosenfeld, A. Picture processing by computer. *ACM Computing Surveys (CSUR) 1*, 3 (1969), 147–176.

[147] Rowley, H. A., Baluja, S., and Kanade, T. Neural network-based face detection. *Transactions on Pattern Analysis and Machine Intelligence 20*, 1 (1998), 23–38.

[148] Russakovsky, O., Deng, J., Huang, Z., Berg, A. C., and Fei-Fei, L. Detecting avocados to zucchinis: what have we done, and where are we going? In *International Conference on Computer Vision (ICCV)* (2013).

[149] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* (2015).

[150] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *arXiv.org* (2014).

[151] Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision 77*, 1-3 (2008), 157–173.

[152] Sampo, J., Kamarainen, J.-K., Heiliö, M., and Kälviäinen, H. Measuring translation shiftability of frames. *Computers & Mathematics with Applications 52*, 6-7 (2006), 1089–1098.

[153] Schölkopf, B., and Smola, A. *Learning with kernels : support vector machines, regularization, optimization, and beyond*. MIT Press, 2002.

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

[155] Shen, L., and Bai, L. A review on Gabor wavelets for face recognition. *Pattern Analysis and Applications 9*, 2-3 (2006).

[156] Simoncelli, E., Freeman, W., Adelson, E., and Heeger, D. Shiftable multiscale transforms. *IEEE Transactions on Information Theory 38*, 2 (1992), 587–607.

[157] Simonyan, K., Vedaldi, A., and Zisserman, A. Deep fisher networks for large-scale image classification. In *Advances in Neural Information Processing Systems (NIPS)* (2013).

[158] Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., and Freeman, W. T. Discovering objects and their location in images. In *International Conference on Computer Vision (ICCV)* (2005).

[159] Sivic, J., and Zisserman, A. Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision (ICCV)* (2003).

[160] Sun, M., and Savarese, S. Articulated part-based model for joint object detection and pose estimation. In *International Conference on Computer Vision (ICCV)* (2011).

[161] Suykens, J. A., and Vandewalle, J. Least squares support vector machine classifiers. *Neural Processing Letters 9*, 3 (1999), 293–300.

[162] Swain, M. J., and Ballard, D. H. Color indexing. *International Journal of Computer Vision 7*, 1 (1991), 11–32.

[163] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. *arXiv.org* (2014).

[164] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv.org* (2013).

[165] Szummer, M., and Picard, R. W. Indoor-outdoor image classification. In *IEEE International Workshop on Content-Based Access of Image and Video Databases* (1998).

[166] Tamminen, T., and Lampinen, J. Sequential Monte Carlo for Bayesian matching of objects with occlusions. *Transactions on Pattern Analysis and Machine Intelligence 28*, 6 (2006), 930–941.

[167] Tang, S., Andriluka, M., and Schiele, B. Detection and tracking of occluded people. *International Journal of Computer Vision 110*, 1 (2014), 58–69.

[168] Todorovic, S., and Ahuja, N. Unsupervised category modeling, recognition, and segmentation in images. *Transactions on Pattern Analysis and Machine Intelligence 30*, 12 (2008), 2158–2174.

[169] Tuytelaars, T., Lampert, C. H., Blaschko, M. B., and Buntine, W. Unsupervised object discovery: A comparison. *International Journal of Computer Vision 88*, 2 (2010), 284–302.

[170] Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *Transactions on Pattern Analysis and Machine Intelligence 13*, 4 (1991), 376–380.

[171] Vailaya, A., Figueiredo, M. A., Jain, A. K., and Zhang, H.-J. Image classification for content-based indexing. *IEEE Transactions on Image Processing 10*, 1 (2001), 117–130.

[172] Vapnik, V. Pattern recognition using generalized portrait method. *Automation and Remote Control 24* (1963), 774–780.

[173] Vapnik, V., and Chervonenkis, A. A note on one class of perceptrons. *Automation and Remote Control 25* (1964).

[174] Vapnik, V., and Vashist, A. A new learning paradigm: Learning using privileged information. *Neural Networks 22*, 5-6 (2009), 544–557.

[175] Vedaldi, A., and Fulkerson, B. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[176] Verbeek, J. J., Vlassis, N., and Kröse, B. Efficient greedy learning of Gaussian mixture models. *Neural Computation 5*, 2 (2003), 469–485.

[177] Viola, P., and Jones, M. Robust real-time face detection. *International Journal of Computer Vision 57*, 2 (2004), 137–154.

[178] Vondrick, C., Khosla, A., Malisiewicz, T., and Torralba, A. Hoggles: Visualizing object detection features. In *International Conference on Computer Vision (ICCV)* (2013).

[179] Vukadinovic, D., and Pantic, M. Fully automatic facial feature point detection using Gabor feature based boosted classifiers. In *Interational Confernce on Systems, Man and Cybernetics (ICSMC)* (2005).

[180] Wang, X., Lin, L., Huang, L., and Yan, S. Incorporating structural alternatives and sharing into hierarchy for multiclass object recognition and detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2013).

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

[182] Wang, Y., Tran, D., and Liao, Z. Learning hierarchical poselets for human parsing. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2011).

[183] Weber, M., Welling, M., and Perona, P. Unsupervised learning of models for recognition. In *European Conference on Computer Vision (ECCV)* (2000).

[184] Weinzaepfel, P., Jégou, H., and Pérez, P. Reconstructing an image from its local descriptors. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2011).

[185] Wiskott, L., Fellous, J.-M., Krüger, N., and von der Malsburg, C. Face recognition by elastic bunch graph matching. *Transactions on Pattern Analysis and Machine Intelligence 19* (1997), 775–779.

[186] Wu, Y., Si, Z., Gong, H., and Zhu, S.-C. Learning active basis model for object detection and recognition. *International Journal of Computer Vision 90* (2010), 198–235.

[187] Xiang, Y., Mottaghi, R., and Savarese, S. Beyond PASCAL: A benchmark for 3d object detection in the wild. In *Winter Conference on Applications of Computer Vision (WACV)* (2014).

[188] Yancheshmeh, F. S., Chen, K., and Kamarainen, J.-K. Unsupervised visual alignment with similarity graphs. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[189] Yang, H., and Patras, I. Face parts localization using structured-output regression forests. In *Asian Conference on Computer Vision (ACCV)* (2013).

[190] Yang, J., Li, Y., Tian, Y., Duan, L., and Gao, W. Group-sensitive multiple kernel learning for object categorization. In *International Conference on Computer Vision (ICCV)* (2009).

[191] Yang, J., Shi, Y., and Yang, J. Finger-vein recognition based on a bank of Gabor filters. In *Asian Conference on Computer Vision (ACCV)* (2009).

[192] Yang, M., Zhang, L., Shiu, S. C., and Zhang, D. Gabor feature based robust representation and classification for face recognition with gabor occlusion dictionary. *Pattern Recognition 46*, 7 (2013), 1865–1878.

[193] Yang, Y., and Ramanan, D. Articulated pose estimation with flexible mixtures-of-parts. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2011).

[194] Zhang, D.-Q., and Chang, S.-F. A generative-discriminative hybrid method for multi-view object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2006).

[195] Zhang, H., Berg, A. C., Maire, M., and Malik, J. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2006).

[196] Zhang, L., and Maaten, L. v. d. Improving object tracking by adapting detectors. In *International Conference on Pattern Recognition (ICPR)* (2014).

[197] Zhang, N., Donahue, J., Girshick, R., and Darrell, T. Part-based R-CNNs for fine-grained category detection. In *European Conference on Computer Vision (ECCV)* (2014).

[198] Zhang, N., Donahue, J., Girshick, R., and Darrell, T. Part-based r-cnns for fine-grained category detection. In *Computer Vision–ECCV 2014*. Springer, 2014, pp. 834–849.

[199] Zhao, G., and Pietikainen, M. Dynamic texture recognition using local binary patterns with an application to facial expressions. *Transactions on Pattern Analysis and Machine Intelligence 29*, 6 (2007), 915–928.

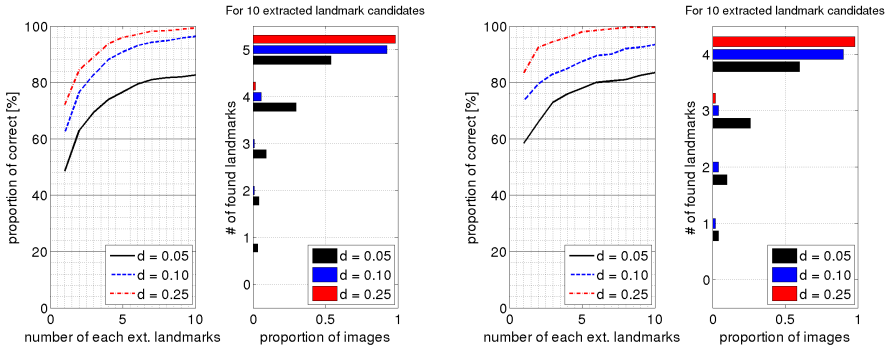
[200] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems (NIPS)* (2014).

[201] Zhu, L., Chen, Y., and Yuille, A. Unsupervised learning of probabilistic Grammar-Markov models for object categories. *Transactions on Pattern Analysis and Machine Intelligence 31*, 1 (2009), 114–128.

[202] Zhu, L., Chen, Y., Yuille, A., and Freeman, W. Latent hierarchical structural learning for object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2010).

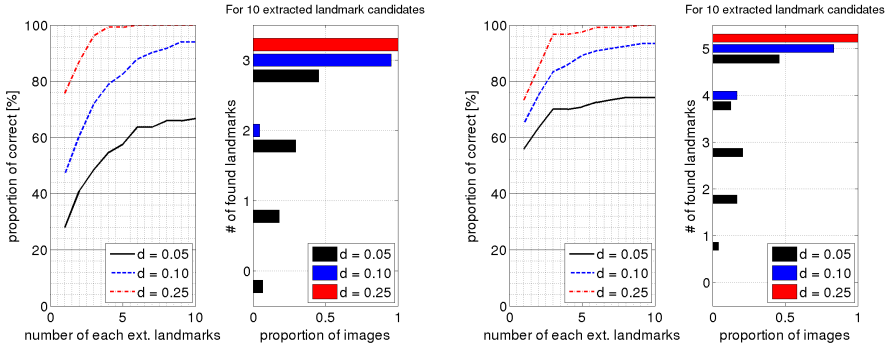
Appendix I

Gabor Local Part Detector Example Images



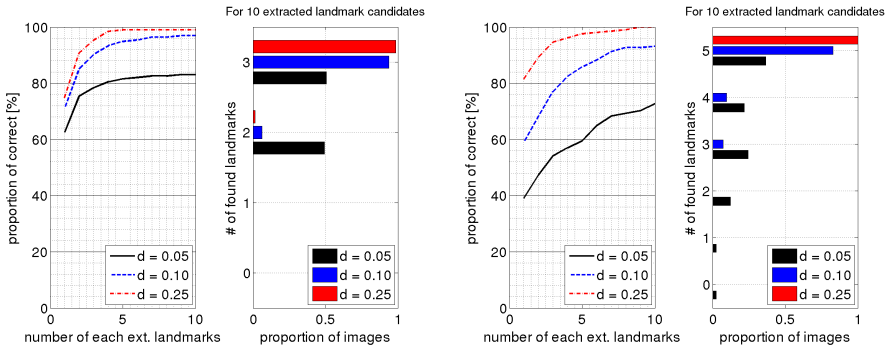
(a) watch

(b) grand_piano



(c) menorah

(d) dollar_bill



(e) car_side

(f) revolver

Figure I.1: Caltech-101 landmark detection (left: cumulative detection graph, right: detection bars for 10 best candidates).

Appendix II

Part-Based Gabor Object Detector Example Images

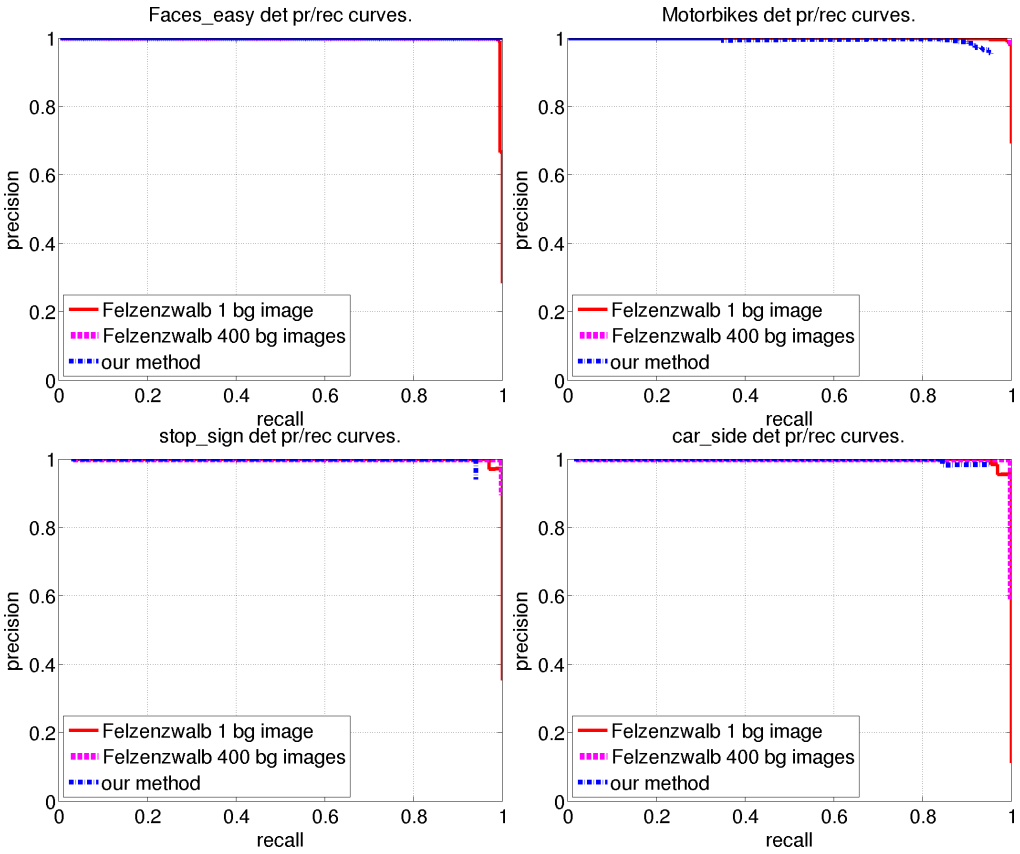


Figure II.1: Comparison of the generative positive examples only method and a state-of-the-art discriminative method (Felzenszwalb et al. [55]) in the object detection task of Caltech-101 categories: *faces, motorbikes, stop sign* and *car side* (from left-to-right and top-down).

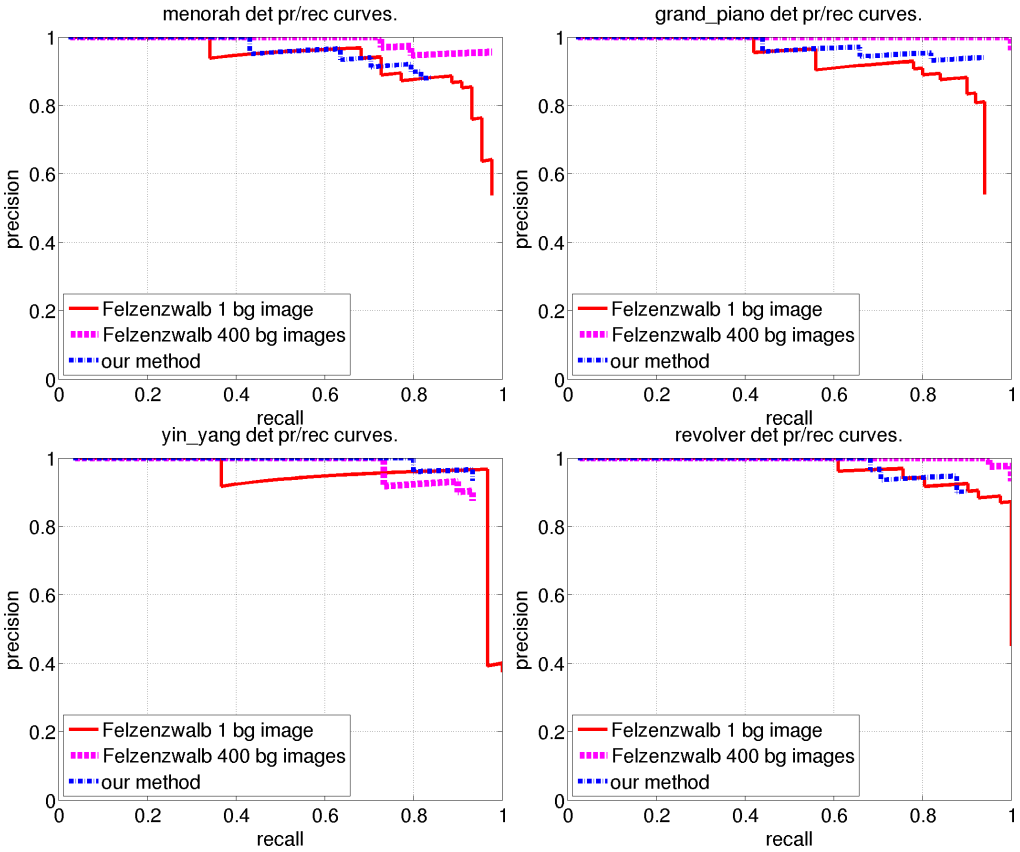
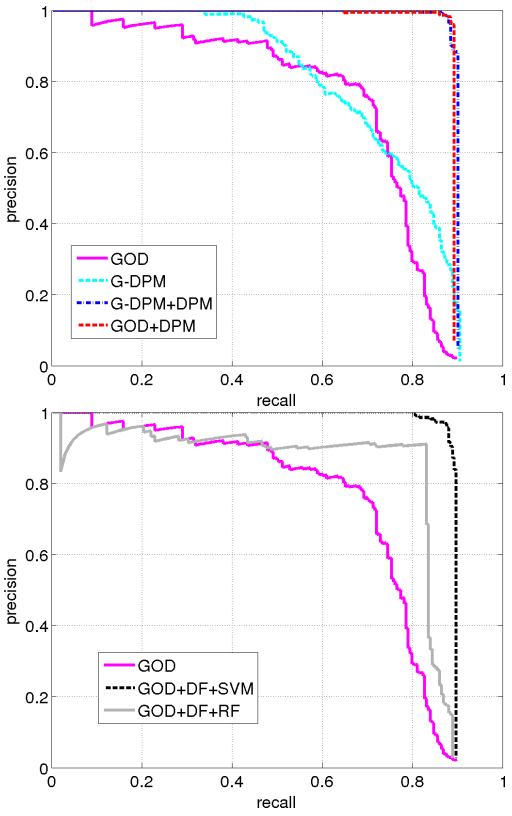


Figure II.2: Comparison of the generative positive examples only method and a state-of-the-art discriminative method (Felzenszwalb et al. [55]) in the object detection task of Caltech-101 categories: *menorah, grand piano, yin yang* and *revolver* (from left-to-right and top-down).

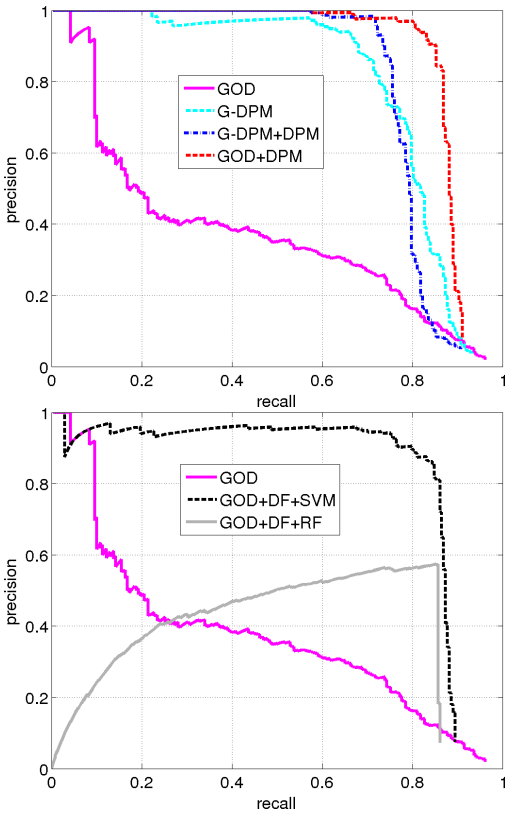
Appendix III

Genertive-Discriminative Hybrid Example Images



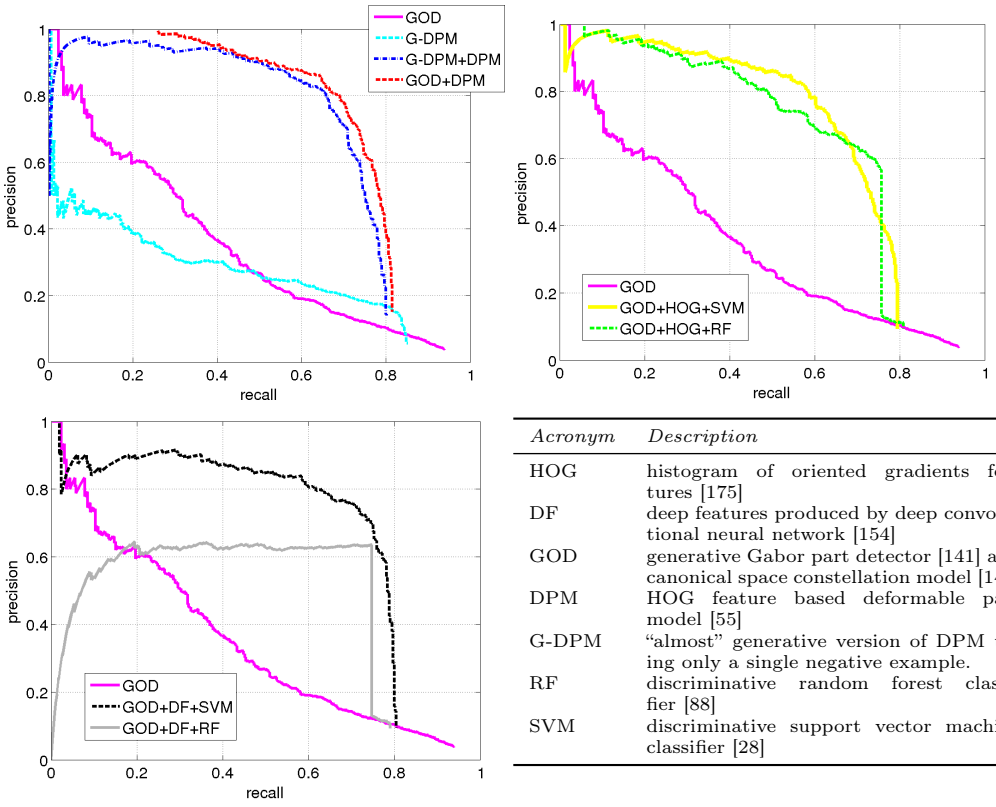
Acronym	*Description*
HOG	histogram of oriented gradients features [175]
DF	deep features produced by deep convolutional neural network [154]
GOD	generative Gabor part detector [141] and canonical space constellation model [140]
DPM	HOG feature based deformable part model [55]
G-DPM	"almost" generative version of DPM using only a single negative example.
RF	discriminative random forest classifier [88]
SVM	discriminative support vector machine classifier [28]

Figure III.1: Precision-recall curves for the Imagenet category *grey owl*.



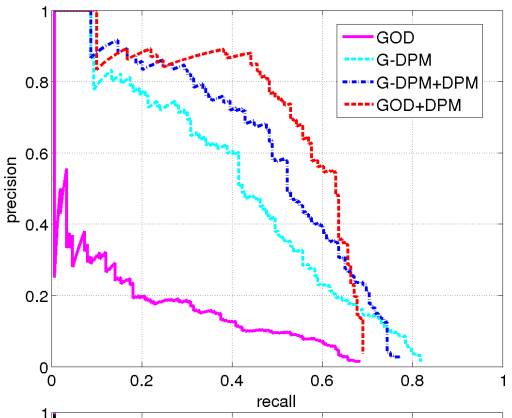
Acronym	*Description*
HOG	histogram of oriented gradients features [175]
DF	deep features produced by deep convolutional neural network [154]
GOD	generative Gabor part detector [141] and canonical space constellation model [140]
DPM	HOG feature based deformable part model [55]
G-DPM	"almost" generative version of DPM using only a single negative example.
RF	discriminative random forest classifier [88]
SVM	discriminative support vector machine classifier [28]

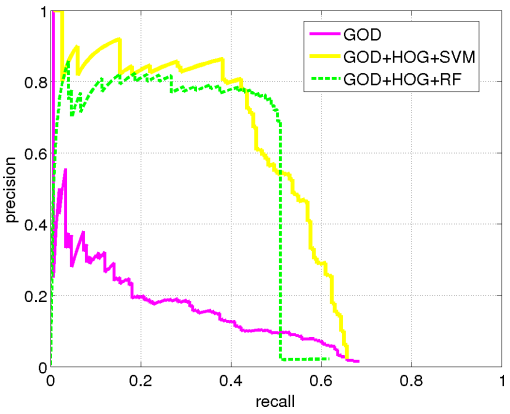
Figure III.2: Precision-recall curves for the Imagenet category *acoustic guitar*.

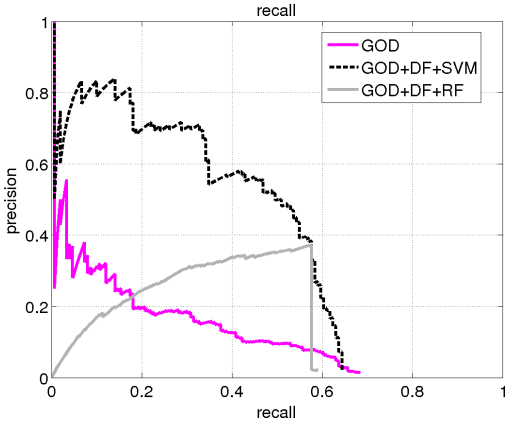


Acronym	*Description*
HOG	histogram of oriented gradients features [175]
DF	deep features produced by deep convolutional neural network [154]
GOD	generative Gabor part detector [141] and canonical space constellation model [140]
DPM	HOG feature based deformable part model [55]
G-DPM	"almost" generative version of DPM using only a single negative example.
RF	discriminative random forest classifier [88]
SVM	discriminative support vector machine classifier [28]

Figure III.3: Precision-recall curves for the Imagenet category *garden spider*.







Acronym	*Description*
HOG	histogram of oriented gradients features [175]
DF	deep features produced by deep convolutional neural network [154]
GOD	generative Gabor part detector [141] and canonical space constellation model [140]
DPM	HOG feature based deformable part model [55]
G-DPM	"almost" generative version of DPM using only a single negative example.
RF	discriminative random forest classifier [88]
SVM	discriminative support vector machine classifier [28]

Figure III.4: Precision-recall curves for the Imagenet category *snail*.

Appendix IV

Supervised Object Class Color Normalisation Example Images

Figure IV.1: *Strawberry* and *Garfield* (originals left).

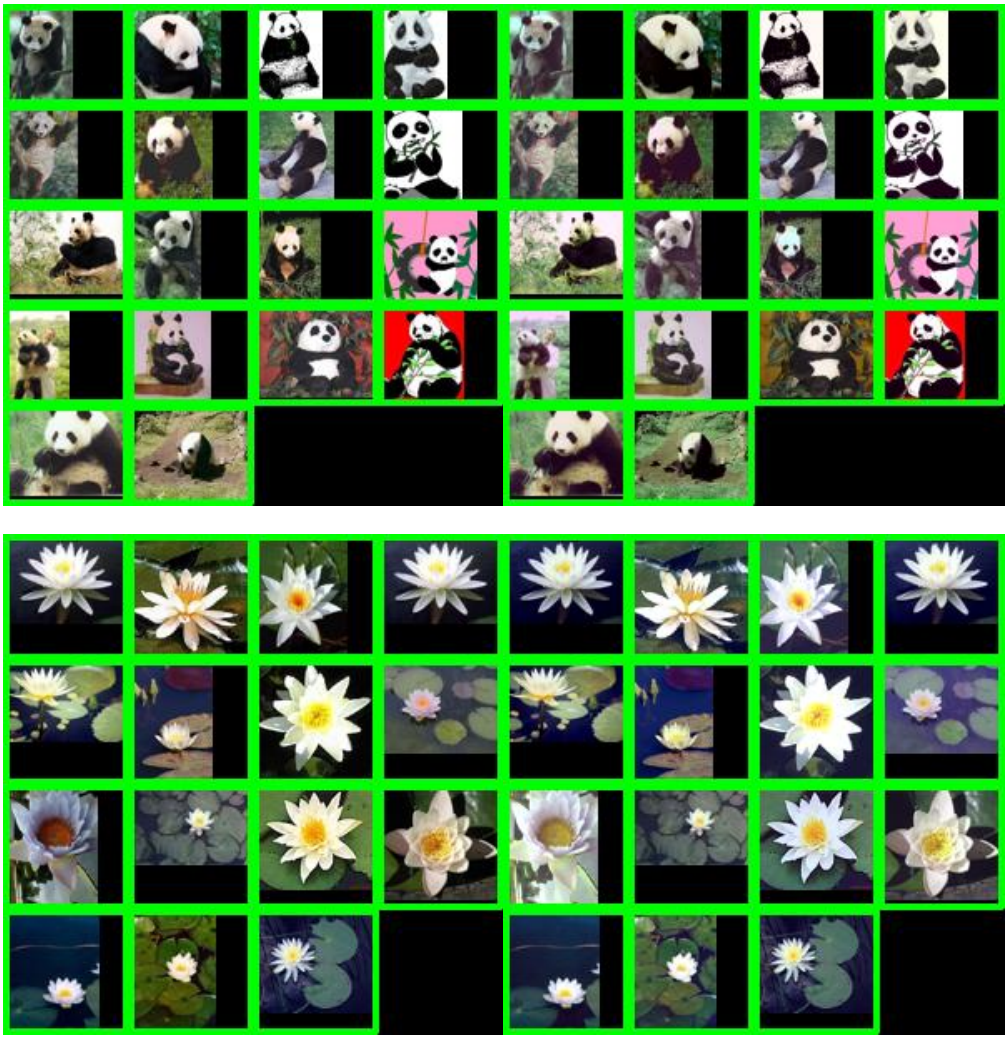


Figure IV.2: *Panda* and *water lily* (originals left).

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

624. RANTANEN, NOORA. The family as a collective owner – identifying performance factors in listed companies. 2014. Diss.

625. VÄNSKÄ, MIKKO. Defining the keyhole modes – the effects on the molten pool behavior and the weld geometry in high power laser welding of stainless steels. 2014. Diss.

626. KORPELA, KARI. Value of information logistics integration in digital business ecosystem. 2014. Diss.

627. GRUDINSCHI, DANIELA. Strategic management of value networks: how to create value in cross-sector collaboration and partnerships. 2014. Diss.

628. SKLYAROVA, ANASTASIA. Hyperfine interactions in the new Fe-based superconducting structures and related magnetic phases. 2015. Diss.

629. SEMKEN, R. SCOTT. Lightweight, liquid-cooled, direct-drive generator for high–power wind turbines: motivation, concept, and performance. 2015. Diss.

630. LUOSTARINEN, LAURI. Novel virtual environment and real-time simulation based methods for improving life-cycle efficiency of non-road mobile machinery. 2015. Diss.

631. ERKKILÄ, ANNA-LEENA. Hygro-elasto-plastic behavior of planar orthotropic material. 2015. Diss.

632. KOLOSENI, DAVID. Differential evolution based classification with pool of distances and aggregation operators. 2015. Diss.

633. KARVONEN, VESA. Identification of characteristics for successful university-company partnership development. 2015. Diss.

634. KIVYIRO, PENDO. Foreign direct investment, clean development mechanism, and environmental management: a case of Sub-Saharan Africa. 2015. Diss.

635. SANKALA, ARTO. Modular double-cascade converter. 2015. Diss.

636. NIKOLAEVA, MARINA. Improving the fire retardancy of extruded/coextruded wood-plastic composites. 2015. Diss.

637. ABDEL WAHED, MAHMOUD. Geochemistry and water quality of Lake Qarun, Egypt. 2015. Diss.

638. PETROV, ILYA. Cost reduction of permanent magnet synchronous machines. 2015. Diss.

639. ZHANG, YUNFAN. Modification of photocatalyst with enhanced photocalytic activity for water treatment. 2015. Diss.

640. RATAVA, JUHO. Modelling cutting states in rough turning of 34CrNiMo6 steel. 2015. Diss.

641. MAYDANNIK, PHILIPP. Roll-to-roll atomic layer deposition process for flexible electronics applications. 2015. Diss.

642. SETH, FRANK. Empirical studies on software quality construction: Exploring human factors and organizational influences. 2015. Diss.

643. SMITH, AARON. New methods for controlling twin configurations and characterizing twin boundaries in 5M Ni-Mn-Ga for the development of applications. 2015. Diss.

644. NIKKU, MARKKU. Three-dimensional modeling of biomass fuel flow in a circulating fluidized bed furnace. 2015. Diss.

645. HENTTU, VILLE. Improving cost-efficiency and reducing environmental impacts of intermodal transportation with dry port concept – major rail transport corridor in Baltic Sea region. 2015. Diss.

646. HAN, BING. Influence of multi-phase phenomena on semibatch crystallization processes of aqueous solutions. 2015. Diss.

647. PTAK, PIOTR. Aircraft tracking and classification with VHF passive bistatic radar. 2015. Diss.

648. MAKKONEN, MARI. Cross-border transmission capacity development – Experiences from the Nordic electricity markets. 2015. Diss.

649. UUSITALO, ULLA-MAIJA. Show me your brain! Stories of interdisciplinary knowledge creation in practice. Experiences and observations from Aalto Design Factory, Finland. 2015. Diss.

650. ROOZBAHANI, HAMID. Novel control, haptic and calibration methods for teleoperated electrohydraulic servo systems. 2015. Diss.

651. SMIRNOVA, LIUDMILA. Electromagnetic and thermal design of a multilevel converter with high power density and reliability. 2015. Diss.

652. TALVITIE, JOONAS. Development of measurement systems in scientific research: Case study. 2015. Diss.

653. ZUBEDA, MUSSA. Variational ensemble kalman filtering in hydrology. 2015. Diss.

654. STEPANOV, ALEXANDER. Feasibility of industrial implementation of laser cutting into paper making machines. 2015. Diss.

655. SOKOLOV, MIKHAIL. Thick section laser beam welding of structural steels: methods for improving welding efficiency. 2015. Diss.

656. GORE, OLGA. Impacts of capacity remunerative mechanisms on cross-border trade. 2015. Diss.

657. AURINKO, HANNU. Risk assessment of modern landfill structures in Finland. 2015. Diss.

658. KAIJANEN, LAURA. Capillary electrophoresis: Applicability and method validation for biorefinery analytics. 2015. Diss.

659. KOLHINEN, JOHANNA. Yliopiston yrittäjämäisyyden sosiaalinen rakentuminen. Case: Aalto-yliopisto. 2015. Diss.

660. ANNALA, SALLA. Households' willingness to engage in demand response in the Finnish retail electricity market: an empirical study. 2015. Diss.