Active Occlusion-handling for Appearance-based Object Recognition Models

Marvin Struwe

Dissertation

Thesis submitted for obtaining the academic degree

Doctor of Engineering (Dr.-Ing.)

The thesis was submitted to the Faculty of Technology, Bielefeld University (P.O. Box 10 01 31, D-33501 Bielefeld, Germany), on May 21st, 2016.

Marvin Struwe
marvin.struwe@arcor.de

Gedruckt auf alterungsbeständigem Papier ○○ ISO 9706.

Abstract

Despite extensive research, visual detection of objects in natural scenes is still not robustly solved. The reason for this is the large variation in appearance in which objects or classes occur. A particularly challenging variation is occlusion, which is caused by the constellation of objects in a scene. Occlusion reduces the number of visible features of an object, but also causes accidental features. Current object representations yield acceptable results during a low to medium level of occlusion, but fail for stronger occlusions.

This thesis addresses single image-based object recognition during occlusion and proposes different occlusion-handling strategies. Initially, it depicts a holistic discriminative car detection framework, which several chapters use as reference system. Motivated by a label analysis of hand-annotated video traffic scenes, it then presents a car detector, taking car-car constellations into account. The following chapter illustrates a modification of the reference system to cover with more general occlusion constellations. Inspired by the fact that parts-based detection approaches are more robust against occlusion, the next chapter discusses a parts-based car detector with active occlusion-handling at the detection step. At first, this exploits a strategy using the mask of the occluding object to re-weight the score of possible car hypotheses. This is followed by the presentation of an extended version, which especially targets strongly occluded cars.

Due to the fact that hand-annotated video streams do not provide pixel-level information about the object instances, this thesis presents a rendered benchmark data set to resolve this issue. The pixel-level information permits intensive evaluation of occlusion-handling strategies. An eye-tracker study also uses this rendered data set to explore how humans cope with the absence of visual object features, and which information they use to deal with occlusion.

Danksagung

Mein größter Dank gilt meiner Familie und meiner Lebensgefährtin, welche mir stets viel Geduld entgegen gebracht und mir die nötige Stärke gegeben haben. Diese Unterstützung und Geborgenheit haben mir die nötige Kraft und Zeit gegeben, welche eine Dissertation von jedem einfordert.

Mein nächster Dank gilt Stefan Hasler und Thomas Weißwange, von denen ich alles wichtige über maschinelles Lernen, Klassifizierung und Lernalgorithmen gelernt habe.

Prof. Dr. Bauer-Wersing und Prof. Dr. Ritter danke ich für Ihr Vertrauen und dass Sie die vorliegende Arbeit überhaupt erst ermöglicht haben. Ich danke Prof. Dr. Bauer-Wersing dafür, dass sie meine Ausbildung fachlich gefördert hat.

Den Doktoranden und Studenten möchte ich für die regelmässigen Meetings, inklusive Kaffeepausen, danken, in denen wir viele fachliche Diskussion aber auch freundschaftliche Gespräche hatten. Der respektvolle, freundliche und kollegiale Umgang mit euch hat jeden Bürotag besonders gemacht.

Mein letzter Dank gilt all denen, welche sich oben eventuell nicht wiederfinden, aber auf Ihre eigene Art meine Dissertation beeinflusst haben. Im Speziellen...

Magda, Viktor, Benjamin, Lydia, Stefan, Florian und vielen anderen...

“Stay hungry. Stay foolish.”
– Steven ”Steve” Paul Jobs

Contents

List of Tables

List of Figures

1 Introduction

Chapter overview *This chapter motivates the topic of the thesis and presents a structural overview.*

In daily life, humans make decisions based on image information. Therefore, they have to recognize objects in highly complex scenes in order to interpret their interactions. For example, while driving a car, a person is able to recognize and classify other traffic participants, such as cars, pedestrians, bicycles, traffic signs, and traffic lights. Within a short time, the person understands the meaning of an object, as well as achieving its localization. This also works if object information is lost or the appearance of an object is modified. Because it is so easy for humans to do this, it might be assumed that it is not difficult to define algorithms that can perform this recognition. However, despite various biological and psychophysical studies, it is not fully comprehendible how human beings solve such challenging tasks. Occlusion-handling at recognition in particular is not well understood.

Object recognition is a key function for many systems, such as driver assistance, image-based localization, visual perception in robotics, industrial quality proofs, and face recognition at security checks. Current object recognition approaches in these domains show acceptable detection performance at low rates of occlusion, but fail at moderately to strongly occluded object views.

Innovations in hardware and software are decreasing the prices of components that are necessary for object recognition systems and encourage development in this area. Currently, there are many different strategies to improve the performance of object recognition systems and make them more suitable for daily-use systems, such as driver-assistance systems. Thereby, these use a variety of sensors, such as radar, laser, or cameras. Radar provides a narrow lateral resolution, but is also able to support information of visibly occluded objects by using so-called “tunneling.” This enables the radar waves to find a path around an object to capture other objects behind it. Laser provides three-dimensional (3D) information, but with limited resolution. Both radar and laser sensors are expensive. In contrast, vision-based systems such as cameras are quite cheap and provide pixel-level and multi-channel information, such as color. This multi-channel

information can be used for grouping objects, which can improve detection performance. The use of visual information is also biologically inspired since humans strongly rely on it to recognize objects. Driver-assistance systems often employ a combination of multiple sensors. For example, ultra-sonic sensors can determine the distance of objects in a near range of 1.2 to 4 meters. This type of sensor works only at close range, but provides highly accurate distance estimations. In order to capture objects at mid-range distances, a short-range radar is often used. This sensor works for a maximum distance of 30 meters.[1] By decreasing the opening angle, radar systems can also work for higher distances.

Visual stereo systems utilize disparity to obtain depth information. Multiple vision sensors are necessary to acquire the disparity information. However, accuracy decreases as the object's distance increases. Sensor suppliers of stereo systems state that depth information can be accurately obtained for distances up to 80 meters.[2] However, the operation range for the distance estimation strongly depends on the used components for the camera, e.g. photo sensor size and focal length. Due to the aforementioned highly encouraged developments in this area, the accuracy of the sensors is approaching the physical limitations. Clearly, different sensors have unique advantages and limits; therefore, a fusion of multiple sensors is often used. This provides substantial redundancy, which can serve as a backup mechanism in the case of a failing sensor. Fig. 1.1 presents an exemplary fusion of some sensors.

Some actual premium car models incorporate no fewer than 22 sensors.[2] They use 12 ultrasonic sensors - six at the front and six at the rear. At each of the four corners, a multi-mode radar is installed. Cameras are installed at each window, including the front, rear, and door windows. A long-range radar is installed at the front, and a stereo camera at the top of the windshield. These sensors all provide information about the distance or physical size of the objects.

The obtained camera images can be used for classification and color segmentation, as well as to generate depth information, if used in stereo. This thesis solely focuses on object detection by using visual information for object recognition during occlusion.

As mentioned, occlusion substantially decreases the performance of object recognition models. In many application areas, such systems must contend with many variations of occlusion, especially when used in driver-assistance systems. Fig. 1.2 depicts two daily traffic scene examples with numerous occlusion constel-

[1] The operating ranges of the sensors are taken from `https://www.continental-automotive.com/de-DE/Passenger-Cars/Chassis-Safety/Advanced-Driver-Assistance-Systems/Radars/Long-Range-Radar/Advanced-Radar-Sensor-ARS441`

[2] The operating ranges of the sensors are taken from `https://www.mercedes-benz.com/de/mercedes-benz/innovation/mercedes-benz-intelligent-drive/`

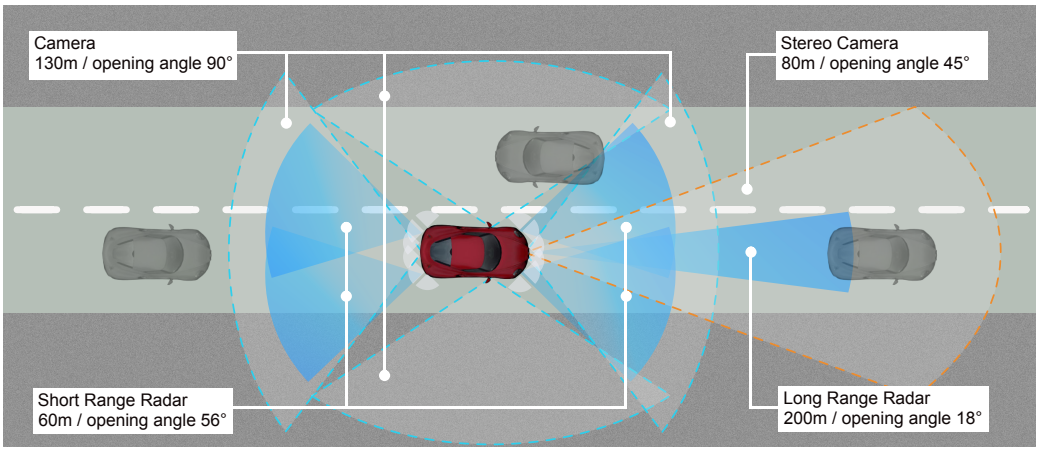


Figure 1.1: Sensor Fusion of a Driver-Assistance System: The blue-filled cones illustrate the working range of short-range and long-range radar systems. Cones with blue dotted outlines indicate the working range of four cameras to build a 360-degree view. The orange-colored outlines depict the working range of a stereo camera system. The small white-colored cones signify the working range of 12 ultrasonic sensors.

lations. Most cars are occluded by other traffic participants or stationary objects. The influence of the occlusion on the recognition system varies depending on the object representation that is used. This thesis examines two kinds of object representations: holistic discriminative approaches and parts-based approaches.

Holistic discriminative approaches employ a global template for the object representation. Their detection performance significantly diminishes with an increasing rate of occlusion.

Parts-based approaches accumulate the pure response of single features in order to build the object representation. They can more effectively handle occlu-

(a) **(b)**

Figure 1.2: Typical Traffic Scene: (a and b) show daily street scenes with strongly occluded car views.

sion, as the loss in score of a possible object hypothesis is proportional to the rate of occlusion.

In general, both object representations fail at medium to strong occlusion and require additional occlusion-handling strategies. This can be improved by approaches with explicit occlusion-handling that utilize the mask information of occluded objects in order to tackle the decreasing detection performance for occluded object views.

This thesis first presents a holistic discriminative approach, which is then adopted to generate a reference detection performance for later comparison. To manage the numerous car-car occlusion constellations, this constellation information is used to improve the detection performance. To address more general occlusion constellations, a framework is presented that uses typical object-car occlusions during training.

The second step demonstrates how depth information can be used as a plausibility check to improve the detection performance for strongly occluded car views.

Since parts-based detection approaches indicate a better ability to resolve limited occlusion, several strategies are proposed to improve their detection of occluded cars.

The quality of the ground truth data is important for a meaningful evaluation of an occlusion-handling strategy. The ground truth is often not precise because there is no pixel-level information available. Additionally, it is manually annotated and therefore prone to human mistakes. To overcome these issues, a newly rendered benchmark data set is illustrated, which provides pixel-level labeling and controlled levels of rotation and occlusion rates.

1.1 Scope of this Manuscript

Chapter 2 offers an overview of current object detection systems, with interest in how they address occlusion. Thereby, the focus is on the part-based approaches and the holistic discriminative approaches

Chapter 3 details a detection framework originally presented by Hasler et al. [2009]. This framework is applied for object detection in complex traffic scenes and is used in the following chapters as reference system.

Motivated by a label analysis of ground truth data, which reveals that most cars are occluded by other cars, Chapter 4 first presents promising results for occlusion-handling. The framework features separate classifiers for unoccluded and occluded cars and takes their mutual response characteristic into account. With this, the car-car relation is used to handle occlusion.

In order to handle general occlusion constellations, Chapter 5 discusses a

framework that focuses on a more general concept to handle vertical occlusion patterns. It also describes a two-stage classifier architecture that detects vertical car parts in the first stage and combines the local response in the second stage. As an extension, depth information is provided for the individual car parts, helping the classifier to reason about typical occlusion patterns.

A fundamental problem for the development and analysis of occlusion-handling strategies is that occlusion information can not be labeled accurately enough in real-world video streams. For this reason, Chapter 6 illustrates a rendered car detection benchmark data set. This data set provides pixel-level information of object instances, which is instrumental for the evaluation of occlusion-handling strategies.

Because of the limitations of a holistic discriminative approach to dealing with untrained occluded object views, a switch to a parts-based car detector is motivated in Chapter 7. This chapter presents an occlusion-handling strategy for such a parts- based detection approach. Thereby, this uses the mask information of the occluding object, taking into account knowledge about the visibility of features. In particular, the limitations and optimal parameter settings of this framework are indicated. These findings and the evaluation results motivate a later proposal of an occlusion-handling strategy that is especially helpful for strongly occluded views.

Another key step of each classifier is the feature extraction. Chapter 8 demonstrates that the used feature descriptor outperforms other state-of-the-art ones. For this, an analysis of the stability under rotation and illumination changes of two state-of-the-art feature descriptors is presented.

Chapter 9 presents a psychophysical study with human participants. The study evaluates which information humans use to adapt to the absence of features by recording eye movements with an eye tracker.

Chapter 10 concludes the thesis and briefly evaluates the results of the previous chapters.

1.2 Publications in the Context of this Thesis

The following articles have been published in the context of this thesis:

Conference articles

[C15a] M. Struwe, S. Hasler, and U. Bauer-Wersing. Rendered Benchmark Data Set for Evaluation of Occlusion-handling Strategies of a Parts-based Car Detector. In *Pacific-Rim Symposium on Image and Video Technology 2015*, pages 99–110, 2015.

[C14a] M. Struwe, S. Hasler, and U. Bauer-Wersing. A Two-stage Classifier Architecture for Detecting Objects under Real-world Occlusion Patterns. In *International Conference on Artificial Neural Networks 2014*, pages 411–418, 2014.

[C13b] M. Struwe, S. Hasler, and U. Bauer-Wersing. Using the Analytic Feature Framework for the Detection of Occluded Objects. In *International Conference on Artificial Neural Networks 2013*, pages 603–610, 2013.

[C13a] M. Struwe, S. Hasler, and U. Bauer-Wersing. Combining Multiple Classifiers and Context Information for Detecting Objects under Real-world Occlusion Patterns. In *New Challenges in Neural Computation Workshop 2013 at the German Conference on Pattern Recognition*, pages 35–42, 2013.

2 Related Work

Chapter overview *This chapter reviews some promising directions for the detection of occluded objects. It discusses different state-of-the-art approaches and groups them by taking into account their classification methods.*

2.1 Introduction

This chapter distinguishes between several main directions that have been found during the literature review: object representation, object relation, global context, and motion features.

2.2 Object Representation

The object representation is the way an object is stored in memory and matched for recognition. There are numerous object representations and each deals with occlusion differently. Here, there is a detailed focus on two methods: parts-based and holistic discriminative approaches. Both approaches employ a spatial representation whereby those that are holistic use a fixed template of the learned object, while parts-based approaches make use of sets of features.

2.2.1 Holistic Discriminative Approaches

Holistic discriminative approaches form a substantial part of prior work on object representations for recognition. Holistic means that a fixed template of an object is learned during training. Additionally, some approaches learn to represent differences between positive and negative examples of a class with a template covering the whole object. For example, methods like [Dalal and Triggs, 2005] train a holistic object template in a discriminative manner and focus resources on differences between classes. This strong specialization on the training problem results in a more drastic decrease in performance for occluded objects when trained on unoccluded views.

Hasler et al. [2007] have computed SIFT [Lowe, 2004] descriptors and matched them to a set of analytic features. After a spatial MAX pooling, a single-layer perceptron (SLP) is trained on top of the features. The SLP uses negative and positive examples to calculate an optimal pattern of negative and positive weights. Hence, the discriminative classifier can more easily separate these classes, which improves the detection performance. However, this ability leads to diminished detection performance if the test set differs from the training set, which

is the case if the classifier is trained on unoccluded object views and tested on occluded views. The reason for this is that the simple SLP cannot compensate for missing occluded features that would make a positive contribution.

Heisele et al. [2001] have presented a similar approach, but instead of a large set of analytic features, they train few linear support vector machines (SVMs) to detect dedicated face parts, and on the top, one SVM that combines the part responses (Fig. 2.1). In this way, a holistic template is generated discriminatively.

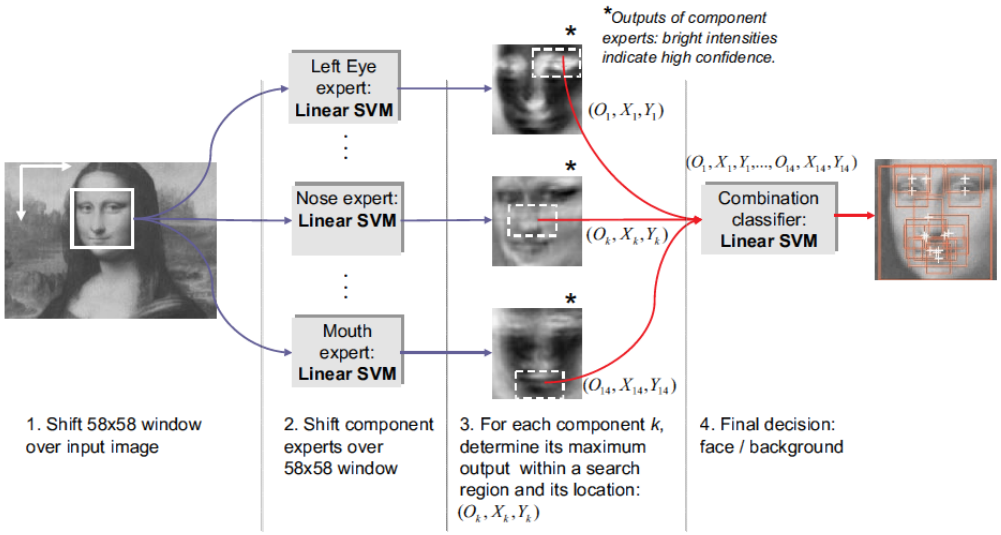


Figure 2.1: Holistic Approach by Heisele et al. [2001]: A linear SVM is trained for each face's part and used to generate a topographic response map. On top, a single SVM combines these maps to a "face" response.

Compared to parts-based approaches, which conduct a purely positive accumulation, holistic approaches can better learn how to separate positive and negative examples by using positive and negative weights, and usually exhibit a higher classification performance on the trained scenario.

However, the approaches do not generalize well to occluded test scenarios. This is because both the missing of object features and the accidental non-object features reduce the score of the object hypothesis (Fig. 2.2).

[Wersing and Körner, 2003] have also used a holistic approach. However, rather than applying discriminative training on top of a feature hierarchy, they have directly stored the training examples as templates (Fig. 2.3). During occlusion, this approach has advantages compared to [Hasler et al., 2007] and other discriminative approaches because there is only a positive contribution of features, like in parts-based voting methods. Nevertheless, for unchanged scenarios, ap-

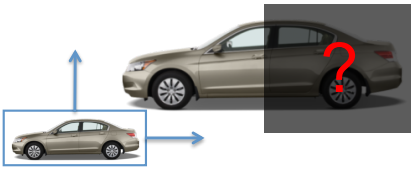


Figure 2.2: Occlusion-handling of Holistic Discriminative Approaches: Usually holistic discriminative approaches are trained with non-occluded cars. The holistic model has no explicit model of missing features.

proaches like [Hasler et al., 2007] demonstrate better detection performance than [Wersing and Körner, 2003].

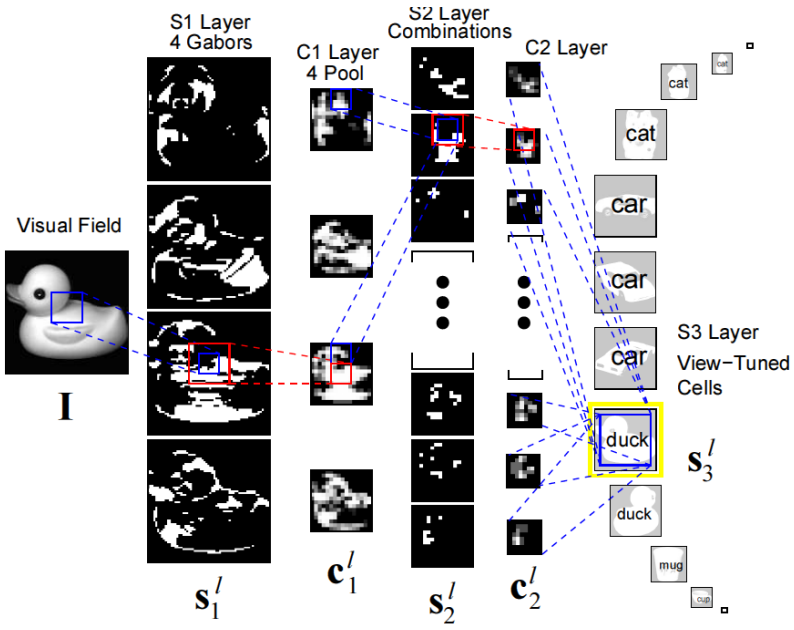


Figure 2.3: Holistic Approach by Wersing and Körner [2003]: The first feature layer S1 computes Gabor features. After a pooling stage, to reduce the resolution of the image in C1, the second feature layer S2 searches for combination of the features. C2 applies a pooling again. The resulting C2 feature activations are stored as templates in the last layer, S3.

2.2.2 Parts-Based Approaches

Parts-based approaches infer the position of an object from the position of detected parts. [Leibe et al., 2004] have used a so-called “code-book,” which is the result of a clustering step and where every feature is stored with the relative posi-

tion to the object's center. During training, an "interest point detector" defines the positions for the feature extraction. During detection, an interest point detector is used again, and for each point, the corresponding image region is matched to the code-book. The entry with the highest activation votes for possible positions of the object's center. To refine an initial hypothesis, [Leibe et al., 2004] have determined which interest points contribute to the local maximum and searched within a small vicinity for additional weakly activated features. In this way, they can increase the number of image regions contributing to the hypothesis (Fig. 2.4).

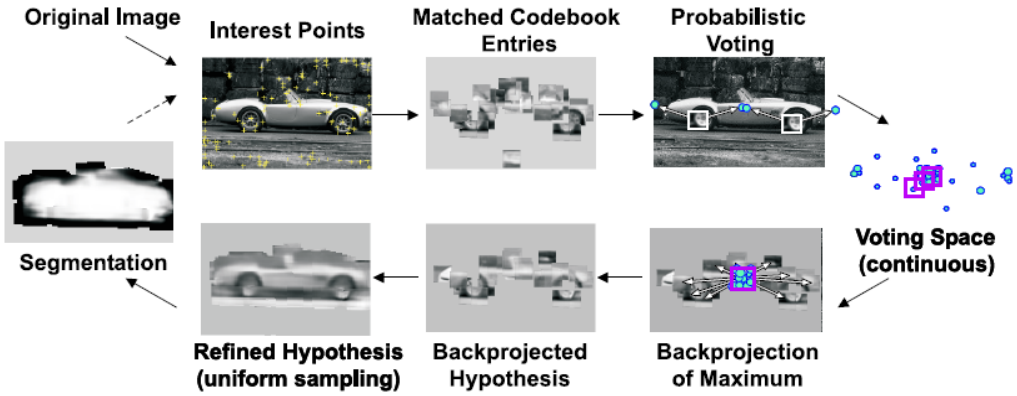


Figure 2.4: Parts-based Approach by Leibe et al. [2004]: An algorithm first selects some interest points. The features in a code-book are matched on these points, and the winners vote for the object's center. In the voting space, the maxima represent possible object locations. These maxima are used for a back-projection step to refine the object hypothesis.

A similar approach by Lowe [2004] has also determined interest points and used scale-invariant feature transform (SIFT) descriptors. In contrast with [Leibe et al., 2004], it uses no feature alphabet, but all SIFT descriptors of the training images are stored together with the relation to the object's center, the scale, and the planar orientation.

In general, the parts-based approaches accumulate the weak evidence of many detected features, as Fig. 2.5 illustrates. Still, this object representation is also affected by occlusion. The percentage of occlusion is usually proportional to the loss in score. Also, when trained with un-occluded views, these methods can handle arbitrary occlusion patterns, but require that a sufficient amount of features can still be detected. During strong occlusion, only certain parts of a car are visible, and therefore only a small score can be gained, which usually results in a false negative detection. However, because every feature votes individually in a purely positive manner and accidental features generate no negative contribution, parts-based approaches are less affected by arbitrary occlusion. However,

in general, the voting methods perform more poorly than those that are discriminative whenever test and training set does not show such systematic differences, as [Yi-Hsin et al., 2010] have discussed and the detection results in [Dollár et al., 2009] have confirmed.

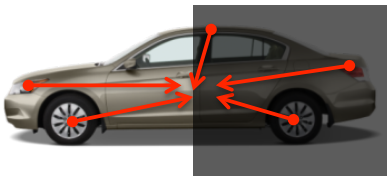
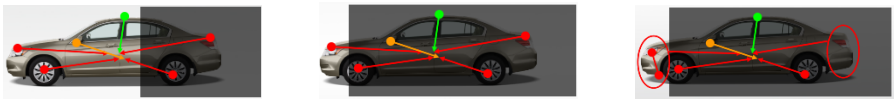


Figure 2.5: Parts-based Voting Approaches: Parts-based voting approaches accumulate the weak evidence of many detected features. The features are marked in red with their votes for the object's center. In general, the percentage of occlusion is proportional to the loss in score.

To improve the approaches for strong occlusion as well, "specialized features" (Fig. 2.6a) can be used. Accordingly, some kind of strength of the features has to be labeled. Features that are highly specific to cars, like floodlights, are strong features. Features that are also common among other objects, such as the edge of the roof, are weaker. By detecting only a few specific features with high confidence, it can be assumed that the detected features belong to a strongly occluded car (Fig. 2.6b). During strong occlusion, only small parts of a car are visible. In such cases, information from these visible parts must be used as much as possible. Increasing the feature's size also increases the contribution of the feature's (Fig. 2.6c) visible parts. Moreover, the relative position from one feature to a neighboring feature can be exploited more strongly.



(a) **(b)** **(c)**

Figure 2.6: Weak and Strong Features: (a) Features are labeled with different strength for each feature separately. Green arrows represent weak and non- specific features. Orange arrows represent more specific features, and red arrows represent highly specific features. (b) Strong occlusion results in the loss of features. (c) Making use of the relative position of neighboring features.

[Collet et al., 2011] have presented an interesting approach in this direction. First, nearby features are clustered together into object parts. Afterwards, the relation of parts is exploited in order to group them to objects. The first step puts

more emphasis on local relations; with this, the approach might be less affected by occlusion (Fig. 2.7).

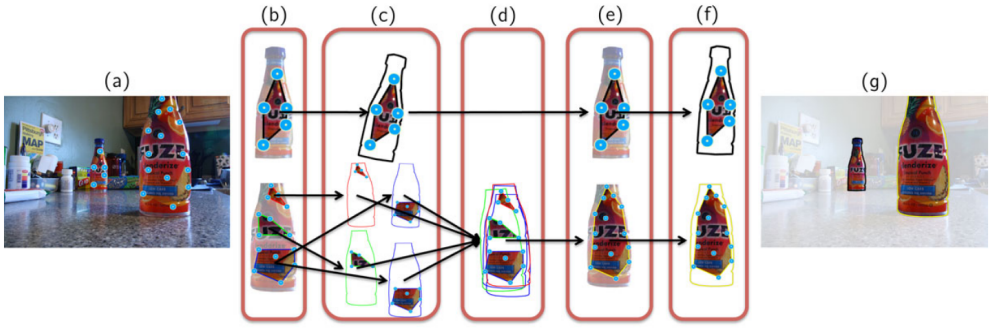


Figure 2.7: Parts-based Approach by Collet et al. [2011]: First, nearby features are clustered together into object parts. Then, the relation of parts is exploited to group them to objects. The first step puts more emphasis on local relations; thus, the approach might be less affected by occlusion.

One drawback of the parts-based methods discussed in this section is their dependency on the interest point detector. This can result in a missed detection if too few interest points or interest points that are not the same as in the trained example are found. Chapter 7 details how to use a regular grid to define the position of the interest points in order to fix this.

2.3 Object Relation

As the previous section has described, the object representation uses the information of a single object. It can be beneficial for the recognition to also utilize the relations between objects in a scene, e.g. the relative position of two objects or a segmentation mask of objects. The segmentation and object relation can help to differentiate between objects in the foreground and the background, and consequently to handle the problem of occlusion during detection. This section elaborates upon some approaches that can use this information to improve the detection performance.

The previous section has illustrated the parts-based object representation approach by Leibe et al. [2004]. In [Leibe and Schiele, 2006], the authors have described how the method can be applied to generate a segmentation mask by using a back projection step to refine the object hypothesis. In further steps, this segmentation mask could be utilized to reason about occluded objects.

The approach by Torralba et al. [2004] optimizes the detection performance of

occluded objects by using an object-object relation. [Torralba et al., 2004] have combined object detection and segmentation by labeling every pixel in the image. The algorithm learns to first recognize large and easy-to-detect objects. The information from the detected objects is then used to locate hard-to-detect objects. For example, in an office scene, computer screens can be easily detected. The vicinity of these detections is used to find objects with more challenging conditions, such as a keyboard or mouse (Fig. 2.8). This approach can improve the detection performance on occluded objects by first searching for the easily detectable non-occluded objects, and then searching for less detectable occluded objects in the vicinity. Chapter 5 demonstrates an implementation of object-object relation by utilizing [Hasler et al., 2007] as initial system. To address more general occlusion, Chapter 5 determines a later split of the holistic car template.

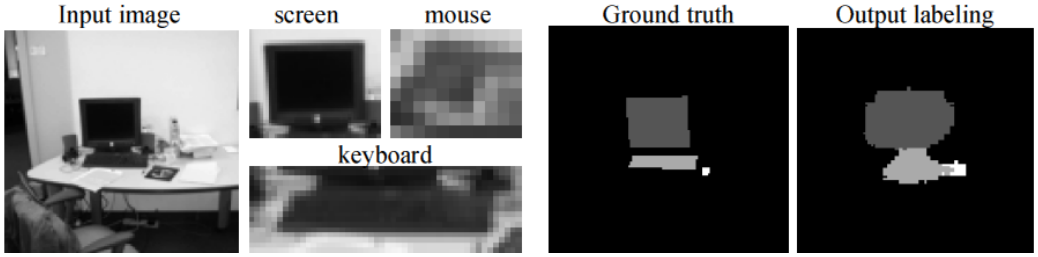


Figure 2.8: Use of Object-Object Relation by Torralba et al. [2004]: Torralba et al. [2004] first searched for easily detectable objects, and then searched in the vicinity for objects with more challenging conditions. In the example, the monitor is detected first. After this step, the approach searches for more difficult-to-detect objects in the surroundings of the detected monitor (in this example, the mouse and the keyboard) and generate the output labeling.

The detection approach by Gao et al. [2011] learns about typical occlusion constellations. Therefore, bounding boxes are sub-divided into a number of cells, and the approach determines for each cell if the underlying pixels belong to the object (Fig. 2.9). The activation of the cell can enable the generation of a depth ordering and the identification of typical occlusion patterns, e.g. pedestrians are usually occluded by an object at the bottom, not at the top.

In [Winn and Shotton, 2006], a layout consistent random field is used to infer which object class and which instance are present at a certain pixel. For each instance, they use a hidden random field to mark parts as belonging or not belonging to the object. So, neighboring pixels with labels that are not layout consistent are not part of the same object. Winn and Shotton [2006] have evidenced that recognizing several parts can help detect objects under partial occlusion. Local spatial interaction between parts is used to make a hypothesis where neighboring pixels can be found. Long-range spatial dependencies are used to find the rest

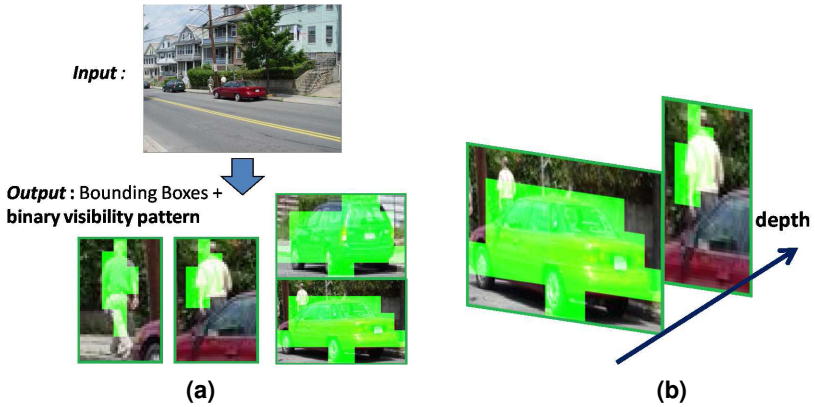


Figure 2.9: Implicit Occlusion-handling by Gao et al. [2011]: (a) Some cropped bounding boxes of detected objects are shown. Green cells mark pixels belonging to the object centered in the bounding box. (b) Example for the depth ordering.

of the object.

The approaches of [Winn and Shotton, 2006] and [Gao et al., 2011] can both manage a low percentage of occlusion and provide a kind of mask information through pixel-wise labeling. The drawback is that they make a parallel full-image search and use a time-consuming iterative approach. Also, they cannot manage moderately to strongly occluded object views.

[Makris et al., 2013] has demonstrated a parts-based detection approach that makes use of object relations and mask information. Like in [Leibe et al., 2004], this approach builds a code-book during training. During testing, the input image is filtered with the stereo signal, as Fig. 2.11a illustrates. As a result, a disparity map is calculated, and pixels that cannot belong to a car object are filtered out (e.g. road surface, sky, buildings, etc.). In this filtered image, intensity and depth information are fused for the code-book matching step. After an accumulation step, a re-weighting of the score from the un-occluded regions of a car hypothesis is performed by taking into account the rate of occlusion. For the occlusion-handling, a possible detection is sub-divided into 10 patches. The sub-patches can be considered either visible, occluded considering stereo depth information, occluded considering another detection nearer to the sensor, or occluded considering both of the above (Fig. 2.11b). With this, they generate a kind of mask of the occluded regions. After this, the evidence of a detection is multiplied by the reciprocal of the predicted occlusion in order to normalize its evidence with respect to a detection that is fully visible.

Inspired by this concept, Chapter 7 depicts an extended occlusion-handling, which makes also use of mask information. Furthermore, it presents an improved occlusion-handling, which takes the contribution of the features of the learned car

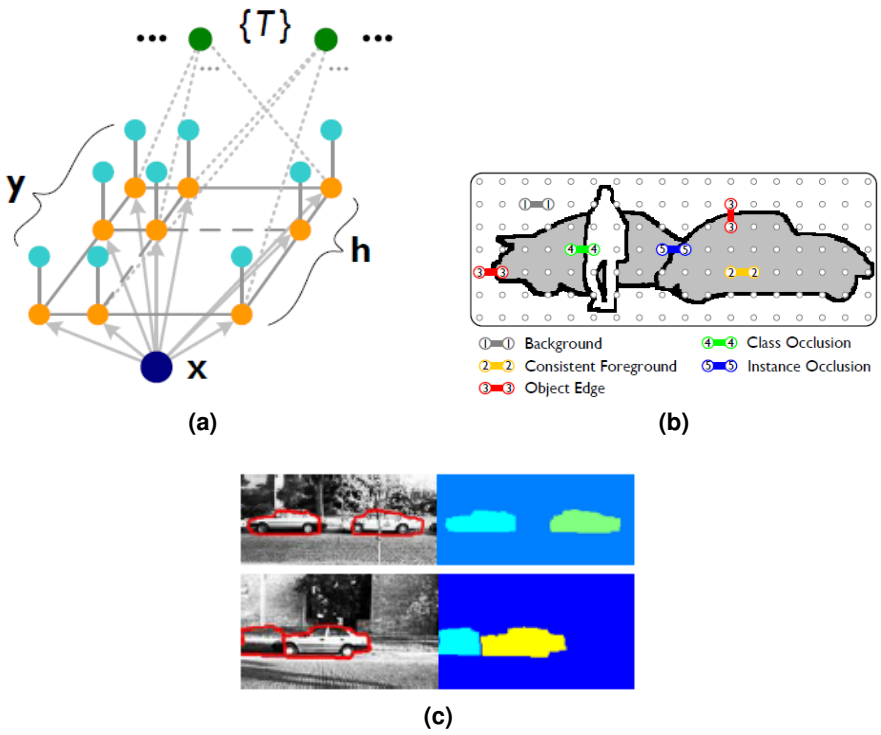


Figure 2.10: Implicit Occlusion-handling by Winn and Shotton [2006]: (a) A hidden random field is used to infer local presence of certain objects x. h is used for the part labels, y for the instance labels, and T for the instance transitions. (b) illustrates the types of transitions between objects. (c) shows the result of the detection approach for a street scene.

model more precisely into account.

[Hoiem et al., 2007] have also used an iterative segmentation approach to extract the occlusion boundaries and depth ordering. In comparison to [Gao et al., 2011] and [Winn and Shotton, 2006], their approach is based on three-dimensional (3D) information and learns to identify boundaries based on a wide variety of cues, such as color, position, surface orientation, and depth estimation (Fig. 2.12).

In general, occlusion is related to the 3D relation of objects. A common cue of 3D information is depth, which can be used to check the physical plausibility of an object's position and size [Gould et al., 2008].

Other strategies make use of 3D annotated data of car views. A popular strategy is the use of the deformable part model (DPM) [Felzenszwalb et al., 2010]. In [Pepik et al., 2013], the 3D annotated data of the KITTI dataset [Geiger et al., 2012] is used to generate bounding boxes of the occluder, the occluding object,

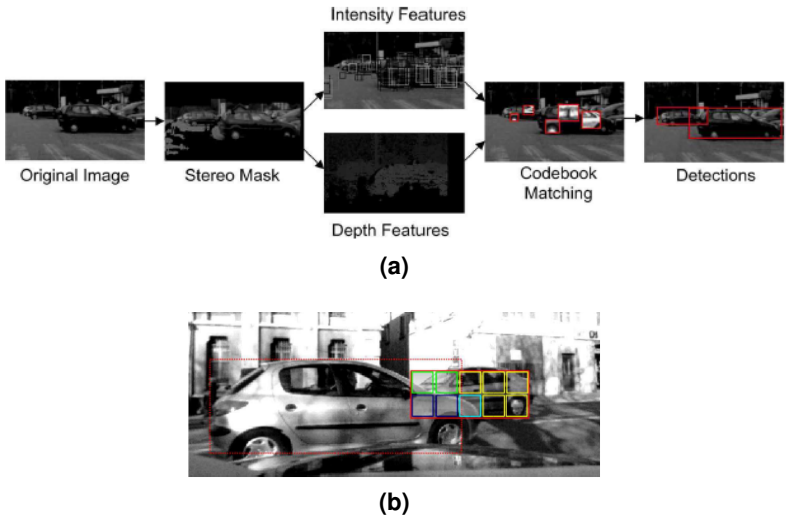


Figure 2.11: Explicit Occlusion-handling of Parts-based Approach by Makris et al. [2013]: (a) The authors first filter out irrelevant regions. Then, they fuse the intensity and depth information before the code-book matching. At the end, possible car hypotheses are computed. (b) An occlusion-handling is included at the detection step. The patch of a detection is sub-divided into ten sub-patches to estimate the rate of occlusion for a later re-weighting of the activation score.

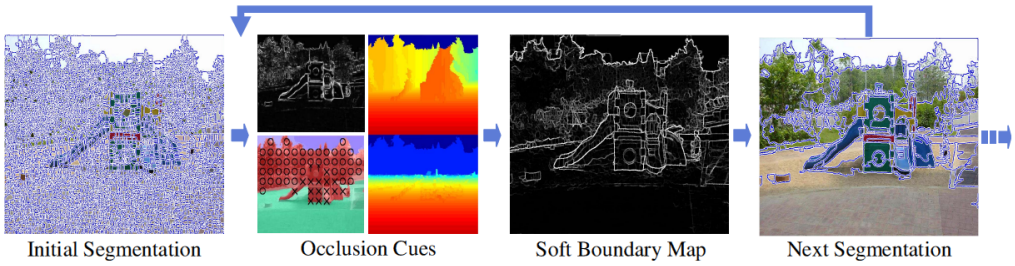


Figure 2.12: Iterative Segmentation by Hoiem et al. [2007]: First an initial segmentation is determined by using 3D information. After this occlusion cues help to determine a soft boundary map which is used for a refined segmentation. This refinement is usually repeated several times.

and their union. For each of the three types, a separate DPM is trained. In contrast to this, a system that does not need any labeled information of the occluder during training is presented in Chapter 5. In [Zia et al., 2013], the authors have used hand-annotated 3D computer-aided design (CAD) models and generated part models in addition to the full car view. A single component DPM detector is trained for each part configuration. To handle occlusion, 288 artificial occluder masks are generated for the training data. The approach does not work in real

time, and can handle only occlusion cases that somehow match with the generated occlusion masks. Chapter 5 proposes a way to use the occlusion pattern in real-world scenes without generating artificial masks. Here, the training data is used to learn typical occlusion constellations.

2.4 Global Context

The approaches described in the previous section make use of object relations to improve the detection or recognition of an object. This section focuses on methods that exploit the global or unspecific context in a scene.

Many approaches use physical size estimation and ground plane assumption to exclude some false positives (FPs). For example, if the classifier detects a car directly in front of the camera, it is possible to use the physical size estimation to check the size in relation to the distance. If the object is too small for a car at the calculated distance, it can be excluded and does not cause an FP.

There are also approaches that contend with the problem of occlusion by using some global context. Vedaldi and Zisserman [2009] have analyzed how to handle occlusion by the image border (Fig. 2.13). Mask information represents the image border at the detection step. Because occlusions caused by truncation by the image border are numerous, this approach can substantially improve the recognition. This thesis uses more general global context to find more candidates for occluded objects.

Figure 2.13: Occlusion-handling at the Image Border by Vedaldi and Zisserman [2009]: Vedaldi and Zisserman [2009] model occlusion at the image borders. Occlusion often appears by truncation from the image border.

2.5 Motion Features

This thesis focuses on single-view based object recognition models. Nevertheless, this section gives an example of a method that makes use of so-called "motion features." Motion features are features that include temporal information.

One approach by Liu et al. [2009] uses a contour-motion feature (CMF) to simulate a type of short-time memory called space-time volume (STV). Fig. 2.14 illustrates the approach in detail. First, an edge detection and distance transform is applied to the STV to generate the distance transform volume (DTV), which is sensitive to motion. Haar-like filters extract the features. This approach improves the detection or recognition of pedestrians, but is not robust enough to handle the problem of occlusion.

In general, motion features can detect some fast moving parts that a classifier cannot, and can also refine the segmentation of objects. This thesis focuses on object recognition on single images and does not use such information for the presented detection approaches.

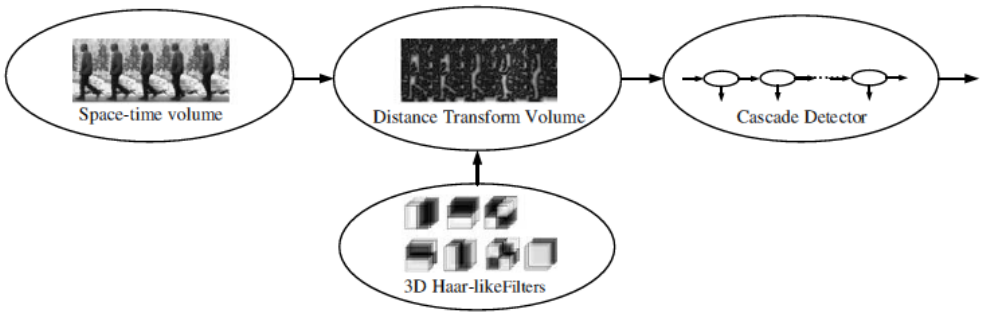


Figure 2.14: Overview of the Approach by Liu et al. [2009]: A kind of short-time memory, called space-time volume, builds the distance transform volume, which is sensitive to motion. Haar-like filters extract features.

Additionally, tracking is not taken into account. However, for later developments, it is possible to integrate a tracking at the post-processing, which might improve the detection performance to recognize occluded objects what is already mentioned by approaches like [Pourtaherian et al., 2013, Schmaltz et al., 2007, van Gastel et al., 2015].

3 Analytic Feature Framework

Chapter overview *This chapter details the adaption of the analytic feature framework ([Hasler et al., 2009]) for object detection in complex traffic scenes. Originally, the framework was proposed for the large-scale identification of segmented objects. Besides the necessary adaptations, it also presents the result of an evaluation of the competitiveness of the framework on different real-world datasets. Additionally, it provides a numerical analysis of typical occlusion cases for a car detection task, as well as some examples.*

Parts of this chapter are based on:

[C13b] M. Struwe, S. Hasler, and U. Bauer-Wersing. Using the Analytic Feature Framework for the Detection of Occluded Objects. *ICANN*, pages 603–610, 2013.

3.1 Introduction

Currently, the best detection approaches usually extract unspecific local features and apply a powerful classifier directly on top. For example, the combination of histograms of oriented gradients (HOG) [Dalal and Triggs, 2005] with an SVM is reported to yield reliable performance in various detection benchmarks. In contrast with [Dalal and Triggs, 2005], there are methods that strive to learn a more problem-specific feature representation, wherein a very simple classifier can be used on top for discrimination. An example of such a method is the analytic feature architecture proposed in [Hasler et al., 2009], which evidences high performance for large-scale identification of segmented object views but has not been applied to detection tasks at full scenes. This chapter demonstrates that such an architecture can also provide competitive results in detection tasks. Sec. 3.2 outlines the adaptation of the analytic feature framework for detection tasks. After briefly describing the traffic scene data used in this study, Sec. 3.5 presents an evaluation of the performance of a car detection task. For this purpose, the focus is on the detection performance during occlusion. The adapted system is referred to in the following as the reference system.

3.2 Analytic Feature Framework

The appearance-based detector is based on the real-time object identification framework in [Hasler et al., 2009], which uses an attention mechanism to generate size-normalized segments of the input object. First, over the gray-scale segment, SIFT descriptors [Lowe, 2004] are computed on a regular grid. Each descriptor is then matched to a set of 421 analytic features, which are the result of the supervised selection process proposed in [Hasler et al., 2009]. Second, for

each feature, the global maximum is computed over the segment, thus removing all spatial information. Finally, an SLP (according to [Minsky and Papert, 1969, Rosenblatt, 1958]) separates the 126 objects in the 421-dimensional space. This approach works robustly for full 3D rotation, even for untextured objects, which are notoriously difficult for the standard SIFT approach [Lowe, 2004].

3.3 Adaptation of the Analytic Feature Framework to Car Detection

The application of the existing framework requires several adaptations for full-scene object detection. The rotation normalization of the SIFT descriptors is switched off because cars usually occur only upright. To expedite processing, only 96 analytic features are used, where only features of the car class and not the background class were selected (see some examples in Fig. 3.1b). On the top layer, the local SLP template is shifted over the input image. This convolutional step generates the car response map producing broad activation blobs for a car. Fig. 3.1a illustrates the resulting feature architecture.

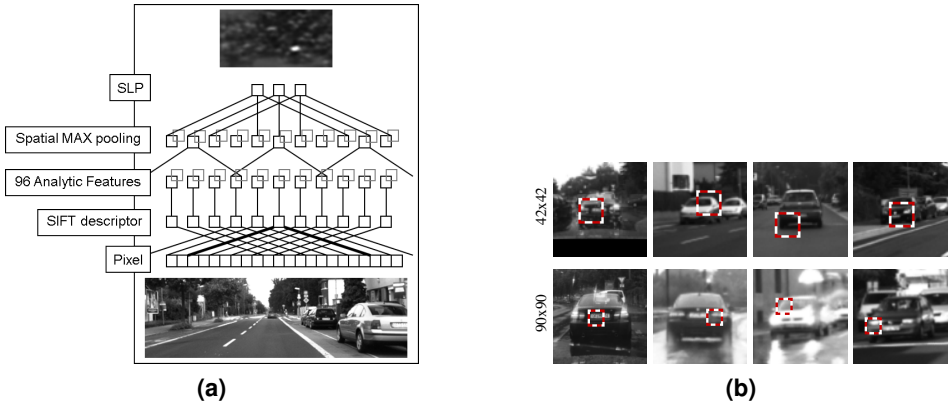


Figure 3.1: Analytic Feature Hierarchy: (a) SIFT descriptors are computed on a regular grid and matched to 96 analytic features. After a local maximum filter per feature, the SLP templates are used in a convolutional step. Maxima in the final response map denote possible car locations. (b) Some analytic features for two template sizes.

To process cars recorded at different distances, analytic features and SLPs on three different segment sizes are trained and used on the largest image resolution, while the largest template is also used on successively reduced resolutions. This combined strategy improves detection performance because no compromise between the minimal template size and most discriminative template size

needs to be found, which is a common drawback of other detection approaches.

3.4 Datasets

The framework that the previous section has proposed supports locating cars in real-world traffic scenes. To evaluate the detection algorithms, a car was equipped with a stereo camera, and different streams were acquired with a total length of 45 minutes. The streams covered various weather conditions, such as sunny, rainy, and overcast weather, as well as several scene types, such as city, rural, industry, and highway. For each first frame per second, typical traffic participants were labeled with a region of interest (ROI) and a roughly estimated percentage of occlusion. Fig. 3.2 displays an example of a labeled street scene. The dataset and the annotations were already generated by other engineers and only some slight corrections for some annotations were necessary for the proposed usage.

Figure 3.2: Example Image of the Annotated Video Stream: Street scenes are hand-annotated by using 2D bounding boxes for important objects or areas. A label tool labels objects such as cars, obstacles, pedestrians, road terrain, etc. For simplicity, the example image only shows the car objects. Some detailed information, such as the rate of occlusion, is also stored for each object.

To evaluate the car detection performance, the image streams were divided into 30-second segments. The odd segments were used for the training of analytic features and SLP templates, and the even segments for testing. Car ROIs with a width greater than twice the height (roughly 10% of the data) were excluded from training. In this way, the use of smaller and squared SLP templates was sufficient. Thus, the templates were trained on segments of 42×42, 66×66, and 90×90 pixels (plus 18 pixels border at each side), depicting the car in the

center. Initially, the SLPs were trained to separate a few thousand segments of un-occluded cars from a larger set of randomly chosen non-car segments. During five bootstrapping [1] steps, more negative examples were generated on the training data.

3.5 Detection Results

Please note that the disparity information available for the image data was used to reject implausible car candidates with simple hand-tuned heuristics on height above ground and physical size. Finally, a local competition removed further weak hypotheses if they overlapped too strongly with more confident ones. For the input images sized at 800×600, the graphics processing unit (GPU) implementation of the framework ran at 10 frames per second on a mobile Geforce GTX580M.

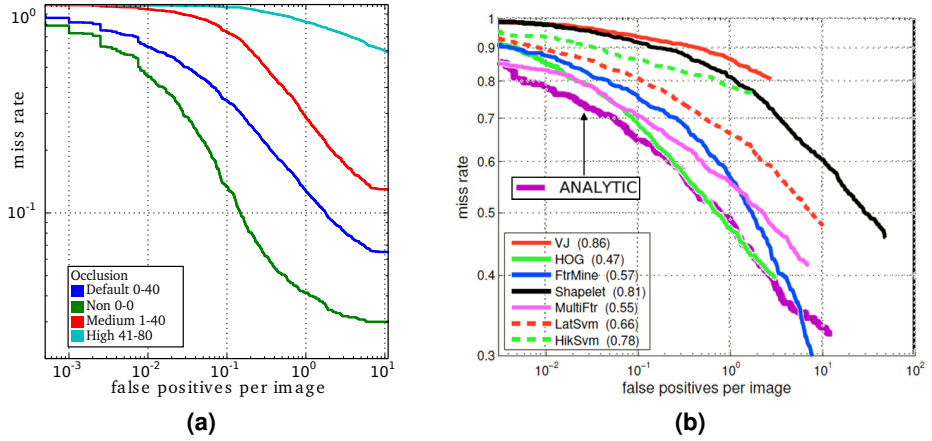


Figure 3.3: Detection Results: (a) Receiver operating characteristic (ROC) for the car detection scenario. The performance decreases strongly with the percentage of the car's occlusion. (b) ROC for pedestrian benchmark [Dollár et al., 2009] (un-occluded, 50 pixels or taller). The analytic approach is on par with state-of-the-art approaches.

The detection performance on the previously described car scenes is visible in Fig. 3.3a. The miss rate is plotted over the false positive per image (FPPI) rate. A miss rate of e.g. 10^{-1} means that 10% of the cars were not detected. The FPPI rate indicates the amount of false detections that are generated at the

[1] Bootstrapping is a method to improve the detection performance of a machine learning algorithm. First, the false positives of the classification output of a classifier are added to the training examples. After this the classifier is trained again. These steps can be repeated several times until the classifier shows acceptable or converged detection results.

corresponding recall value. The curves reveal a strong dependency between the car's occlusion and detection performance. For an FPPI of 10^{-1}, 70% of the cars with an occlusion between 0–40% are detected. This pure detection performance is usually sufficient for a system that applies some form of temporal integration (tracking). However, the recall drops severely for higher percentages of occlusion, which can no longer be compensated for at the system level. During testing, cars with a height below 35 pixels are excluded, and a 50% mutual overlap criterion between labels and detections is used.

In order to compare it with state-of-the-art methods, the reference system was applied to the pedestrian detection benchmark proposed in [Dollár et al., 2009]. To evaluate the performance, a six-fold cross-validation over the six streams in the dataset was completed and the results were averaged. In contrast with this, the competitors in Fig. 3.3b used all streams for testing and trained on other pedestrian data. So, the results are not 100% comparable. However, because the streams are significantly different from each other and the overall label quality is not particularly high, there is no obvious advantage to using the streams for training. Fig. 3.3b roughly demonstrates that the reference system is competitive with the popular HOG approach [Dalal and Triggs, 2005] for an FPPI greater than 10^{-1}, and more effective for a smaller FPPI. The main conceptual difference from HOG is that it applies an SVM directly on top of the local gradient histograms, while the reference system uses an additional projection to the analytic features and a simple SLP as classifier. Please note that only the mutual overlap heuristic was used here because of the missing stereo information.

To more thoroughly understand occlusion of cars in traffic scenes, the number of typical occluders and types of occlusion for ground truth data were counted. In total, the traffic scene included 15,514 labeled cars. Of these, 8,796 were occluded. The result in Tab. 3.1 reveals that almost all cars were occluded by other cars, but many cars are also occluded by the image border.

Fig. 3.4 offers some examples of occluded car views. The occluding objects are other cars, pedestrians, bicycles, obstacles, and the image border.

3.6 Conclusion

This chapter has presented the reference framework for this study. This framework is based on the analytic feature representation originally proposed for object identification, so the chapter has also described the necessary adaption of the framework. It has demonstrated the detection performance on a public pedestrian detection benchmark and another benchmark in order to evaluate how strong occlusion affects car detection performance. Furthermore, this chapter has pro-

Table 3.1: Label Analysis of Ground Truth Data: Counts of car occluders and occluded car parts for the ground truth data. In total, 8,796 of the 15,514 cars were occluded, most of them by other cars.

Occluding object	#
Another car	7,061
Image border	2,137
Motor bike	82
Pedestrian	70
Traffic sign	31
Other/non-labeled	1,125

Occluded part	#
Left	3,730
Right	3,124
Middle (only)	90

Figure 3.4: Examples of Occluded Car Views: Several occluded car views are shown with different types of occluding objects.

vided a numerical analysis of typical occlusion cases for a car detection task by using the presented ground truth data. The next chapter illustrates a car detector, which takes car-car occlusion constellation into account in order to improve detection performance on occluded car views. This approach is motivated by the fact that most cars are occluded by other cars.

4 Use of Object-Object Relations with a Holistic Discriminative Classifier to Handle Typical Occlusion Constellations

Chapter overview *This chapter applies an occlusion-handling strategy to the analytic feature framework. Similar to the current state-of-the-art strategies, the reference system shows a strong degradation of performance with increasing occlusion of objects. A brief discussion identifies possible steps to address this problem. Motivated by the fact that most cars are occluded by other cars, this chapter first presents promising results for a framework that uses separate classifiers for un-occluded and occluded cars, and it takes their mutual response characteristic into account. This training procedure is applicable to many other trainable detection approaches.*

Parts of this chapter are based on:

[C13b] M. Struwe, S. Hasler, and U. Bauer-Wersing. Using the Analytic Feature Framework for the Detection of Occluded Objects. *ICANN*, pages 603–610, 2013.

4.1 Introduction

The previous chapter has demonstrated the decreasing detection performance of the reference system during occlusion.

To exploit the benefits of discriminative approaches for occluded objects, one could simply train them with occluded and un-occluded views. However, this will likely reduce performance for un-occluded views during testing, which Section 5.2 discusses in detail. So, more advanced processing is necessary.

One possibility is to exploit the relation between occluding and occluded objects, which typically exhibits rather systematic patterns for natural scenes. The aforementioned detection approach by [Torralba et al., 2004] first searches for larger, more easily detectable objects, and later exploits spatial relations to improve the detection of smaller, more difficult objects. This concept can transfer to the occlusion problem. So, one could train special detectors for different types of occlusion and exploit their mutual response characteristics in a scene.

Instead of using a demanding iterative processing over the full scene, such as in the approaches by [Winn and Shotton, 2006] and [Gao et al., 2011], this chapter proposes a more directed search for occluded objects.

Additionally, convolutional neural architectures were recently applied with great success to current recognition [Ciresan et al., 2010] and segmentation benchmarks [Schulz and Behnke, 2012]. The problem of occlusion, however, was not actively treated in these models so far. This chapter proposes a particular training procedure for occluded and un-occluded detectors that can be applied for these architectures in a similar way.

Motivated by the label analysis in the previous chapter, the consideration of car-car constellation illustrates a possible way to improve the detection of occluded cars. Thereby, this chapter provides a first proof of concept on segmented car

images. Section 4.2 explains the architecture of the detection framework and details how the training and test data are prepared before reporting the training and the detection results. Sec. 4.3 concludes the chapter.

4.2 Occlusion-handling Using Object-Object Relations

This section proposes a method to increase detection performance for occluded cars. After the discussion in the introduction, the object-object relations are exploited. Taking into account that most cars are occluded by other cars, as the analysis of the ground truth shown in Tab. 3.1 has revealed, the following simple strategy is implemented: A special classifier on occluded cars is trained in order to avoid decreasing the performance for un-occluded cars, which would be the case if a single classifier for occluded and un-occluded car views was trained. The additional variation in the occluded training examples would confuse the representation of the un-occluded views. The new classifier, which is only trained on occluded car views, is applied only in the vicinity of already detected cars (Fig. 4.1). In a scene, the detection framework generates these initial car hypotheses, as Chapter 3 has described, using the classifiers trained on un-occluded cars. The conditional application rule is necessary to avoid a strong influx of FPs, which would be a result of the independent usage of both classifiers.

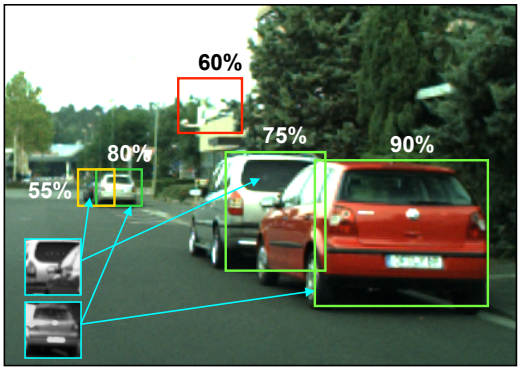


Figure 4.1: Two Individual Classifiers for Non-Occlusion and Strong Occlusion: The classifier for the detection of occluded objects searches only in the vicinity of already detected occluding objects. The exclusive use of the classifier trained for occluded cars generates too many FPs. The combination of both classifiers avoids this problem.

$\mathbf{F}_i$	$\mathbf{B}_i$	
 car	 car	**1. Car Occluding Car:** The occluding car is inserted as positive to $\mathbf{F}$, and the occluded car as positive to $\mathbf{B}$.
 car	 no car	**2. Car not Occluding Car:** The car is put as positive to $\mathbf{F}$, and a randomly chosen car-free region in its vicinity as negative to $\mathbf{B}$.
 no car	 no car	**3. Car-free Pairs:** In a real scene, the initial detector produces FPs. The FPs of the detection framework are inserted as negatives to $\mathbf{F}$, and a randomly chosen car-free region next to each FP as negative to $\mathbf{B}$.

Figure 4.2: Segment Pair Types: Each pair has a foreground segment $\mathbf{F}_i$ and a background segment $\mathbf{B}_i$. The positive examples (in gray) are generated from ground truth. For simplification, only samples with occlusion at the left side and mirrored examples with right occlusion are used to get more data. The classifier views the marked inner 42×42 pixel region of the segments into which the cars are fitted.

4.2.1 Datasets

For a fast proof of concept, this strategy was tested on segmented car and non-car views first. So, data pairs i were generated, each having a **F**oreground segment $\mathbf{F}_i$, containing the occluder, and the corresponding **B**ackground segment $\mathbf{B}_i$, containing an occluded item. The set of all foreground/background segments are referenced with $\mathbf{F} = \{\mathbf{F}_i\}$ and $\mathbf{B} = \{\mathbf{B}_i\}$, respectively. Fig. 4.2 describes the types of pairs that represent all possible constellations in a scene.

4.2.2 Training of the Car Classifier

For the segment scenario, the already trained SLP for a car size of 42×42 pixels was simply used as initial classifier, which is referred to as C_{Std}, and the new classifier C_{Occ} was trained on the background segments $\mathbf{B}$ of same size using cars with an occlusion up to 80%. The logic C_{Com} combines C_{Std} and C_{Occ} in a conditional manner and predicts the labels $L_{\mathbf{F}_i}$ and $L_{\mathbf{B}_i}$ for each pair using the following code:

$L_{\mathbf{F}_i}$ = 'no car'

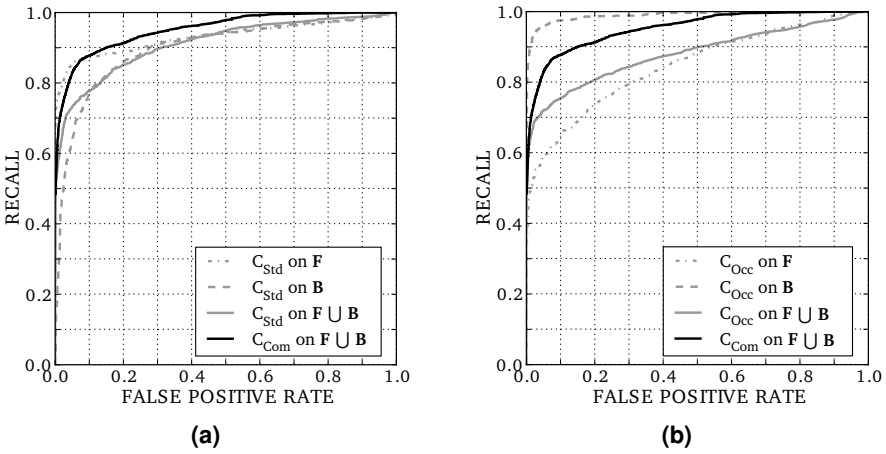


Figure 4.3: Comparison of C_{Com} with C_{Std} and C_{Occ}: (a) C_{Std} indicates a strong performance for the foreground segments $\mathbf{F}$, while the result for the occluded cars $\mathbf{B}$ is significantly weaker. On the combined data set $\mathbf{F} \cup \mathbf{B}$, C_{Com} is generally significantly better than C_{Std}. (b) C_{Occ} performs very well on $\mathbf{B}$, but exhibits strong problems on the unfamiliar occluders $\mathbf{F}$. C_{Com} also clearly outperforms C_{Occ} on $\mathbf{F} \cup \mathbf{B}$.

$L_{\mathbf{B}_i}$ = 'no car'
if $C_{\mathrm{Std}}(\mathbf{F}_i) \geq T_{\mathrm{Std}}$ then
  $L_{\mathbf{F}_i}$ = 'car'
  if $C_{\mathrm{Std}}(\mathbf{B}_i) \geq T_{\mathrm{Std}}$ then
    $L_{\mathbf{B}_i}$ = 'car'
  else if $C_{\mathrm{Occ}}(\mathbf{B}_i) \geq T_{\mathrm{Occ}}$ then
    $L_{\mathbf{B}_i}$ = 'car'

So, $\mathbf{B}_i$ is predicted as car if either C_{Std} or the new classifier C_{Occ} reaches its corresponding threshold, and only if a car was already found in the foreground segment $\mathbf{F}_i$.

4.2.3 Detection Results

To highlight the benefit of the combined logic, the performance of C_{Com} is compared with C_{Std} in Fig. 4.3a, and with C_{Occ} in Fig. 4.3b.

Please note that the combined approach depends on the two thresholds T_{Std} and T_{Occ}, and thus the performance of C_{Com} for each combination requires evaluation. T_{Std} was increased in 1,000 steps from zero to one, and for each T_{Std}, T_{Occ} was increased in 1,000 steps from zero to one. In the end, a set of 1,000,000 points in the ROC plot was generated. However, most of the resulting points in

the ROC curve are dominated by a small set of other points. Fig. 4.4 displays a plot with 10 exemplary values for T_{Std}. The amount of the points is reduced, and only a so-called "Pareto Front," which describes the set of optimal combinations, is shown in Fig. 4.3 (curve C_{Com}).

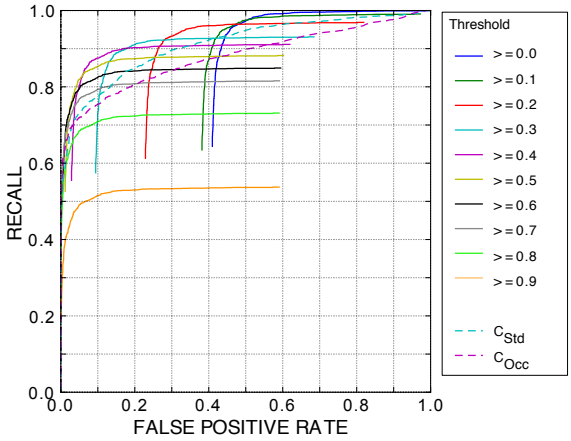


Figure 4.4: Two Individual Thresholds: This image illustrates the detection performance for 10 unique values for T_{Std} as well as the detection performance of C_{Std} and C_{Occ} on $\mathbf{F} \cup \mathbf{B}$.

Figure 4.3a confirms again that C_{Std} can cope substantially better with the familiar foreground segments $\mathbf{F}$ than with the occluded segments in $\mathbf{B}$. The classification on the combined data set $\mathbf{F} \cup \mathbf{B}$ has somewhat an intermediate quality, but is clearly dominated by C_{Com}. For example, at a recall of 0.8, C_{Std} has an FP rate of 0.13, while that of the combined curve is 0.04. This is a threefold reduction in the number of FPs.

In Figure 4.3b, C_{Occ} indicates an excellent detection performance on the occluded segments $\mathbf{B}$, for which it was trained. However, the performance for the un-occluded cars in $\mathbf{F}$ is much worse. One reason for this might be that C_{Occ} specialized too strongly on the edge caused by the occluder, which was not present in the un-occluded examples. Also, in comparison to C_{Occ}, C_{Com} evidences a substantially improved performance on the full data ensemble. In view of this, the combined approach clearly outperforms the reference system.

A low FP rate of 0.1 is chosen to avoid disturbing false alarms, which accompanies a recall of 0.84. For this FP rate, T_{Std} is set to 0.37 and T_{Occ} to 0.34. Fig. 4.5 presents some classification examples.

$\mathbf{F}_i$						
Ground truth	car	car	car	no car	car	no car
$C_{\mathrm{Std}} \geq T_{\mathrm{Std}}$	yes	yes	no	no	yes	yes
C_{Com} result	TP	TP	FN	TN	TP	FP
$\mathbf{B}_i$						
Ground truth	car	car	car	no car	no car	no car
$C_{\mathrm{Std}} \geq T_{\mathrm{Std}}$	yes	no	no	no	no	yes
$C_{\mathrm{Occ}} \geq T_{\mathrm{Occ}}$	yes	yes	yes	yes	yes	no
C_{Com} result	TP	TP	FN	TN	FP	FP

TP	-	true positive
TN	-	true negative
FP	-	false positive
FN	-	false negative

Figure 4.5: Pair Classification Examples: For each foreground sample $\mathbf{F}_i$ the ground truth label, the decision of C_{Std}, and the result of the combined approach C_{Com} are reported. For $\mathbf{B}_i$ additionally, the decision of C_{Occ} is included because C_{Com} depends on both classifiers and on the result for $\mathbf{F}_i$. Dark gray is used for "no car" labels and responses below threshold, while light gray is used for the opposite. The conditional logic can correct FPs of C_{Occ} (4^{th} column), but in rare cases also prevents correct detections (3^{rd} column). The classifiers look at the marked inner 42×42 region.

4.3 Conclusion

Motivated by an analysis of typical occlusion cases, this chapter has presented a new combination of detectors that takes into account the occlusion of cars by other cars. It has also revealed that the system clearly outperforms the original reference system. However, the approach was specifically trained for the explained car-car constellations and suffers from the same drawback as the holistic discriminative framework if the test scenario is different from the trained scenario (as already mentioned in Sec. 2.2.1). This means that the system exhibits improved detection performance only for car-car constellations, and will likely fail for other occlusion constellations. In general, the concept can transfer to other constellations, as well as to other detection tasks. Because it is difficult to anticipate all possible occlusion constellations at the training step, the goal in the next chapter is to identify a more general approach to handle more variable object constellations.

5 Split of the Holistic Car Model

Chapter overview *This chapter investigates a strategy to improve the detection performance of occluded objects based on the analytic feature framework presented in Chapter 3.3, and compares the results in a car detection task. Motivated by an analysis of annotated traffic scenes, the focus is on a general concept to handle vertical occlusion patterns. For this, the chapter describes a two-stage classifier architecture that detects vertical car parts in the first stage and combines the local responses in the second. As an extension, it provides depth information for the individual car parts helping the classifier in the second stage to reason about typical occlusion patterns.*

Parts of this chapter are based on:

[C14a] M. Struwe, S. Hasler, and U. Bauer-Wersing. A Two-stage Classifier Architecture for Detecting Objects under Real-world Occlusion Patterns. *ICANN*, pages 411–418, 2014.

5.1 Introduction

The previous chapter demonstrated an improved detection performance by taking car-car constellations into account. However, it has also explained that the framework evidences improved performance for these car-car constellations, but fails for others. This chapter concerns more general car-object constellations.

Occlusion is linked to the 3D relation of objects. A general cue of 3D information is depth, which can be used to check the physical plausibility of an object's position and size [Bo et al., 2014, Gould et al., 2008] or to segment a scene and direct attention to individual scene elements [Caron et al., 2014, Stückler and Behnke, 2013].

As mentioned, other strategies incorporate 3D annotated data of car views. A common strategy is the use of the deformable part model (DPM) [Felzenszwalb et al., 2013, 2010, Pepik et al., 2013, Yan et al., 2014, Zia et al., 2013]. However, all these approaches make use of annotated information of the occluding object or use some artificially generated occluder masks. This chapter details a way to handle occlusion without using any labeled information of the occluder. With this, each type of car-object constellation can be taken into account. Instead of using artificial occluder masks, it presents the use of occlusion patterns in real-world scenes.

Chapter 4 has illustrated a framework for detecting front and back views of cars in real-world traffic scenes. For this, image streams were taken under various weather conditions (sunny, rainy, overcast) and in different scene types (city, rural, industry, highway), and contained cars under all levels of occlusion. The final SLP car template was trained on un-occluded views only. This reference system is referred to as C_{Std} throughout the chapter.

Sec. 5.2 motivates a two-stage architecture for vertical occlusion-handling where the responses of discriminative vertical part detectors are integrated in a second stage. Finally, Sec. 5.3 identifies a means of exploiting depth information in the two-stage architecture before Sec. 5.5 presents conclusions.

5.2 Split of the Holistic Car Template

Inspired by the result in Tab. 3.1, Chapter 4 has exploited car-car occlusion. An additional classifier C_{Occ} on occluded cars was trained and was applied only in the vicinity of cars already detected by C_{Std}. The segment dataset used in Chapter 4 was simple in its normalization of the position and size of the occluded car, whereas in a real scene, a strong variance can be expected relative to the position and size of the un-occluded car. Because the concept focused on car-car occlusion, it neglected general occluders. Thus, the following chapter proposes other new strategies to contend with a variation of occluder types.

Tab. 3.1 illustrates that most cars are occluded on either the right or left side. This vertical occlusion is due to other cars, unlabeled walls, or the image border, and results in a mismatch of the holistic car template used in C_{Std}. In order to improve the detection of this type of occlusion, the following strategy is used: The holistic classifier is sub-divided into three vertical parts, and each part-classifier is trained with un-occluded car views. With this, each classifier is forced to make a more local decision about the presence of the car, and is later not affected by occlusion of a different part. To integrate the responses of the part-classifiers, their confidence values serve as input for an additional SLP, which is trained with cars with occlusion rates of 0–80%. Fig. 5.1b depicts the resulting two-stage architecture, which is referred to as C_{3Split}. The structure is equivalent to a multilayer perceptron [1] (MLP), but instead of back-propagation learning for each stage, it uses a different training set. In this way, the second stage is forced to deal with occlusion in a more symbolic way.

The comparison of C_{Std} and C_{3Split} in Fig. 5.2a indicates an improved detection performance for occlusion rates of 1–40%, while there is no gain for occlusion rates of 41–80%. A possible reason for the strong gain for un-occluded cars might be that each vertical part template of C_{3Split} is forced to make more effectively use of its local information, and thus finds a more general car concept. An analysis of the weights learned by the holistic SLP C_{Std} reveals a rather sparse contribution

[1] A multilayer perceptron is a feed-forward artificial neural network. The networks consist of multiple layers of computational units, usually interconnected in a feed-forward way. The most popular training technique for an MLP is back-propagation. In contrast to an SLP, an MLP is capable of solving the XOR-problem.

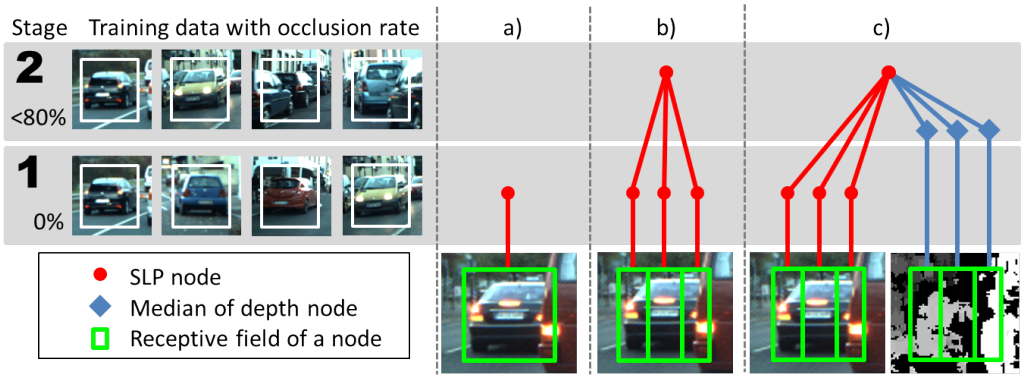


Figure 5.1: Different Detection Architectures: (a) Architecture of C_{Std}: A holistic SLP template is trained on non-occluded car views. (b) Architecture of C_{3Split}: The three vertically sub-divided SLP templates are trained on the same examples as (a). In the second stage, a separate SLP learns to combine the three confidence values of the first stage using non-occluded and occluded car views. (c) Architecture of $C_{\text{3SplitDepth}}$: The SLP on the second stage uses the quadratic combination of median depth for each vertical car part and the three confidence values of the first stage. Thereby, non-occluded and occluded car views are used.

of a small set of analytic features at specific locations, while each part-classifier of C_{3Split} integrates all features at all positions in a much broader manner.

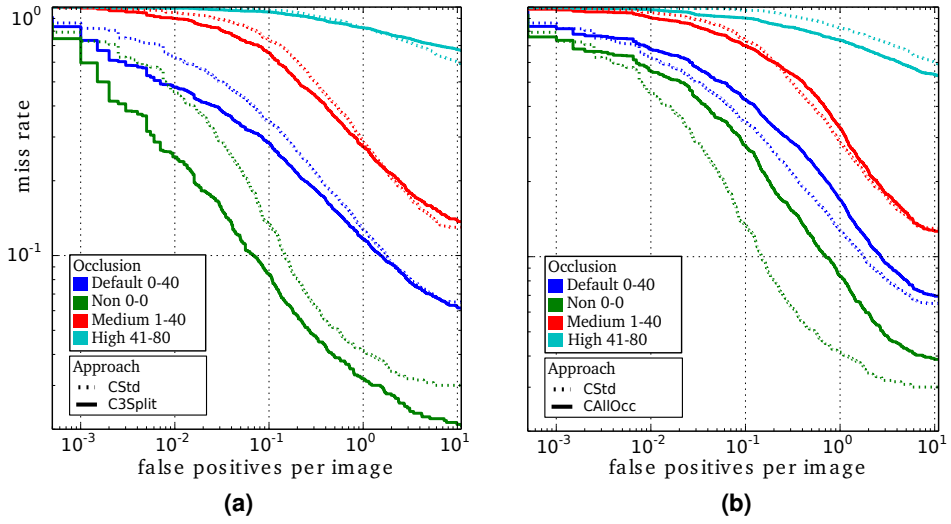


Figure 5.2: (a) Performance of C_{3Split}: C_{3Split} generally dominates C_{Std} at occlusion rates up to 40%, with an unexpected strong gain for un-occluded views. (b) Performance of C_{AllOcc}: For un-occluded views, the performance is worse than C_{Std}. However, C_{AllOcc} outperforms C_{Std} on strongly occluded car views.

Because of the improved detection performance of C_{3Split}, each vertical part is further divided into two horizontal regions. However, this C_{6Split} exhibits a much worse performance compared to C_{3Split} in a similar range as C_{Std}. So, the pure vertical split seems to more accurately reflect the occlusion constellations in the data.

Besides the adapted classifier architecture, C_{3Split} also differs from C_{Std} in training procedure by using additional occluded training examples in the second stage. In order to demonstrate that the improved performance of C_{3Split} is not simply caused by using different training data, a holistic detector that is similar to C_{Std}, but which uses the training data of the second stage of C_{3Split} with occlusion rates of 0–80%, is trained. This classifier is referred to as C_{AllOcc}. The worse performance of C_{AllOcc} (Fig. 5.2b) for un-occluded cars indicates that the additional variation in the occluded training examples confuses the representation of un-occluded views. However, for 41–80% occlusion, C_{AllOcc} outperforms C_{3Split} and C_{Std}, so the holistic approach can better employ the remaining information in case of strong occlusion, maybe by directly representing the effect of the occlusion edge.

5.3 Additional Use of Depth Information

C_{3Split} demonstrates an improvement in detection performance for cars with an occlusion rate of 0–40%. For cars with an occlusion rate of 41–80%, the performance is nearly the same as C_{Std}. The confidence values of the sub-segments seem to be insufficient information for the SLP in the second stage.

An analysis of the results reveals that very low confidence values in two of three sub-segments can result in a low confidence value at the top, regardless of the quality of the confidence value of the third sub-segment. There are two noticeable scenarios where this constellation occurs. First, some non-car objects accidentally generate a high confidence value in one sub-segment (clutter example in Fig. 5.3 top). This potentially happens often. Second, a strongly occluded car is only visible in one of three sub-segments (car example in Fig. 5.3). Unfortunately, it is not possible to distinguish both patterns by using only the confidence values. Therefore, additional context cues, e.g. stereo disparity, are needed to overcome this limitation. For example, the fact that the occluded part of a car is closer to the camera than the visible part could be exploited.

One idea is to calculate the median depth of each sub-segment and provide it as additional input to the SLP at the second stage. The depth values are divided by 100 in order to scale them to a similar range as the confidences. The resulting architecture is shown in Fig. 5.1c, and referred to as $C_{\mathrm{3SplitDepth}}$. The

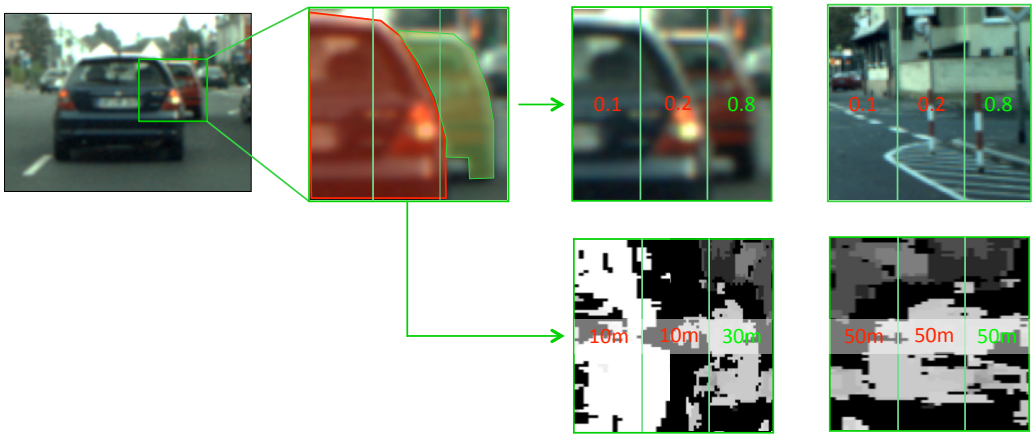


Figure 5.3: Strongly Occluded Car Views: (Top line) The green rectangles indicate the sub-segments of the divided car template. High confidence values are generated only at sub-segments that include visible car parts. So, only one sub-segment generates a high activation. The same effect is visible at non-car objects, as the clutter example demonstrates. Both cases generate the same output. (Bottom line) In addition to the confidence values for each sub-segment, the median depth information is shown. The classifier can use this to distinguish between both cases.

absolute depth values exhibit a wide range of variation, which might be too difficult to handle with an SLP. It is assumed that only the relative distance between the car and a possible occluding object contains the significant information. Therefore, the smallest depth value is simply subtracted from all other segments. The bottom line of Fig. 5.3 displays the two same constellations as the top line, but now the corresponding depth values of each sub-segment are included. Each sub-segment of the clutter example shows the same depth value, while the sub-segments of the car example show lower values if an occluding car is inside the sub-segment. Therefore, an edge in the depth information distinguishing between both cases can be seen.

To better comprehend the data in the resulting feature space, the difference between left and right depth over the difference between left and right confidence have been plotted. Fig. 5.5a demonstrates that there is no way to separate the car and clutter linearly. To solve this, the information was transformed as follows. Instead of the six features (three confidence and three relative depth values), the quadratic combinations of these six features as input-dimensionality were used. Use of all combinations generates 36 features. Some of these feature combinations occurred twice at a time, which is no benefit for the detection performance. Because of this, the double feature combinations were deleted. Finally, the SLP

on the second stage of $C_{\text{3SplitDepth}}$ was trained in the resulting 21-dimensional feature space.

The results in Fig. 5.4a reveal a significant gain compared to C_{3Split} for all occluded car views, so the detector can exploit the additional information. Only for non-occluded car views does $C_{\text{3SplitDepth}}$ evidence some loss in detection at high or low FP rates per image. However, it is at least still superior to C_{Std}. Fig. 5.4b illustrates that $C_{\text{3SplitDepth}}$ outperforms C_{Std} at all rates of occlusion.

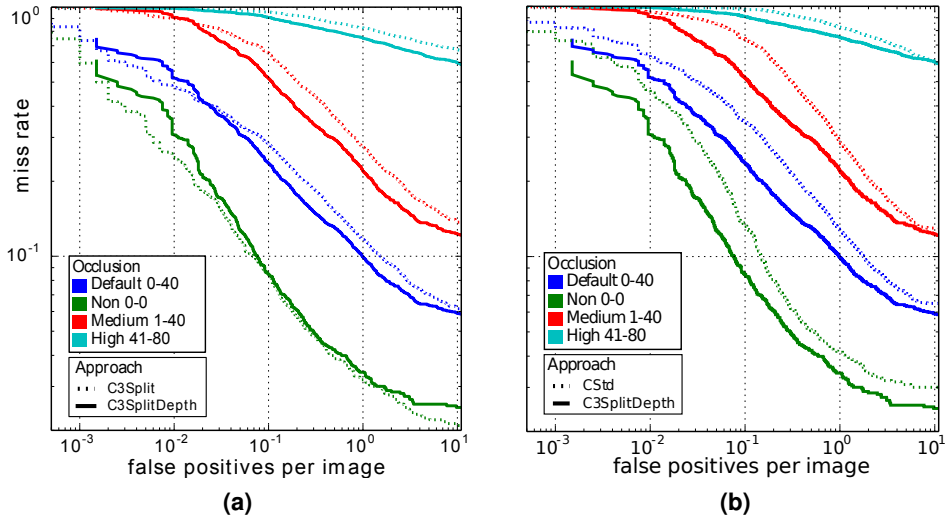
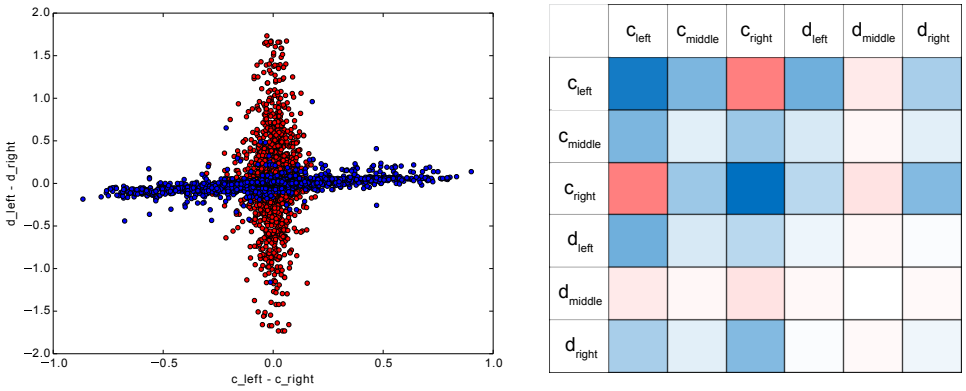


Figure 5.4: Performance of $C_{\text{3SplitDepth}}$: (a) $C_{\text{3SplitDepth}}$ shows a significant gain for occluded views compared to C_{3Split}. (b) $C_{\text{3SplitDepth}}$ generally dominates C_{Std} at all rates of occlusion.

Fig. 5.5b visualizes the learned weights in order to facilitate a better estimation of which feature combinations are exploited by the SLP. In general, the SLP seems to use the confidence values more than the depth values. The weights for c^2_{left} and c^2_{right} have the highest positive values, which suggests that a high activation in one of these sub-segments can generate a high final confidence. In contrast, the weight for the combination of $c_{left} * c_{right}$ indicates a high negative value, so the classifier tries to limit the confidence if c_{left} and c_{right} are activated strongly together.

5.4 Detection Examples

This section presents some detection examples of the classifiers. It uses a fixed FPPI of 10^{-1} for the evaluation. Fig. 5.6 and Fig. 5.7 show generated false negatives of C_{Std} (top line), C_{AllOcc} (middle line), and $C_{\text{3SplitDepth}}$(bottom line). False



(a) (b)

Figure 5.5: (a) Visualization of Confidence and Depth Values: Blue dots denote car segments, and red dots denote clutter. (b) Visualization of the SLP's weights: Blue boxes denote positive weights, while red boxes denote negative weights. The color's saturation indicates the absolute value of the weights.

negatives are marked with red rectangles for C_{Std} and with green rectangles for C_{AllOcc}. By using C_{Std}, numerous generated false negatives can be seen in both figures. The most false negatives were generated at occluded car views. Thereby, non-occluded and weakly occluded cars were also not detected. The second line in both figures represents the generated false negatives by using C_{AllOcc}. The classifier can deal with occlusion more effectively than C_{Std}, but also produces some new false negatives at non-occluded car views. In general, C_{AllOcc} displays a significant decrease of generated false negatives, but also more FPs (not shown). The bottom line in both figures indicates the results of $C_{\mathrm{3SplitDepth}}$. In both test images, no false negatives are generated. In general, $C_{\mathrm{3SplitDepth}}$ presents significantly fewer FPs compared to C_{Std} and C_{AllOcc}.

5.5 Conclusion

This chapter has presented a two-stage architecture to improve the detection of occluded objects. It has demonstrated a vertical three-part split of the holistic car template in the first stage to deal with vertical occlusion. Thereby, each part was trained on un-occluded car views. In the second stage, the typical combinations of these part responses were learned on occluded examples. This contrasts with a MLP, where each stage is trained with the same data and this split helps the architecture deal with occlusion on a more symbolic level. The prototype outperformed the reference system for different levels of occlusion. However, there

Figure 5.6: Detection Examples I: The figure depicts a detection example of C_{CStd} (top), C_{AllOcc} (middle), and $C_{\mathrm{3SplitDepth}}$(bottom). The false negatives are marked in red at the top and in green in the middle. $C_{\mathrm{3SplitDepth}}$ produces no false negatives.

was no gain for strongly occluded cars. It was argued that, based on a high confidence in only one sub-segment, the classifier cannot distinguish between car and clutter. Therefore, depth information was integrated into the vertical occlusion prototype to give the classifier an independent cue to reason about typical occlusion patterns. It was demonstrated that the additional information improves the detection performance for occluded car views. Fig. 5.4b indicates that $C_{\mathrm{3SplitDepth}}$ outperforms C_{Std} at all rates of occlusion.

In general, the model can also transfer to other application areas that include structured occlusion that the sub-division of a holistic template can model. Despite improvements, there is a persisting limitation of a holistic discriminative approach that the detection performance decreases if untrained occlusion constellations occur. The next chapters establish a motivation to switch to a parts-based classifier and discuss some occlusion-handling strategies that are espe-

Figure 5.7: Detection Examples II: The figure shows a detection example of C_{CStd} (top), C_{AllOcc} (middle), and $C_{\mathrm{3SplitDepth}}$(bottom). The false negatives are marked in red at the top and in green in the middle. $C_{\mathrm{3SplitDepth}}$ produces no false negatives.

cially helpful for strongly occluded car views.

6 Rendered Benchmark Dataset

Chapter overview *A fundamental problem for the development and analysis of occlusion-handling strategies is that occlusion information cannot be labeled accurately enough in real-world video streams. Often, these video streams provide only rectangles for the object instances in the scene, regardless of whether hand-annotation or sensor-based annotation is used. This chapter presents a rendered car detection benchmark with pixel-level information of the objects and with controlled levels of occlusion.*

Parts of this chapter are based on:

[C15a] M. Struwe, S. Hasler, and U. Bauer-Wersing. Rendered Benchmark Data Set for Evaluation of Occlusion-handling Strategies of a Parts-based Car Detector. *PSIVT*, pages 99–110, 2015.

6.1 Introduction

The previous chapters have engaged with various occlusion-handling strategies for a holistic discriminative detection framework. The first framework presented the use of car-car constellations and demonstrated that the system clearly outperforms the reference system. There was also the illustration of an extended framework that takes into consideration more general occlusion patterns during training and makes additional use of depth information as a plausibility check. The system also clearly outperforms the reference system. For each classifier, the detection performance was evaluated by taking different rates of occlusion into account. This is important in order to see the limitations of the classifiers for occluded object views. The previous experiments used hand-annotated video streams, which contain 2D bounding boxes as labels for the objects and additional information, such as orientation and rate of occlusion. However, there are more public available datasets like [Everingham et al., 2010] and [Dollár et al., 2009] that are often used for benchmarking but do not provide detailed occlusion information of the object instances. Some benchmark datasets like KITTI ([Geiger et al., 2012]) use 3D bounding boxes for objects, but do not provide pixel-level information on the constellation of occlusions. Implementing a render framework to generate a new dataset especially tailored for the evaluation of occlusion-handling strategies can facilitate more accurate labeling and a more effectively control of the scene conditions. Hence, pixel-level mask information about the object instances can be generated. Also, the rate of occlusion can be defined more precisely if hand-annotated data is used. The strategy of using synthetic or artificial data for the training is also followed by other publications like [Rozantsev et al., 2015], [Peng et al., 2014], [Jaderberg et al., 2014], [Zhang et al., 2015], and [Yu et al., 2010].

6.2 Rendered Benchmark Dataset

With a rendering framework, it is possible to define the position of a car, the position of the light source, the intensity of the light, and the angle of rotation of the car. Moreover, it is possible to estimate which pixels of the rendered image belong to which object instance, and this information can be stored in a mask that is more precise than a usual bounding box. For the rendering, a framework called Blender[1] was used. Blender supports the programming language Python for scripting, which offers the possibility to generate complex scenes automatically. With such a script at hand, many scene parameters, e.g. rate of the object's rotation, can be set randomly. The scene is built in the so-called "object-view." In addition to the object-view, the "node-editor" is used, which allows for defining which information has to be generated from the scene. Since the mask information is important for the occlusion-handling, the pixel-level information for each object instance is generated with the alpha channel (Fig. 6.1). The result is the mask information of the object instances. Also, occluding objects can be set as invisible to obtain the full mask information of the occluded object. Overall, it is easy to generate each possible mask information of an occlusion constellation for later evaluation. The pixel-level information can help determine the exact rate of occlusion of an object constellation, which is an immense advantage in comparison to hand-annotated video streams.

For the dataset, the minimum and maximum border of a parameter were defined, then randomly generated inside these ranges. For example, the position and illumination intensity of the light source were generated randomly at a transparent dome around the centered car. In total, 88 car models were used. Fig. 6.2 offers an overview of the various car models with the changing scene conditions.

The size of each rendered segment was set to 175×70 pixels, which corresponds to the template size of the car detector plus a border. The size of each car model was normalized so that the side view covered a given width of the segment. In general, a parts-based model is trained to a limited rotation change. Multiple detectors have to be trained for full rotation. Since this thesis concentrates on the evaluation of occlusion-handling strategies, it used only side views of cars to omit full rotation-handling. Therefore, it used only small rotation changes. The views were randomly rotated in a range of 30° in both directions. Since most recognition models use a regular grid for feature extraction at the detection step, the car template does not always fit to the car at the test image. To cope with this, the center position of the car was randomly shifted in a range of five pixels to the top, the bottom, the left, and the right. After rendering, the car models

[1] Blender is an open-source rendering framework. https://www.blender.org

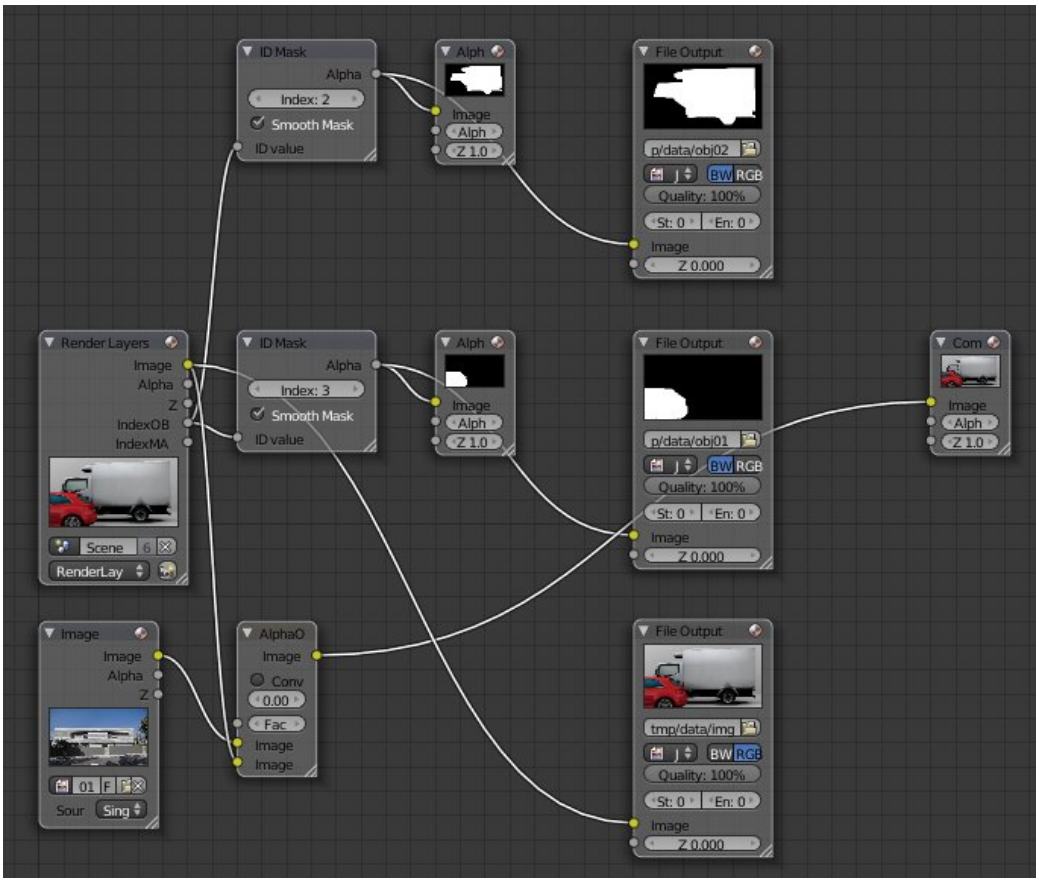


Figure 6.1: Node-Editor for Alpha Channel Extraction: The node-editor is used to filter out the alpha channels of each object instance in the scene. The result is a separate mask for each object instance.

Figure 6.2: Dataset with Different Scene Conditions: A dataset using all available car models with randomly chosen conditions for light position, light intensity, and angle of rotation is generated.

Figure 6.3: Example of Training Segments: The upper line depicts some car segments with a cluttered background, while the bottom line presents the corresponding masks.

were pasted in segments of car-free street scenes in order to incorporate realistic clutter in the background. The top row of Fig. 6.3 depicts some segments with cluttered background for the training. The bottom row presents the corresponding masks.

A total of 400 views for each car model were used for the dataset. Segments of 44 car models were used as the training set, while the other 44 car models served as the test set. The training set included non-occluded car views with a segment size of 175×70 pixels.

Different test sets with changing rates of occlusion were generated. The set with 0% occlusion portrayed a car object in the center of a car-free street scene. These images were then combined with an occluding object to generate test sets with 20, 40, 60, and 80% of occlusion. A car-like shape was used to avoid feature activation at the occluding object, which is the case when just using car objects as occluding objects. An ellipse-shaped patch was used to achieve a car-like shape for the occluding object. This enabled a stronger analysis of how the occlusion strategies affect the activation of the features at the occluded object. Instead of a simply black-colored ellipse, occluding object patches of car-free street scenes were cropped out. A black-colored shape generated artificial edges at the image, which led to feature activation at these areas and influenced the evaluation results. The black-colored area itself also produced no feature activation, which might be unfair because no FPs can be generated at this area. Fig. 6.4 offers an example for a generated test image and the corresponding mask. Occluding constellations without any cars were also generated for a consistence evaluation of the investigated occlusion-handling approach. So, the car-like shape occluded the background image. A total of 17,600 segments of un-occluded car views were generated for the training set, and 105,600 images with different rates of occlusion were generated for the test set.

The following chapter employs this dataset for an evaluation of occlusion-handling strategies for a parts-based car detector. Thereby, it uses the generated mask information of the occluding object to predict the occlusion rate of some occluded objects. This occlusion rate is used to re-weight the score of the visible object parts.

Figure 6.4: Example of the Test Set: At the left, a test image with an occluding ellipse is shown. The right image displays the mask of the occluding object.

7 Occlusion-handling Strategies of a Parts-based Car Detector

Chapter overview *Chapter 4 and 5 have discussed several strategies to improve the detection performance for holistic discriminative approaches. Even with occlusion-handling strategies, the detection performances of the holistic discriminative approaches decrease with the appearance of an occlusion constellation that is not learned during training, or if the objects are strongly occluded. In contrast, parts-based detection approaches can deal implicitly with occluded or untrained object views. This chapter presents a parts-based car detector and evaluates several occlusion-handling strategies on the aforementioned rendered dataset.*

Parts of this chapter are based on:

[C15a**]** M. Struwe, S. Hasler, and U. Bauer-Wersing. Rendered Benchmark Data Set for Evaluation of Occlusion-handling Strategies of a Parts-based Car Detector. *PSIVT*, pages 99–110, 2015.

7.1 Introduction

Chapter 2 has presented several detection approaches, focusing on how they address occlusion. These approaches reveal insufficient detection results for moderately to strongly occluded object views. Chapter 4 and 5 have discussed some strategies to improve the detection performance of these holistic discriminative approaches. The results showed that these approaches clearly outperform the reference system. However, if an occlusion constellation was not learned during training or the objects are strongly occluded, the detection performance decreases, regardless of which occlusion-handling strategy was used. Motivated by research findings that suggest that parts-based detection approaches are more flexible in handling occlusion strategies, this chapter presents a novel parts-based detection framework.

As mentioned, there are some approaches that make use of mask information to deal with occlusion. Various methods can generate this mask information, e.g. using depth information, as in [Makris et al., 2013], or object segmentation, as in [Fidler et al., 2013], [Borenstein et al., 2004], [Serre et al., 2007] or [Torrent et al., 2011]. Serre et al. [2007] differentiate between shape-based and texture-based objects (Fig. 7.1). Texture-based objects are used for the segmentation, and shape-based objects are found through a windowing process. At the end, the results of both, the segmentation and the windowing process, are used to generate a fully segmented scene output.

The approach by Torrent et al. [2011] is based on a boosting procedure that automatically decides - according to the object properties - whether it is better to give more weight to the detection or segmentation process to improve the result. One idea is to use the segmentation mask by Serre et al. [2007] and Torrent et al. [2011] to find typical areas for occluded objects.

In our framework, the ground truth data of the rendered data set is utilized to demonstrate the best use of mask information to improve the detection performance during occlusion. The ground truth data of the rendered dataset provides optimal mask information, which seems to be unrealistic to obtain if, for example, stereo images are used to generate the mask. Because the focus is on using mask information, the following discussion neglects its extraction. At a later stage, mask information can be generated from depth information, generated by a stereo vision system. Of course, this information will be less optimal and more noisy than the rendered ground truth data.

The parts-based detection approach by Makris et al. [2013] presented earlier uses depth information to determine the visibility of a car hypothesis. This information functions as a kind of mask for re-weighting the score of an object candidate. This chapter evaluates some occlusion-handling strategies and illustrates ways to improve the detection performance, especially for strongly occluded object views.

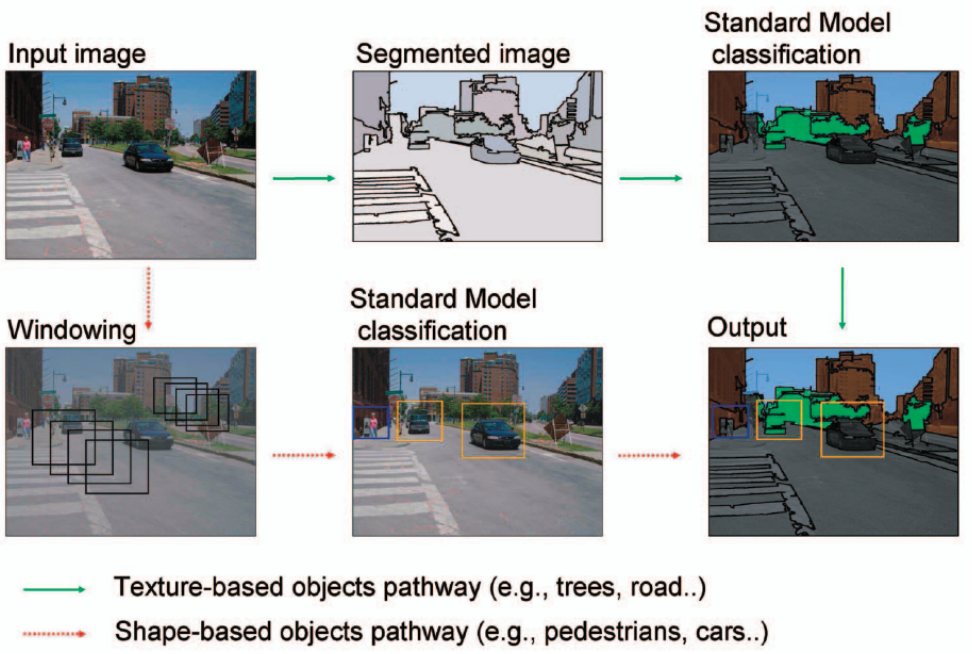


Figure 7.1: Shape-based and Texture-based Segmentation by Serre et al. [2007]: Shape-based and texture-based channels are processed in separate ways. The green arrows mark the segmentation based on the texture information of the object. The lower red arrows mark the windowing process for the classification of the shape-based objects.

7.2 Parts-based Car Detector

A parts-based detection framework contains several steps at training and detection: extraction of texture descriptors, learning the parts-based object representation, feature matching, and accumulation of single-feature activations. This chapter explains all steps in detail and shows how they are integrated in the car detector.

7.2.1 Extraction of Texture Descriptors

Texture descriptors are used at a defined position in an image to transfer the pixel information to another representation. Thereby, in most cases, gray-value pixel information at a defined patch is transformed to a gradient-based vector representation. Popular texture descriptors are SIFT [Lowe, 2004] and speeded up robust features (SURF, [Bay et al., 2006]). Chapter 8 presents a comparison and evaluation of both descriptors and highlights that SIFT descriptors are more robust against changing scene conditions and more suitable for parts-based detection methods. Therefore, SIFT descriptors are used in this chapter.

Voting methods such as [Calonder et al., 2010, Higa et al., 2013, Leibe and Schiele, 2006] use a key-point detector to define positions for the extraction of texture descriptors in the input image. These positions are often called interest points. Please note that these points are mainly generated in highly textured areas, which in turn leads to a sparse distribution of the interest points over the object's surface. Fig. 7.2 illustrates some detected interest points at the background in addition to the sparse distribution. Instead of a key-point detector, a regular grid is used to create more interest points. Therefore, a dense grid over the input image is used to define possible interest point positions. A closely meshed grid delivers a high number of interest points, and consequently a higher chance of finding similar features after position shift. However, increasing the number of interest points also increases the computing time at all further steps of the framework. A gap of 5 pixels is used both horizontally and vertically for the framework, which is the best compromise. To limit the number of interest points, the extraction of descriptors from low-textured areas is avoided because these areas result in non-discriminative code-book entries. Therefore, an edge detection with field same the size as the patches is performed. A patch is selected if the edge detector indicates a minimum response above a predefined threshold. This filtering step yields a threefold reduction in the number of features. The combination of the grid and the edge detector generates more interest points than the default key-point detector of SIFT, but also takes the computing time into account. Fig. 7.3 displays all described steps for the interest point selection. Motivated by

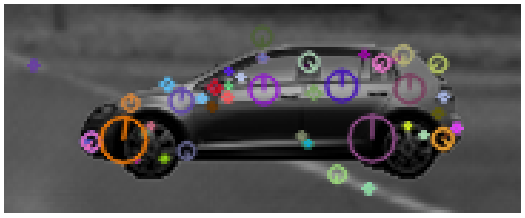


Figure 7.2: SIFT Key-Point Detector: The detector delivers a low number of interest points sparsely distributed on the object's surface. Also, some interest points in the background are generated.

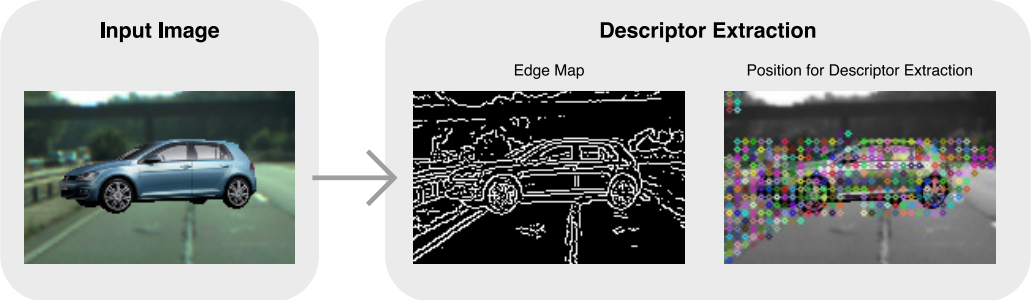


Figure 7.3: Extraction of Texture Descriptors: A regular grid is used to define possible positions for texture extraction. The edge map of the descriptor extraction layer is used to reject points that do not provide enough pixel information. The resulting positions for the texture extraction can be seen in the right-most panel.

the feature descriptor analysis in Chapter 8, SIFT descriptors are used at the selected interest points. For the feature extraction, the rendered training dataset with segments that was already presented in Chapter 6 was used. The segments in the rendered training dataset have a resolution of 175×70 pixels.

7.2.2 Learning of Parts-based Object Representation

The object representations of a parts-based detector are stored in a so-called "visual code-book." The code-book includes the features and their relative position to the object's center. One method to build the code-book is to use a clustering method. The resulting clusters are the features or the entries of the code-book. Annotated video streams usually provide bounding box information for object instances. During training, these bounding boxes are used to extract patches. The resulting patches have only partial car information, but also include information from the background. The background information interferes with the object that has to be learned. These accidental non-car features must be filtered out afterwards, which is a challenging task by itself. By using the rendered data, the mask

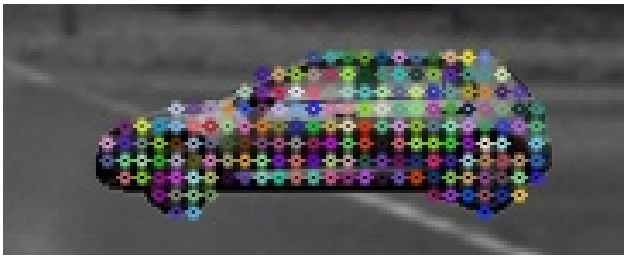


Figure 7.4: Use of Mask Information for the Extraction of Texture Descriptors: During training the mask information is used to avoid the extraction of features at the background.

information of the object is utilized to limit the extraction of features to the object to be learned. Interest points are selected only if the center of the patch is located at the object, as can be seen in Fig. 7.4.

For training, segments without occlusion are used. Parts-based detection approaches like [Makris et al., 2013] use a code-book to store the generalized model of the object that has to be learned. Here, a MiniBatchKMeans[1] clustering is performed to build such a code-book. MiniBatchKMeans splits the data into chunks, making the clustering much faster and more memory efficient. Unlike the original Kmeans, this ensures that there is no limitation in the amount of data used during training. Fig. 7.5 illustrates two exemplary features from the learned code-book. The left side shows the "occurrence maps," which describe the distribution of occurrence for each stimulus of the code-book cluster. The activations in the occurrence maps indicate the relative position of the stimuli to the center position of the car. The use of a regular grid at the feature extraction step, as described in Sec. 7.2.1, results in a map size of 36×7 pixels by using a gap width of 5 pixels and the initial size of the training segment. To obtain a probability distribution, the sum of each occurrence map is normalized to one. The right side shows corresponding stimuli of the descriptor clusters.

The next section describes how the code-book is used for detection.

7.2.3 Parts-based Detection Framework

This section introduces the parts-based framework for car detection. The first step was the extraction of descriptors on the full test image dataset described in Sec. 7.2.1. This performed a fivefold reduction in both directions in the resolution of the input image. For each descriptor, the best-matching feature is determined

[1] MiniBatchKMeans is an extended version of the KMeans clustering. In contrast to KMeans, the method splits the data into chunks to save memory and computing time. More information can be found at http://scikit-learn.org/stable/modules/clustering.html

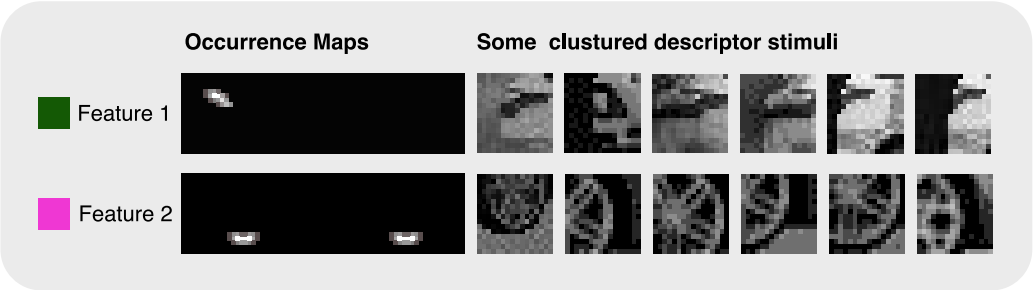


Figure 7.5: Examples of Code-Book Features: The left images show the occurrence maps, which describe the distribution of the occurrence of each stimulus of the code-book cluster. The right images show some corresponding stimuli of the descriptor cluster.

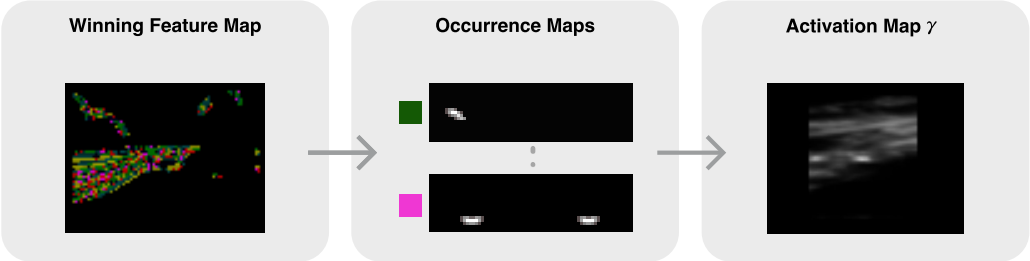


Figure 7.6: Detection Framework: The winning feature map shows the best-matching feature for each pixel. Each feature votes with its occurrence map for the object's center. An accumulation of all features is used to build the activation map.

by using the response of an activation function. To get a confidence value between zero and one, the following activation function is used:

$$act(f_i|I) = exp\{-\frac{||f_i - f_{ci}||^2}{o}\} \tag{7.1}$$

Here, I is the input image. The Euclidean distance of the current feature f_i to each feature of the code-book f_{ci} is iteratively calculated. The denominator o controls the intensity of the activation.

The winning feature votes with its occurrence map for an object's center. An accumulation of the votes of all features on the test image supportes the construction of an activation map (Fig. 7.6). The accumulated score map is referred to as γ.

During detection, simple thresholding was used to define the FPs per image. Fig. 7.7 reports the detection performance of this parts-based car detector for 0, 20, 40, 60, and 80% occlusion. Car center hypotheses that indicated a minimum

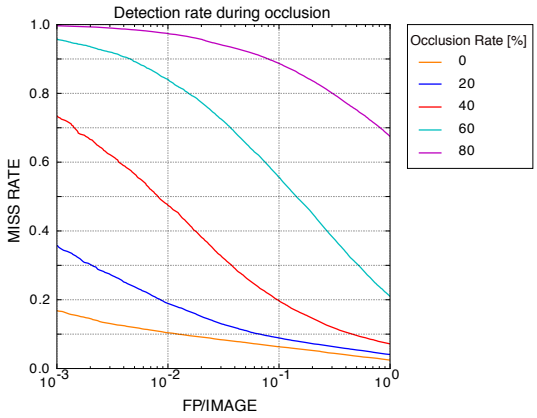


Figure 7.7: Detection Performance: The ROC plot shows the detection result of the visibility-based parts-based car detector for five rates of occlusion. For the evaluation, 17,600 test images are used for each occlusion rate, separately. Additionally, 17,600 test images with occluding ellipses but without any cars are utilized. So the occluding objects occlude the background. In total, 105,600 test images are used.

overlap of 80% in height and 60% in width with the ground truth were counted as detected cars. A loss in detection performance corresponds with increasing occlusion.

7.2.4 Analysis of the Detection Performance and Computation Time of the Code-book

For MiniBatchKMeans, the number of used clusters has to be initialized and is decisive for the detection performance at test time. An insufficient amount of clusters generates widely dispersed occurrence maps, which in turn results in an inaccurate hypothesis for the position of the car's center. An excessive number of clusters increases computing time significantly and generates a poorly generalizing model of the trained car model. Spatially-specific occurrence maps are not widely spread and vote for a defined object center, which yields an improved detection performance.

To find the optimal amount of clusters, the computing time is checked and a quality value for the generated clusters is calculated with a self-defined function. This quality is calculated using a pixel-wise squaring, as shown in the following.

$$quality(All) = \frac{\sum_{k=0}^{k \leq K} \left(\frac{O_k}{\sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} O_k(i,j)} \right)^2}{K} \tag{7.2}$$

Table 7.1: Quality of the Occurrence Maps: This table presents the calculated quality value, which is the result of a quality function. The quality value increases as the number of clusters increases. High values denote a dense spatial distribution of stimuli in the occurrence maps, which results in a more accurate hypothesis for the position of the object's center.

# clusters	50	100	150	200	250
Quality	0.012	0.014	0.015	0.016	0.016
# clusters	300	350	400	450	500
Quality	0.017	0.017	0.018	0.018	0.018

Table 7.2: Computation Time for Changing Number of Clusters: The table specifies the computing time during detection by changing the number of used clusters.

# clusters	50	100	150	200	250
Time in s	1.291	1.436	1.893	2.277	2.464
# clusters	300	350	400	450	500
Time in s	2.801	3.279	3.751	4.491	5.414

Here, K is the total number of clusters or code-book entries, and O_k is the occurrence map of the corresponding code-book entry k. High values denote a dense spatial distribution of stimuli in the occurrence maps, which generates a more accurate hypothesis for the position of the object's center. Low values correspond with unreliable clustering results. For evaluation, the quality was calculated for 50–500 clusters by iteratively increasing the number of clusters by 50. The results in Tab. 7.1 indicate the calculated quality value, which is the result of the quality function. High values represent a more balanced distribution of the used features. The quality value increases as the number of clusters increases. However, at 200–250 clusters, the increase saturates.

For evaluation, the computing time of 50 test images was calculated in addition to the quality value. The test was performed on a single-core CPU and implemented in Python. Tab 7.2 demonstrates a significant increase in computation time as the number of clusters increases, as expected. With respect to the computation time and the results in Tab. 7.1, 220 clusters were used, which is a reasonable compromise between both criteria.

It was investigated how the number of clusters influences the computing time and the quality of the clustering result. In order to additionally visualize how the

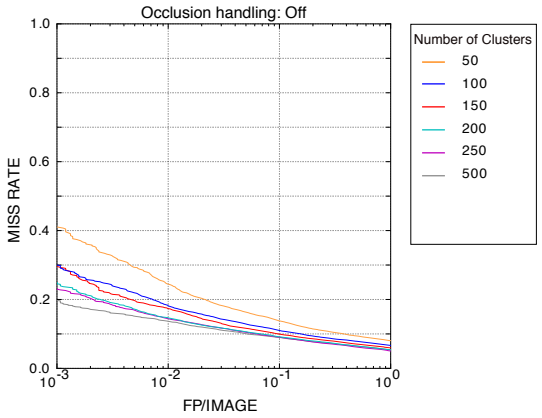


Figure 7.8: Detection Performance at Changing Numbers of Clusters: A total of 17,600 non-occluded car views and 17,600 test images with occluding constellations without any car are utilized for each cluster configuration. In total, 35,200 test images are used per curve.

number of clusters influences the detection performance of the parts-based car detector, a code-book in a range of 50–500 clusters was built by iteratively increasing the number of clusters by 50. The resulting detection performance is plotted in Fig. 7.8. At an FPPI rate of 10^{-2}, increasing the number of clusters to 200 reveals a significantly improved detection performance. However, for a higher number of clusters, only a slight gain is evident. For example, at an FPPI of 10^{-2}, the detection performance for 200 and 500 clusters is nearly the same. Please note that the curve for 0% occlusion and 250 clusters shown in Fig. 7.7 reflects a superior detection performance compared to the curve in Fig. 7.8. The reason for this is the arrangement of the dataset for testing. In both cases, the same amount of negative examples was used, while the number of positive examples was changed. The evaluation in Fig. 7.7 utilized 17,600 test images for each single occlusion rate. Additionally, it applied 17,600 test images with occluding constellations without any cars. This resulted in 105,600 test images. The evaluation in Fig. 7.8 utilized 17,600 test images of car views with 0% occlusion, as well as 17,600 test images with occluding constellations without any cars. This resulted in 35,200 test images. The total number of test images directly influences the FPPI rate in the ROC plot.

7.3 Occlusion-handling of the Parts-based Car Detector

Motivated by the approach of Makris et al. [2013], a visibility-based occlusion-handling strategy that predicts the occlusion of an object by using the mask of

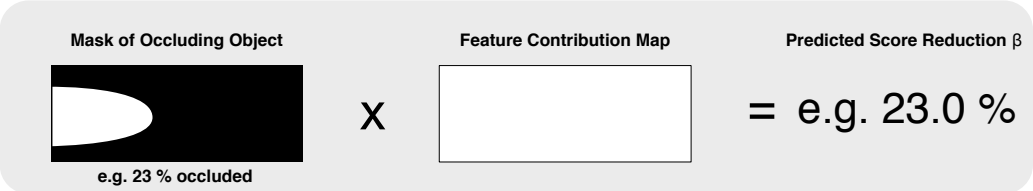


Figure 7.9: Uniform Contribution of β Calculation: On the left, the mask of the occluding object covers 23% of a support window. The feature contribution map indicates a uniform contribution that results in a predicted score reduction of 23%.

an occluder is built. The mask information is used in two ways: First, the accidental features of the occluding object are deleted at the accumulation step. This ensures that no features of the occluding object would influence the score of the object hypothesis. Second, it is used for the re-weighting of the score of the possible car hypothesis. The re-weighting of the score γ is done by taking the predicted score reduction β into account, which is determined by the predicted occlusion. A so-called "support window" is used to calculate the percentage of the occlusion. The support window has the same size as the occurrence maps and covers the area of features that potentially contribute to the car hypothesis at the center of this window. During detection, the support window is shifted over the input image. In order to calculate β, a uniform distribution of the features over all pixels of the support window is assumed (Fig. 7.9). Consequently β is defined here as the ratio of the area covered by the mask divided by the overall feature contribution map area.

As previously mentioned, the predicted occlusion of the support window can be used to reject detected features inside the occluding area before accumulating the score. This score excludes occlusion-handling, but includes rejected features, and is referred to as γ'. The final score γ'' is then calculated by using the predicted score reduction β of the supporting window (Fig. 7.10 with [A] for the re-weighting),

$$\gamma'' = \gamma'/(1-\beta) \tag{7.3}$$

If β is zero, re-weighting has no effect. By increasing β, re-weighting increases the score while the influence of γ' diminishes. This can generate FPs at high values of β while fewer features are detected. The goal is to avoid this case with a β_{max}. For this, it is necessary to find the maximum value for β that improves detection performance, which requires a limitation of β up to a defined maximum score reduction. So, β is defined as follows:

$$\beta := min(\beta, \beta_{max}) \tag{7.4}$$

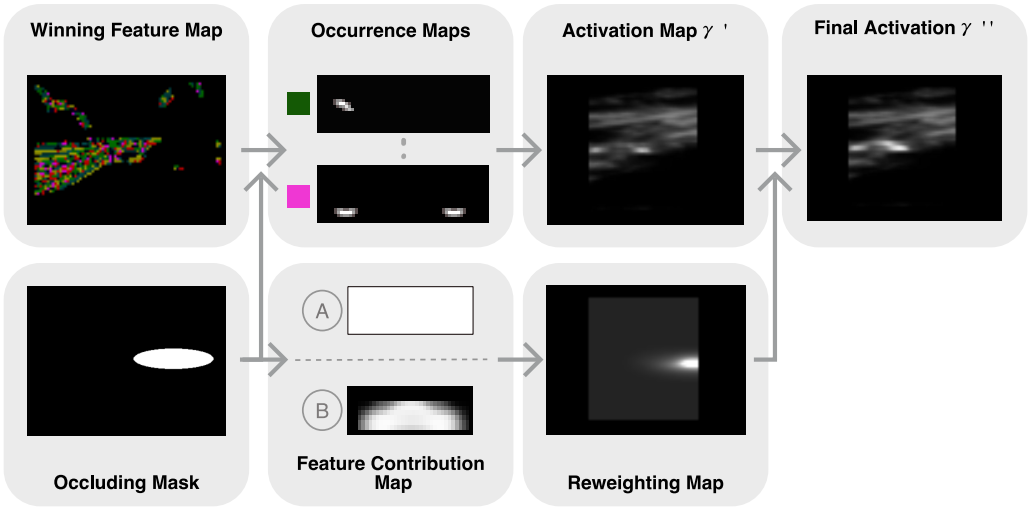


Figure 7.10: Detection Framework with Occlusion-handling Strategy: The mask of the occluding object is used to calculate the activation score by deselecting the winning features inside the occluding area. For uniform occlusion-handling, the re-weighting [A] is used to generate the final score. For the contribution-aware occlusion-handling, the re-weighting [B] is used, which is explained in Sec. 7.4.

The optimal value β_{max} is found by evaluating the detection performance of the car detector with values from 0.1 to 0.9, in increments of 0.1. Occlusion rates of 20, 40, 60, and 80% were utilized in order to study the effect of β_{max} at different rates of occlusion. Fig. 7.11 presents the detection results.

In a driver-assistance system, a high number of FPs would generate many warnings, which can disturb the driver. Therefore, a relative low FPPI of 10^{-2} was chosen as a reference point. Given this FPPI, a β_{max} of 0.2 shows the best detection performance for 20% occluded car views, while for 80% occlusion, a β_{max} of 0.5 yields the best results.

However, the results indicate that a single best value for β_{max} that works equally well for all occlusion rates cannot be found. Because the presented data base provides only mask information of the occluding object, a fixed β_{max} has to be used, which is discussed in detail in Sec. 7.4. To find a β_{max} value that optimizes overall performance, result for the full data set, including all occlusion rates for each β_{max}, was plotted. Fig. 7.12(Left) displays the most significant improvements at the reference point (FPPI of 10^{-2}) by using a β_{max} of 0.2. Fig. 7.12(Right) demonstrates an improved detection performance at all occlusion rates by using the determined optimal β_{max}. The dotted lines signify the detection performance of the car detector without occlusion-handling, while the solid lines indicate the detection performance of the detector with occlusion-handling.

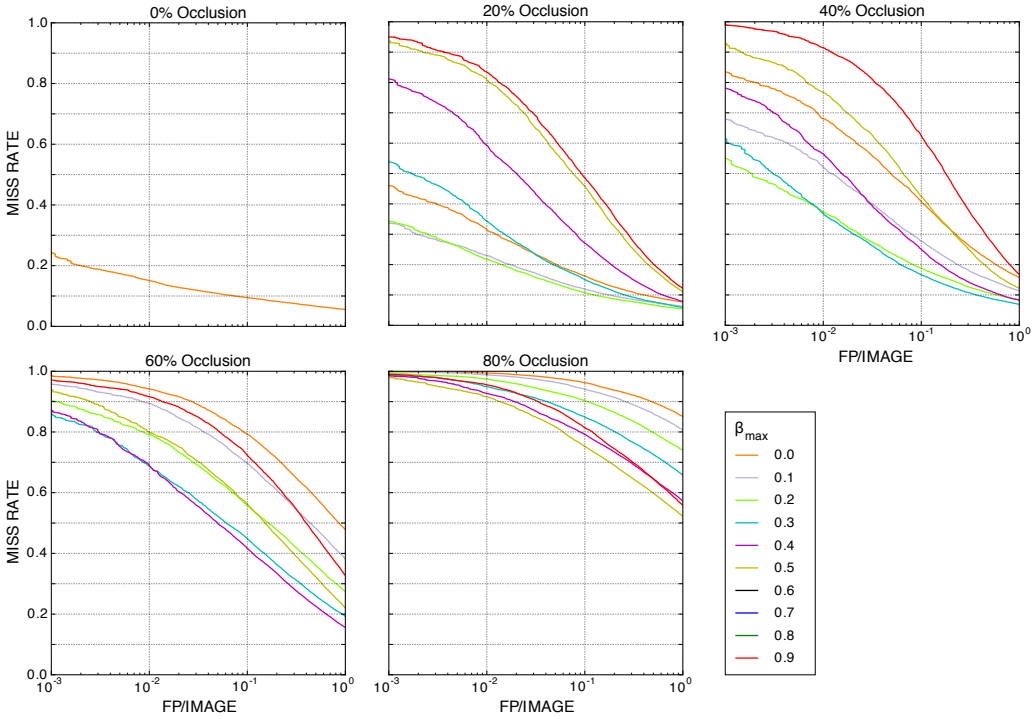


Figure 7.11: Evaluation of the Detection Performance: The plots reflect the detection result for 0, 20, 40, 60, and 80% occlusion and different values of β_{max}. Each curve shows results for 17,600 test images of the corresponding occlusion rate and 17,600 test images with occluding constellations without any cars. In total, each curve use 35,200 test images.

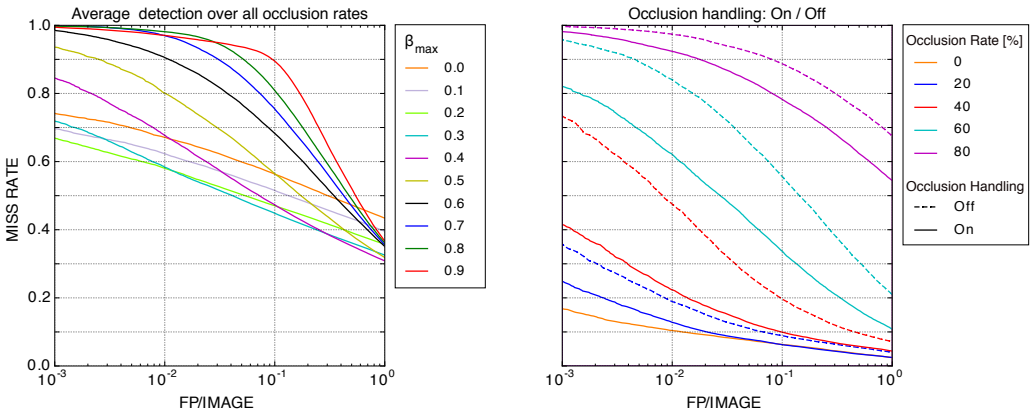


Figure 7.12: Optimal Parameter Setting for Occlusion-handling: (Left) Detection results for the full dataset, including all occlusion rates for each β_{max}. Each curve used 17,600 test images of the corresponding occlusion rate and 17,600 test images with occluding constellations without any cars. In total, each curve used 35,200 test images. (Right) Detection performance for different occlusion rates using the optimal value of 0.2 for β_{max}. The evaluation used 17,600 test images for each occlusion rate separately. Additionally, it used 17,600 test images with occluding constellations without any cars. So, the occluding objects occlude the background. In total, 105,600 test images were used. The dotted lines show results for the parts-based car detector without occlusion-handling, while the solid lines indicate the results of the detector with occlusion-handling.

For the reference point, the most gain is apparent at 40 and 60% occlusion, while there is a weaker gain for 20 and 80% occlusion. The weaker gain at 20% occlusion can be explained with the intrinsic ability of parts-based detection approaches to deal with occlusion. Because β_{max} is chosen the same for all rates of occlusion, the occlusion-handling improves the detection performance at 80% occlusion less significantly than for the other rates. The next section illustrates an occlusion-handling strategy that makes use of the occurrence maps of the code-book in order to predict the percentage of missing features for a possible car hypothesis.

7.4 Contribution-aware Strategy for Occlusion-handling of a Parts-based Car Detector

The previous section has demonstrated an improved detection by predicting the score reduction of a car hypothesis and re-weight the score of activation. However, this concept is based on the assumption that the learned features are uniformly distributed inside the support window. In order to get a more correct esti-

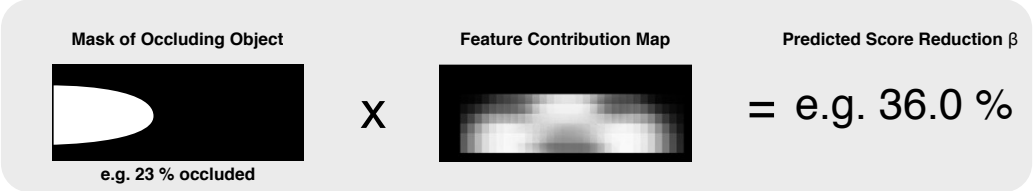


Figure 7.13: Contribution-aware Calculation of β: On the left, the mask of the occluding object for a support window can be seen. The feature contribution map shows a realistic contribution of the learned features. This results in a predicted occlusion rate of 36% if 23% of the supporting window is covered at the shown area.

mation of the missing amount of features, a so-called "feature contribution map" is used. This map is a representation of the contribution of all features at the car template that are stored in the code-book. For this, the occurrence maps of the various features in the code-book are summed, and a pixel-wise division by the total number of clusters is performed. The resulting map is stored in the feature contribution map. The contribution map is used for calculating γ' and β instead of assuming a uniform distribution of features. Now, γ'' takes into account the amount of missing car features (Fig. 7.10 with [B] for the re-weighting), and the parameter β reveals the percentage of missing features of the car hypothesis (Fig. 7.13).

As in Sec. 7.3, the detection results of all occlusion rates for varying β_{max} are plotted. In Fig. 7.14(Left), the use of 0.4 for β_{max} produces the best result by again using an FPPI of 10^{-2} as a reference point. This β_{max} is utilized for the car detector with the contribution-aware occlusion-handling. The detection performance is presented in Fig. 7.14(Right), whereas the new occlusion-handling is referred with the name extended. The plot indicates an improved detection performance at the reference point at occlusion rates of 40, 60, and 80%, while a slight loss is evident at 20% occlusion. The reason for this is the aforementioned overlap criteria. The contribution aware occlusion-handling strategy amplifies the score of car hypotheses that are located in the surroundings of possible car hypotheses at a low level of occlusion. The overlap criteria count these shifted car hypotheses as false FPs, which results in a weak loss in detection performance at 20% occlusion. For example object instance information of the scene can be used to overcome this drawback.

Using depth from stereo or other sensors, such as radar or laser, can generate the mask information of the occluding object, thus also provide instance information about the occluded object. This information can facilitate a plausibility check to reject the FP detection at 20% occlusion. Additionally, this information is help-

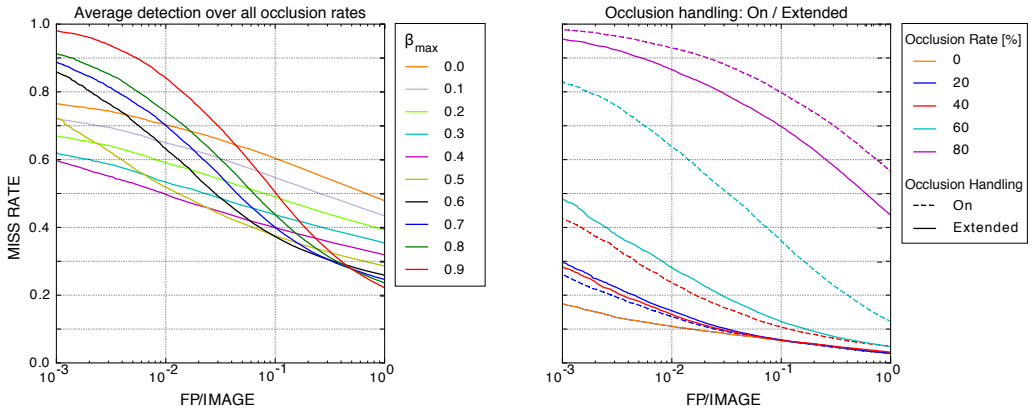


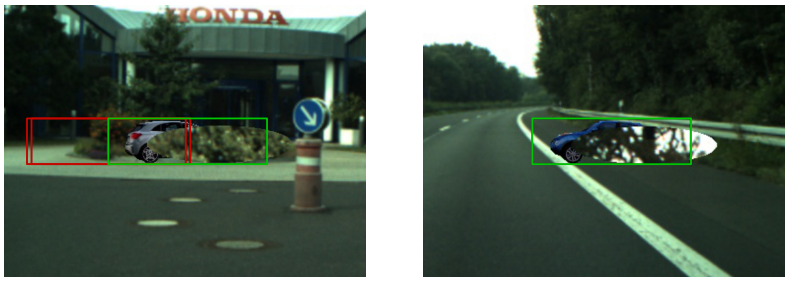
Figure 7.14: Optimal Parameter Setting for Contribution-aware Occlusion-handling: (Left) Detection results for the full dataset, including all occlusion rates for each β_{max}. Each curve used 17,600 test images of the corresponding occlusion rate and 17,600 test images with occluding constellations without any cars. In total, each curve used 35,200 test images. (Right) illustrates the detection results using the determined optimal value of 0.4 for β_{max}. The evaluation used 17,600 test images for each occlusion rate separately. Additionally, it used 17,600 test images with occluding constellations without any cars. So, the occluding objects occlude the background. In total, it used 105,600 test images. The dotted lines signify the detection performance by using the uniform occlusion-handling, while the solid lines indicate results of the contribution-aware occlusion-handling.

ful for defining a variable β_{max} for each object-object constellation in a scene, which can improve the detection performance of strongly occluded car views. In general, the contribution-aware strategy does significantly improve the detection performance for occluded objects.

7.5 Detection Examples

The previous section has presented some detection examples by using the parts-based car detector with contribution-aware occlusion-handing. Fig. 7.15a offers an example of some generated FP detections. Strongly occluded car views significantly decreased the number of remaining features. In both examples, the only features that remain could not be clearly identified as features that belong to the front or back part of a car, e.g. the wheel. A wheel feature votes at the accumulation step for two completely different center positions for the car hypothesis. In strongly occluded car views, the score was not reinforced due to the limitation of β_{max}. Fig. 7.15b depicts an example of a precisely detected car object.

In contrast with [Makris et al., 2013], the missing amount of features was es-



(a) (b)

Figure 7.15: Detection Examples: The figure presents two detection examples of the parts-based car detector with contribution-aware occlusion-handling. (a) portrays two FP detections (marked in red) beside a correctly detected car (marked in green). These FPs can appear if the right part of a car is strongly occluded and the remaining features cannot be clearly assigned as front or rear car parts. In this case, car hypotheses to the left and right of the remaining features are generated. (b) depicts a true positive detection.

timated. The occlusion-handling in [Makris et al., 2013] calculates the predicted occlusion by taking into account the occlusion of 10 sub-parts of the car template (Fig. 7.16). With this, the authors estimate a uniform contribution of the features that are stored in the code-book. The occlusion-handling presented here takes into consideration the distribution of the features stored in the code-book in order to predict the amount of missing features. Therefore, a more precise re-weighting of the possible car hypothesis can be applied. This decreased the number of FPs that demonstrate less occlusion of features than the visible occlusion of the support window. Furthermore, it recognizes more cars exhibiting strong occlusion of features but less occlusion of the supporting window.

In [Makris et al., 2013], a filtering step is used to reduce the potential areas where a car hypothesis can occur. Because the filtering produces varying output, it is not possible to predict the computing time for one test image. A re-implementation of Makris et al. [2013] was used to calculate the computing time of one complete input image, which was the worst-case scenario, using the filtering. The re-implemented version requires more than 12 seconds[2] per frame, and also produces many FPs. Please note that this comparison is not completely fair, as it is partially unclear which parameter settings were used for the results presented in [Makris et al., 2013]. To conclude, Fig. 7.17 reports the overall detection performance of the visibility-based parts-based car detector without occlusion-handling compared to the contribution-aware occlusion-handling. The

[2]The evaluation was done on a single-core CPU and was implemented in Python.

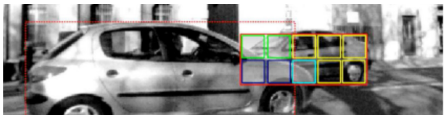


Figure 7.16: Comparison of the Occlusion-handling Strategy presented in [Makris et al., 2013] to the Contribution-aware Occlusion-handling: (a) Makris et al. [2013] first split the car template into 5×2 sub-parts. Depth information is used to mark at which sub-parts of the possible car hypothesis are visible. The predicted occlusion is used to reinforce the activation score of the car hypothesis. With this, the authors estimated a uniform contribution of the features that are stored in the code-book. (b) Here, 36×7 sub-parts are used. Additionally, the distribution of the features stored in the code-book is taken into account in order to predict the amount of missing features.

new detector outperforms the initial detector at 40, 60, and 80% occlusion. However, the results in Fig. 7.14 indicate that the detection performance cannot be improved for 80% occlusion as much as for 40 and 60% occlusion. Of course, this is due to the relatively low β_{max} used in this chapter, but also to the low amount of remaining features at a strongly occluded object view.

7.6 Conclusion

This chapter has discussed several occlusion-handling strategies for a parts-based car detector. Hence, it determined the limitations and optimal parameter settings of a parts-based car detector by evaluating a visibility-based occlusion-handling strategy. For this, it used the rendered data set described in Chapter 6 with pixel-level labeling information, including the mask information of the occluding object.

The result of this evaluation was used to configure the occlusion-handling of the car detector, which evidenced an improved detection performance at all occlusion rates. Therefore, it included a re-weighting of the score of possible car hypotheses by taking their occlusion rate into account. The proposed occlusion-handling strategy is applicable to other detection approaches that include an accumulation step and provide mask information about object instances, e.g. [Felzenszwalb et al., 2010, Pepik et al., 2013]. An improved strategy for occlusion-handling that boosts the detection performance, especially for higher occlusion rates, was also presented. The occlusion-handling considers the distribution of learned features at the car template in order to predict the score reduction. With this, a better re-weighting of the score of the possible car hypothesis can be calculated.

For the parts-based car detection framework, it is possible to build only one

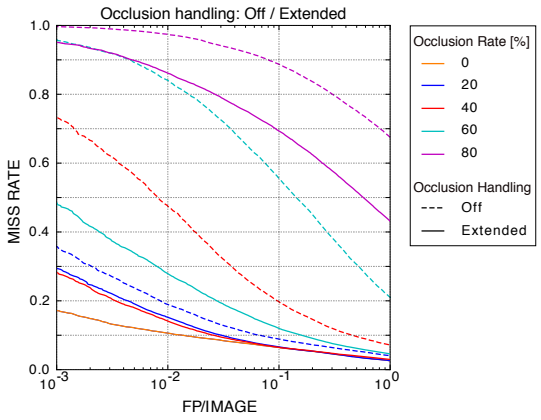


Figure 7.17: Detection Performance: The ROC plot illustrates the improved detection performance of the extended occlusion-handling strategy compared to the initial system. The evaluation used 17,600 test images for each occlusion rate separately. Additionally, it utilized 17,600 test images with occluding constellations without any cars. So, the occluding objects occlude the background. This resulted in a total of 105,600 test images.

code-book to recognize full-rotation views. This requires the use of some additional label information θ, which provides information about the angle of rotation for the object. The rotated views can be divided into groups of defined rotation angles. For example, the full-rotation view is divided into 12 groups, with each containing a rotation range of 30°, e.g. the first group ranges from 0 to 30° of rotation, the second group from 31 to 60°, and so on. At the code-book building step, the θ has to be stored. During detection, several car hypotheses are generated for the different θ. The winning hypothesis votes for the object center and the rotation view. Another possible extension of the occlusion-handling of the parts-based detection framework can incorporate a variable β_{max}, which takes the mask of the occluded object into account. For this, a segmentation of the complete scene is necessary to obtain the mask information for all object instances. By using the mask of the occluded object, the re-weighting of the score of the car hypothesis can be controlled. The current framework strongly re-weights the score around the occluding object. This is also the case if some parts of the occluded object are already outside of the support window. This can cause FPs if the center of the car hypothesis does not exactly match the ground truth after the re-weighting. The use of the mask of the occluded object can help to prevent this effect.

8 Rotation and Illumination Stability of Texture Descriptors

Chapter overview *The previous chapters have demonstrated several strategies for occlusion-handling. For all detectors, SIFT texture descriptors have been used for the feature extraction. SIFT was originally presented in 2004 by Lowe [2004]. Since this time, several authors have presented many other feature descriptors. Bay et al. [2006] have developed one extended version of SIFT, called SURF, which was unveiled as competitive to SIFT, but much faster. This chapter justifies the usage of SIFT descriptors within the presented detection frameworks. For this, it compares SIFT descriptors to SURF descriptors in terms of robustness against rotation and illumination changes. Additionally, it investigates the variance of extracted feature vectors depending on their surface location.*

8.1 Introduction

Object recognition systems consist of a sequence of processing steps. One of the first steps is the feature extraction. Here, feature descriptors are computed at defined positions of an image. This is a crucial part of the procession pipeline since all subsequent steps depend on the extracted information. Hence, the recognition performance strongly depends on the quality of the extracted features. Good features have to fulfill the following criteria:

1. A high robustness against rotation and illumination changes is required. Otherwise, an accurate matching is not possible.

2. In order to obtain a clear allocation of the object's center, each feature has to vary depending on its surface location.

Lowe [2004] initially presented the SIFT descriptor, which current detection approaches frequently use. Another descriptor is SURF, originally presented by Bay et al. [2006]. SURF is similar in many regards to SIFT, but has a lower computational complexity. Both descriptors focus on the distribution of gradient information. One difference is that SURF integrates the gradient information within each sub-patch, whereas SIFT depends on the orientation of the individual gradients. The inventors of SURF have stated that this leads to a higher robustness against noise.

Within the detection system presented in Chapter 7, it is possible to switch between SIFT and SURF descriptors. However, the usage of SURF did not yield satisfactory results. The occurrence maps revealed a broad stimuli distribution, precluding a reliable determination of the object center. This chapter analyzes the robustness of SIFT and SURF against rotation and illumination changes. It uses the squared Euclidean distance to measure the feature similarity in the same

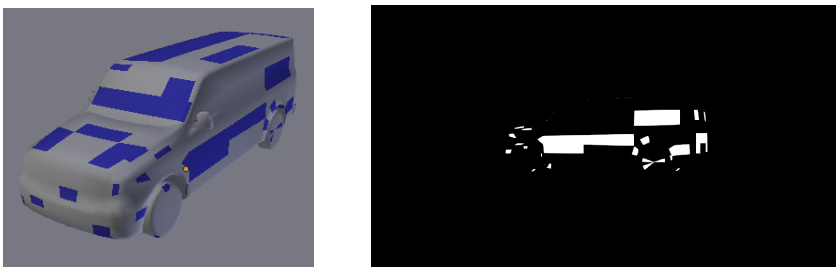
way as for the feature matching in Chapter 7. Additionally, it investigates whether extracted features can be reliably assigned to different locations on an object's surface.

8.2 Datasets

Chapter 7 proposed some strategies for improving the detection performance of occluded cars. The occlusion-handling was based on an estimated score reduction, which crucially depends on an accurately matched position. Rotation and illumination changes can influence the feature representation, and consequently the feature matching. Features that are robust against rotation and illumination changes can drastically simplify the matching. The position of a preselected feature is needed to analyze the rotation robustness. The idea is to apply a texture descriptor at this position during rotation and analyze how the feature representation changes. Once again, the render framework was used to acquire a precise position of the feature. The simplest way is to use the so-called "faces" of a 3D object as markers and track them during rotation. A face is the clamping surface between four points of a polygon skeleton. On average, the used car models were built of 15,000 faces. Using all these features resulted in an immensely high time-complexity. A uniform distribution of the faces on the object's surface was provided for complexity reduction, and k-means clustering was performed to locate these faces. The total number of faces divided by 10 defined the number of clusters. Each face was encoded by its center of gravity position. Fig. 8.1a highlights the corresponding faces of all clusters for an exemplary car. The face with minimum Euclidean distance was assigned to each cluster. An alpha mask was used to generate a mask that indicated the visibility of the selected feature in the scene. Fig. 8.1b displays the mask after rendering of the marked faces of Fig. 8.1a.

A mask at a full rotation was generated for each marked face. Therefore, the car could be rotated in a step-width of 10°. At each rotation step, the illumination conditions were randomly changed to see the robustness against these changes. Fig. 8.2 presents exemplary masks of a car's front light. The example depicts the generated masks for the feature.

In addition to the masks, pixel coordinates for the world coordinates of the balance point of the faces were stored to identify the exact position at the rendered image. Rotation views with masks that did not cover the balance point were filtered, leaving only rotation views of the marked features that included pixel information of the feature. The remaining views of the marked feature were stored at a segment.



(a) (b)

Figure 8.1: Face Selection after the Clustering Step: (a) depicts faces matched to the cluster centers marked in blue. The cluster centers are uniformly distributed over the object's surface. (b) displays the masks generated by the render framework.

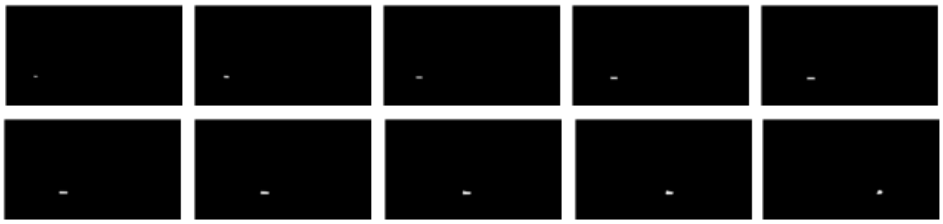


Figure 8.2: Mask Generation of a Single Face: The exact position of a marked feature can be determined with the render framework.

8.3 Proof of Rotation and Illumination Robustness of Texture Descriptors

The segments of the feature representation that are generated by the texture descriptors were utilized to analyze robustness against rotation and illumination changes. This entailed using the segments from the previous section and computing the feature vector with SIFT and SURF as texture descriptor, respectively. The generated feature vector of a single-feature view was used to compute the quadratic Euclidean distance to all other rotation views of this feature. This was done for each rotation view of the feature. The result was plotted in a diagram, which illustrated the quadratic Euclidean distances of all features separately from the other rotation views. Fig. 8.3 reports the evaluation of two characteristics of around 1,500 features. The other features exhibit similar results. The evaluation was completed for 15 distinct cars models, all evidencing the same resulting characteristics. The top line of Fig. 8.3 shows the patterns of the different features at

various rotation views. The left side depicts a feature at a front light, while the right side illustrates a feature located at the rear window. Of the 36 total rotation views, the front light feature is visible in 13 views, while the other feature is visible in 14 views. As mentioned, the quadratic Euclidean distance was utilized to proof the robustness of the extracted features. 8.3a presents the distance plot of the front light feature by using the SIFT texture descriptor. For example, the first top line in the plot signifies the quadratic Euclidean distances of the feature extracted at 50° of rotation compared to all other selected rotations. Low distances are marked in red, while high distances are marked in blue. In other words, a line in the plot with a high number of red-marked values corresponds to rotation-stable features. These features are also robust against illumination changes, because the illumination at the rotation also changed. Fig. 8.3a demonstrates a rotation- and illumination-stable feature by using SIFT as the descriptor. Fig. 8.3c displays the results of the same feature using SURF as the descriptor, which seem to be more affected by the rotation and illumination changes. Fig. 8.3b reports the same analysis using SIFT for a feature located at the rear window of a car. Features extracted at a range of $260 - 330^\circ$ of rotation are robust against this rotational change, which is represented by the red-marked boxes in the plot. At other rotations, the extracted feature representation exhibits high distances to other features. However, the mentioned range of $260 - 330^\circ$ is sufficient to build a detector that recognizes car objects at side views. Fig. 8.3d presents the evaluation using SURF. The extracted features seem to be more stable compared to SIFT.

In sum, both descriptors evidence nearly the same robustness to rotation and illumination changes. In few cases, the SURF descriptor generates less robust feature representations compared to SIFT, but they are still sufficiently robust for detection. This result does not explain the complete failure of the car detector using SURF as the texture descriptor. However, it is important that the generated descriptors are unique for different locations on the surface. The next section analyzes the variation of generated descriptors at different positions on an object's surface.

8.4 Discriminative Features

As mentioned, in addition to robustness of rotation and illumination changes, the extracted features located at different positions of an object have to differ from one another. This is imperative for a precise voting of the position of the object's center. This section offers an analysis to proof this condition. For the sake of simplicity, a feature with a fixed position on the object surface is referred as an

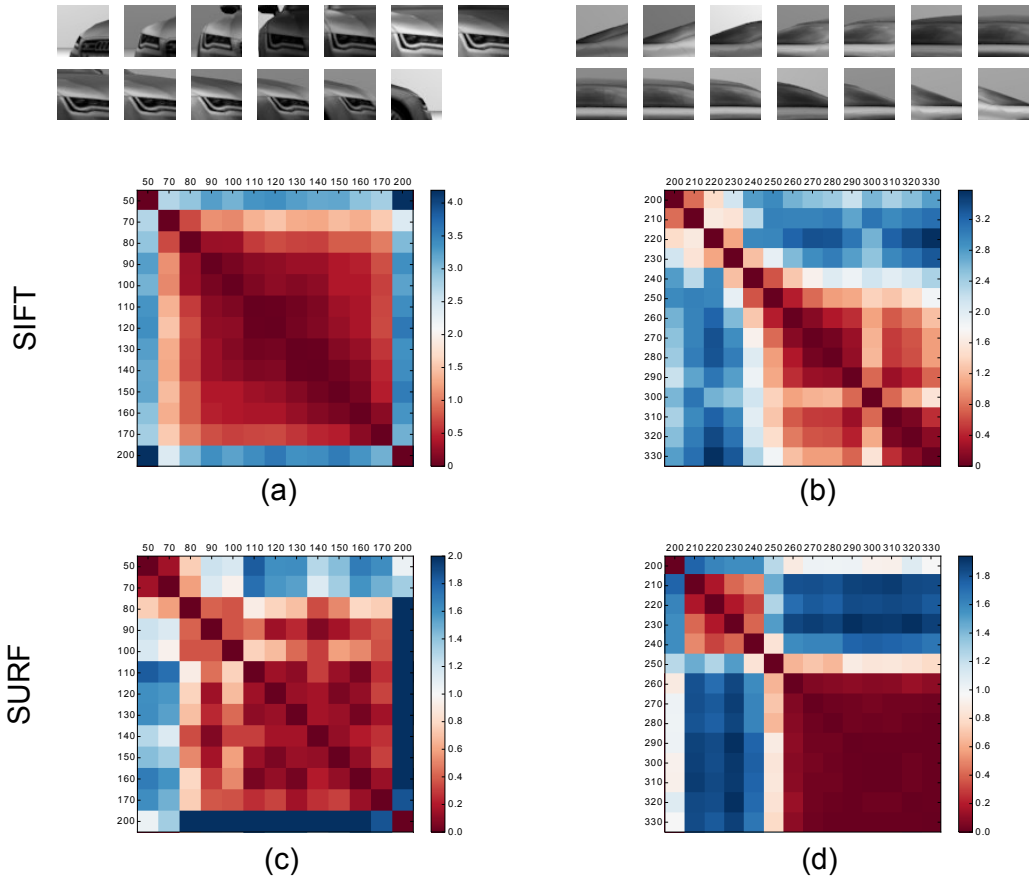


Figure 8.3: Rotation Stability of a Feature: The upper line signifies the receptive field of the feature descriptor following the selected face during rotation. The rotation changes from the left to the right by adding 10° of rotation. (a) and (b) display the quadratic Euclidean distance of each combination of SIFT features of the trajectory. A low value for the distance is marked in red, while a high value is marked in blue. (a) represents a stable feature, while (b) represents a less stable one. (c) and (d) present the same analysis using SURF as the descriptor. (c) seems to be more affected than (a) by illumination and rotation changes, while (d) seems to be more robust than (b).

object feature, e.g. an object feature can be the front light feature at different rotation views.

The evaluation used the most stable rotation view of each single object. For example, Fig.8.3a and c display rotation stability of the front light feature. The row with the lowest sum indicates the most stable rotation view of this feature. Afterwards, the Euclidean distance among all most stable features was calculated. Fig. 8.4 depicts the result for the SIFT descriptors. For example, the first line

illustrates the front light feature. The higher the value, the more distinguishable an object feature is from the others. The plot reveals many discriminative object features. This means that the SIFT descriptor generates feature vectors that can be clearly matched to an object feature. So, a precise vote for the object's center after the feature matching is possible.

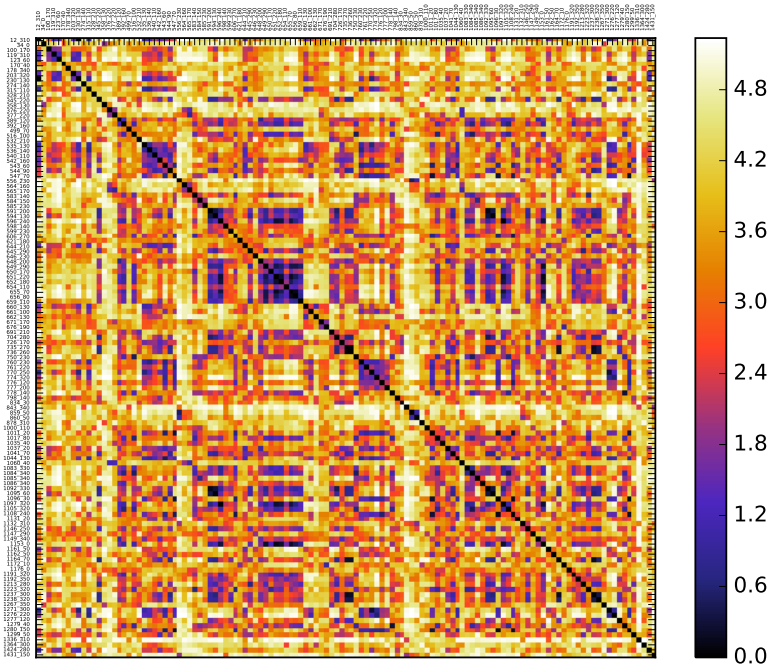


Figure 8.4: Discrimination of SIFT Features: The quadratic Euclidean distance was used to calculate the differences of the features. Low values are marked in black or violet and correspond to non-discriminative features, whereas orange and yellow correspond to highly discriminative features.

Fig. 8.5 presents the same evaluation for the SURF descriptors. Please note that the scale of the values was drastically reduced and ranges between 0.0 and 1.8, whereas the values for SIFT descriptor ranged between 0.0 and 4.8. Overall, the differences are significantly lower in comparison to the SIFT descriptors. Thus, the determination of the object's center is less precise.

The results indicate that the SIFT descriptor is a better choice for the parts-based car detector. The quality of the occurrence maps that were generated for the car detector using SURF as a texture descriptor are encouraging of this observation. The maps evidence a widespread distribution of the cluster stimuli.

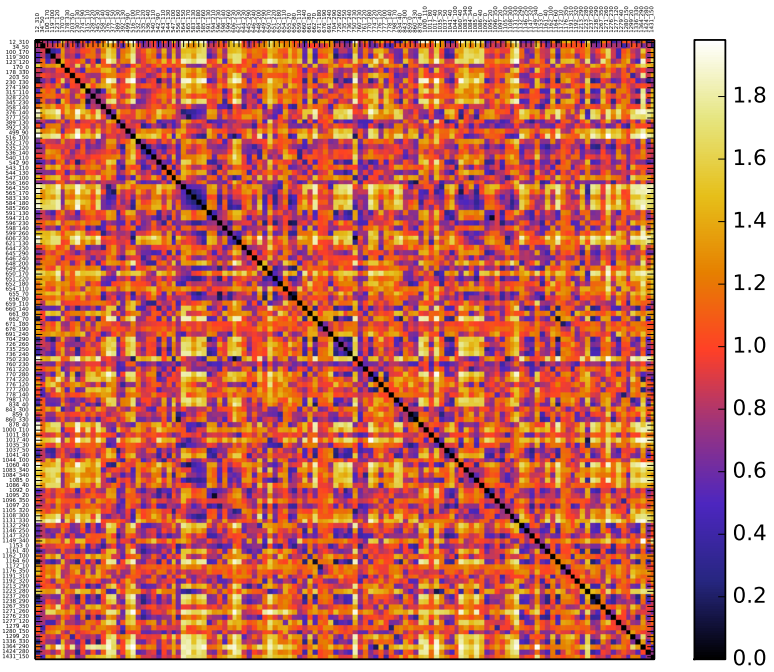


Figure 8.5: Discrimination of SURF Features: The quadratic Euclidean distance is used to calculate the differences of the features. Low values are marked in black or violet and correspond to non-discriminative features, whereas orange and yellow correspond to highly discriminative features.

With these maps, it is impossible to vote for a clear position of the object's center.

8.5 Conclusion

This evaluation has demonstrated that both feature representations are stable against rotation and illumination changes. However, the center of the object can be more precisely determined with the SIFT descriptors. SURF descriptors cannot be robustly assigned to a position of an object, and therefore lead to an inaccurate center estimation. This was the main motivation to use SIFT descriptors for the parts-based car detector.

9 Psychophysical Study on Object Detection of Occluded Objects

Chapter overview *This chapter investigates how human beings deal with occlusions by detailing a small-scale eye-tracker study[1] completed for this research.*

9.1 Occluded Object Recognition by Humans

The previous chapters have discussed several strategies to improve the detection performance of object recognition models, especially for strongly occluded object views. However, conceptualizations of these models were motivated by the detection results of an initial system without analyzing how humans deal with the challenging task. Human beings are able to recognize objects in complex scenes, and are especially adept at detecting occluded objects. Psychophysical studies have reported that the information of the occluding object significantly influences the detection time. In [Fukushima, 2001], the authors performed a psychophysical study on the recognition of occluded objects. In this study, participants viewed letters with erased parts (Fig. 9.1 upper-line) and occluded parts (Fig. 9.1 underline). The authors observed that the participants required more time to recognize the letters with erased parts. Based on this findings, they constructed a neural network model. During recognition, the model struggles to distinguish which features belong to the original pattern if the occluding object is erased. However, the network easily identifies the object features if the occluding object is visible. With this, the system simulates the observed behavior of the participants.

Another study by Johnson and Olshausen [2005] has identified a similar effect. In this study, the authors validated this effect with a simple test image configuration and used an electroencephalography (EEG) to measure the recognition time. They presented two different occlusion constellations to analyze the use of the occluding object. The occluding object is visible in the first constellation, whereas in the second constellation, the occluding object is replaced with background clutter. Fig. 9.2 offers two examples of objects that had to be recognized. At the left, the objects are depicted with some occluding shapes, while at the right, the occluding shapes are replaced with some cluttered background information. The participants were able to recognize occluded objects faster when the occluded object was visible.

During the experiment, the participants were required to recognize an object on an artificial cluttered background filled with ellipses. The ellipses were complex enough as distractors, but included no accidental features. Before each trial, a

[1] The study was performed at the University of Bielefeld. I want to thank Lukas Twardon, Dennis Wobrock, Andrea Finke, and Stefan Genster for supporting me at the eye-tracker study.

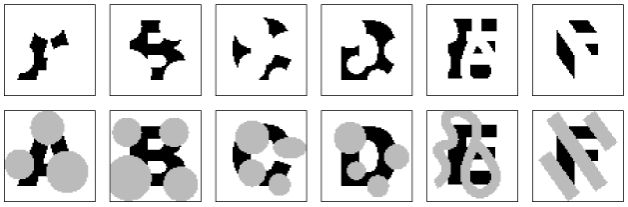


Figure 9.1: Psychophysical Study by Fukushima [2001]: (Upper-line) illustrates some patterns in which parts are cropped out and no occluding objects are visible. (Bottom-line) The same patterns with occluding objects are depicted. Test persons recognized the objects with visible occluders more quickly than the objects with cropped out parts.

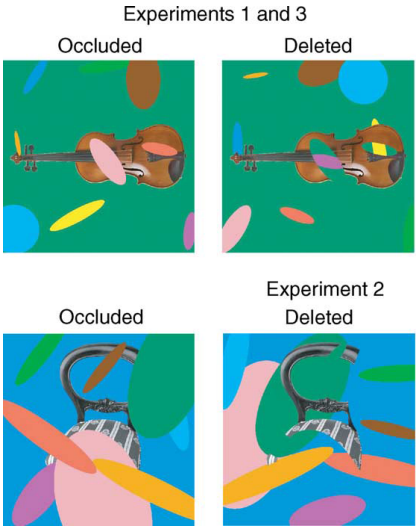


Figure 9.2: Psychophysical Study by Johnson and Olshausen [2005]: Both objects - the violin and the chair - are first presented with some occluders, then subsequently with some cropped parts. Test persons recognized the objects with visible occluders more quickly than the objects with cropped out parts.

word cue informed participants about the target object. After viewing the test image, the participants were prompted to respond as quickly as possible whether or not the object in the test image corresponded to the target cue. The results indicate that, in both cases, the time until recognition increased as the amount of missing pixels increased.

One possible explanation for this effect is the use of the so-called “occlusion boundary.” Occlusion boundaries are the parts of the contour of an occluding object that cover the occluded object. If this occlusion boundary can be clearly identified, then the occluded object can be recognized more quickly.

Bushnell et al. [2011] have evidenced that occlusion in daily scenes also pro-

duces “accidental” contours at the junction between the occluded and the occluding object, as well as that the brain is able to suppress these contours. If there is a small gap between the objects, the observed suppression is reduced. In Chapter 7, an improved detection performance by suppressing these accidental features of the occluding object was revealed by calculating the activation γ'. The presented approach could also benefit from the suppression of accidental contours because the shapes utilized in the experiments might negatively influence the result of the detector.

Meng and Potter [2008] have reported that occlusion affects detection and recognition differently. In their experiments, the participants were assigned to one of two groups. The first group, named the detection group, was instructed to look for a named target picture (e.g. “businessmen at table”). After viewing a picture sequence, the participants had to decide if they had seen a picture that fit the target description. The same picture sequences were presented to the participants in the second group, named the recognition group, but without any target description. After the test sequence, they received the test picture and had to decide whether they had seen the picture in the sequence. Both groups viewed the same picture sequences with and without occlusions. The performance was consistent between both groups regarding un-occluded pictures. In contrast, the performance of the recognition group was significantly lower for occluded pictures. The results of the experiments confirm that a top-down approach, e.g. to first search for easy-to-detect objects, and afterwards for the difficult ones, can improve the detection performance of occluded objects.

This chapter describes a psychophysical detection study that was conducted to determine which information human beings use for the detection of occluded and un-occluded objects. An eye-tracker was used to determine the explored areas of the image viewed by participants. Participants had to count car objects in a test image with un-occluded and occluded views of cars and non-car objects.

9.2 Dataset

This section describes the characteristics of the dataset that was presented to participants during the study. For the experiment, the rendered car segments specified in Chapter 6 were used as recognition objects. Additionally, a data set with views of non-car objects was created. Fig. 9.3 depicts some non-car examples.

The car and non-car objects used as potentially occluded objects were placed on top of a cluttered background image with a size of 1680×1050 pixels, which is the maximal resolution for the screen that was used at the experiment. The total

Figure 9.3: Dataset with Non-Car Objects: A data set with randomly chosen condition for light position, light intensity and angle of rotation is generated.

number of possibly occluded objects varied between two and five. Thereby, the object type was randomly chosen with the restriction that at least one had to be a car. Also, the positions of the objects were generated randomly. Although the objects were randomly placed, it was ensured that only configurations in which the objects did not overlap each other were selected. For each potentially occluded object, it was randomly decided if the object would be occluded by another object or not. Only non-car objects served as occluding objects. Taking into account the accuracy of the eye-tracker and the size of the so-called “foveal” area, a minimum object width of 450 pixels and a minimum height of 400 pixels was used. The foveal vision area is the surrounding area of a fixation. Inside this area, human beings are able to recognize objects.

Inspired by Galerne and Gousseau [2012], the cluttered background included artificial constellations of filled ellipses. Ellipses avoid incorporating any accidental features without oversimplifying the detection task. The ellipse shapes were randomly generated with regard to their color, width, aspect ratio, rotation, and position. The length varied from 10 to 100% of the size of a car object. The artificial background was necessary because a simple background would make the recognition task too easy for participants.

Experiments with eye-trackers require focused participants. A lack of focus significantly influences the results and is especially likely to occur when participants become exhausted. Therefore, the experiment was limited to 100 test images. Each object constellation was included twice during the experiment. The occluded objects were visible the first time (Fig. 9.4a), whereas the second time, the background was simply used to occlude the objects. Additionally, the position of the objects and their orientation along the vertical axis (Fig. 9.4b) were reversed,

to prevent participants from utilizing previously seen constellations. Fig. 9.4c and Fig. 9.4d present the corresponding mask information.

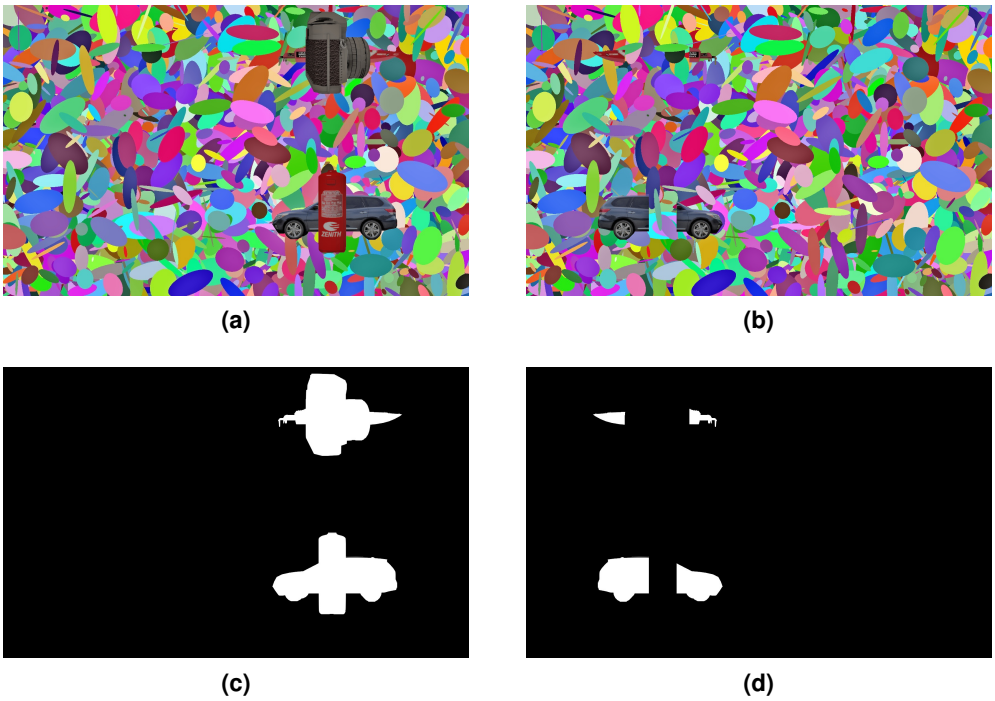


Figure 9.4: Test Images for the Eye-Tracker Study: (a) depicts one occluded car with a non-car object as the occluding object, and a non-car object occluded by another non-car object. (b) displays the same objects, but instead of the features of the occluding object, it shows the cluttered background. The objects and their positions are reversed to avoid having participants recognize the same constellations. (c) presents the mask information for all object instances in the image, while (d) shows the mask information of the occluded objects.

The mask information is necessary in order to count the number of fixations on the object. Exemplary masks are visible in Fig. 9.5.

9.3 Experimental Setup

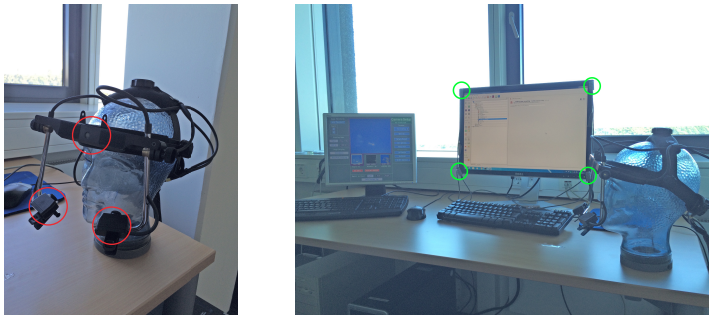
9.3.1 Physical

During the experiment, the test images were displayed at a resolution of 1680×1050 pixels on a light-emitting diode (LED) screen. The distance between the screen and the participants was approximately 60–70 centimeters. The exper-

(a) (b) (c)

Figure 9.5: Additional Mask Information: (a) shows the mask of the occluded objects. (b) shows the mask of the occluding objects while (c) shows the car object only.

iment used the eye-tracker EyeLink II, from SR Research,[2] a head-mounted video-based system consisting of three cameras. The cameras were used as follows: (i) a separate camera was used for each eye to record the movements of the pupil (Fig. 9.6a). (ii) the third camera was used for optical head tracking and was directed at the screen and (iii) LED lights were used as markers for the calibration because they are easy to detect in video-signal images. These LED lights were fixed at each corner of the LED screen (Fig. 9.6b). The experiment was conducted in a darkened room, and the eye-tracker was calibrated for each participant in order to track the position of the pupil. For the sake of simplicity, only the left eye was used for the tracking. Fig. 9.7 offers a screenshot of the calibration software. The blue circle highlights the detected pupil.



(a) (b)

Figure 9.6: Experimental Setup: (a) depicts the EyeLink II with its three cameras. The cameras are marked with red circles. (b) portrays the experimental setup. The LEDs at each corner of the screen are marked with green circles.

[2] http://www.sr-research.com/eyelinkII.html

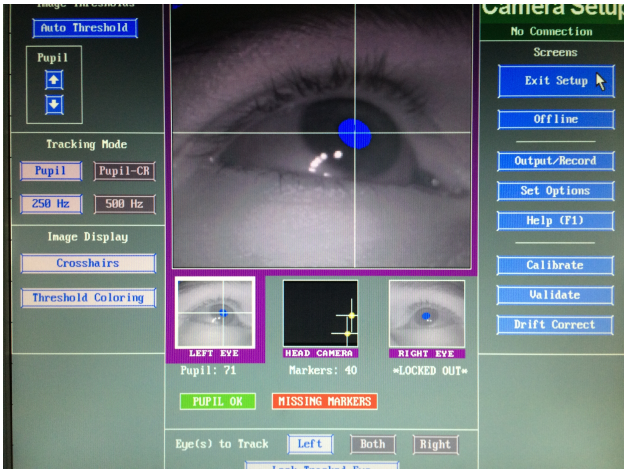


Figure 9.7: Eye-Tracker Calibration: Before beginning the test process, a calibration was performed for each participant. For this, the threshold for the pupil detection was adjusted for each participant prior to the test. The blue circle depicts the detected pupil.

9.3.2 Participants

Twelve students within the age span of 20 to 30 years participated in the study. Seven were female and five were male. Despite the calibration of the eye-tracker, the pupil detection repeatedly failed for two female participants. Hence, their recordings were excluded from the evaluation.

9.3.3 Test Procedure

A simple task was define in order to avoid effects in the measured data from the test sequence itself. The participants were requested to count the number of cars in several images without any time limitation. During the experiment, the participants received no feedback about their reported number of counted cars. They also received no reward or punishment. The experiment was designed as follows:

1. A welcome screen showing the task and the control setup was displayed.
2. Before each image, a shift correction was performed by displaying a white cross in the middle of a black screen. The participants had to press the space bar while looking at the cross.
3. A test image was shown. The participants had to press the space bar before they were able to enter the number of counted cars.

4. A screen with buttons from one to five was shown. The participants had to click on the button with the detected number of cars, and had to click on a confirmation button to confirm the selection..

5. Steps two through four were repeated until the end of the study. In total, 100 images were shown in a random sequence to each participant. One experiment was completed for each participant.

A log file was generated per experiment, including the time stamp of appearance and disappearance for each test image. Furthermore, the duration and position of the fixations and saccades were stored. A visualization of an exemplary eye trajectory can be seen in Fig. 9.8.

The next section demonstrates how the recorded information was used for the evaluation.

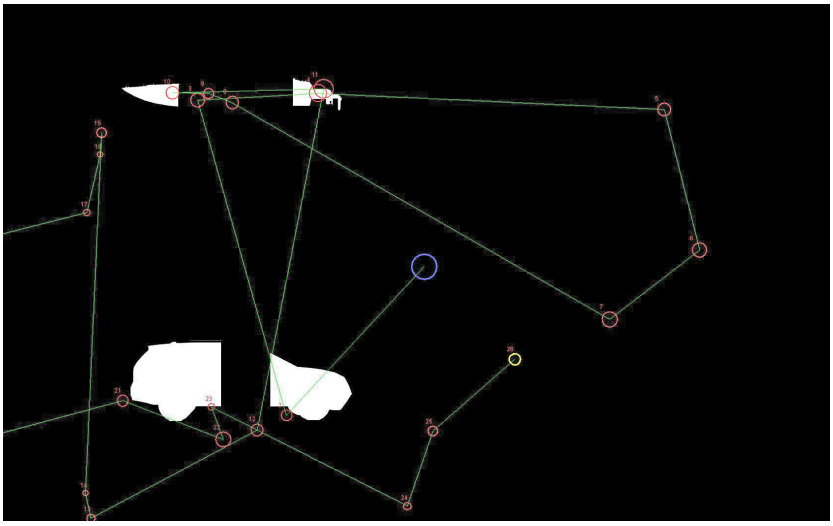
9.4 Evaluation

During the experiments, all participants counted the cars correctly. For evaluation, it was necessary to count the number of fixations on car and non-car objects, on the background, on the intersection area, and on the occluding object. This was done for each test image. Fig. 9.9 presents an example of an occlusion constellation. The orange-marked areas indicate the visible parts of an occluded car, while the occluded parts are marked in green. The green areas are referred to as intersection areas. The combination of the blue and green areas is the shape of the occluding object. The background is marked in black. A calculation was made of the average detection time of the 50 images for all 10 participants divided by the number of occluded objects.

The results in Tab. 9.1 reveal that, in both occlusion constellations, the participants required the same amount of time for the detection of all cars. The constellation in which the occluding object was visible is referred to as VS, whereas NVS is used for the constellation that utilized the background as the occluding object. In case of the NVS, there were less fixations on the occluding object and more on the background compared to VS. This means that the participants explored the scene more broadly when the occluding object was not visible. Participants focused more on the occlusion itself when the occluding object was visible. Therefore, they seemed to use the occluding object. However, the recognition time was the same for both constellations.

The duration of the fixations was also analyzed. Tab. 9.2 displays the average fixation time per occluded object for both constellations. The total duration was

(a)



(b)

Figure 9.8: Exemplary Eye Trajectories of the Study.: (a) shows the saccades and the fixations on a test image while (b) shows the corresponding mask. Saccades are marked with green lines while the fixations are marked with circles. The blue circle shows the first fixation while the yellow circle shows the last fixation. All other fixations are marked in red. The size of the circle reflects the time of the corresponding fixation.

nearly the same, regardless of whether the occluding object was visible or not. Hence, the results do not confirm the effect described by Johnson and Johnson and Olshausen [2005] and Fukushima [2001]. In both of these publications, the

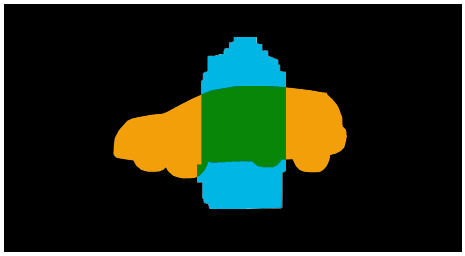


Figure 9.9: Occlusion Constellation of Objects: The orange areas are the visible areas of the occluded car. Blue and green areas combined represent the occluding object. The occluded area of the car is marked in green.

Average Number of fixations per Object						
	Sum Fix	Cars	NCars	Bg	Occluder	Intersec
NVS	4.40	0.54	0.24	2.31	1.32	0.59
VS	4.42	0.56	0.27	1.97	1.62	0.68

Table 9.1: Number of Fixations: The table displays the fixations at car objects (Cars), non-car objects (NCars), the background (Bg), the occluding objects (Occluder), the intersection area (Intersec) - which belongs to the occluding object - and the total sum (Sum Fix). The number of fixations is specified for each occlusion case separately. Visible occluding objects are referred to with VS, while NVS is used when background information was shown instead of the occluding object.

authors noted a longer recognition time for non-visible occluding objects. However, also evidence of an increased fixation time on the background for non-visible occluding objects, is reported.

The results suggest that the participants made use of the occluding object to focus on object constellations. A more detailed analysis of the fixations within the occlusion boundaries can impart a deeper understanding of this effect. The eye-tracker returned one pixel coordinate for each fixation. This pixel coordinate did not cover the complete receptive field of humans. In order to overcome this limitation, the foveal area was taken into account. The thickness of the occlusion boundary was iteratively increased from 1 to 250 pixels in order to simulate different sizes of the foveal area. Fig. 9.10 depicts examples of the occlusion boundary with increasing thickness.

The graph in Fig. 9.11 illustrates the fixation time for a corresponding boundary thickness for both constellations. An increased thickness clearly leads to an increased fixation time, because the number of fixations at the occlusion boundary also increases (Fig. 9.11a). It is evident that both curves are nearly identical

Average Fixation Time in ms of Fixation per Object							
	Sum Fix	Cars	NCars	Bg	Occluder	Intersec	Sum Sac
NVS	710.97	86.50	35.92	375.70	212.85	98.51	339.87
VS	749.41	99.39	48.78	326.35	274.90	99.39	300.52

Table 9.2: Overview of the Average Number of Fixations: Fixations are counted for car (Cars), non-car objects (NCars), the background (Bg), the occluding objects (Occluder), and the intersection area (Intersec). The number of fixations on the occluding object always contains those on the intersection area. The time is specified for each occlusion case separately. VS refers to the constellation with the visible occlusion object, whereas NVS represents the non-visible one.

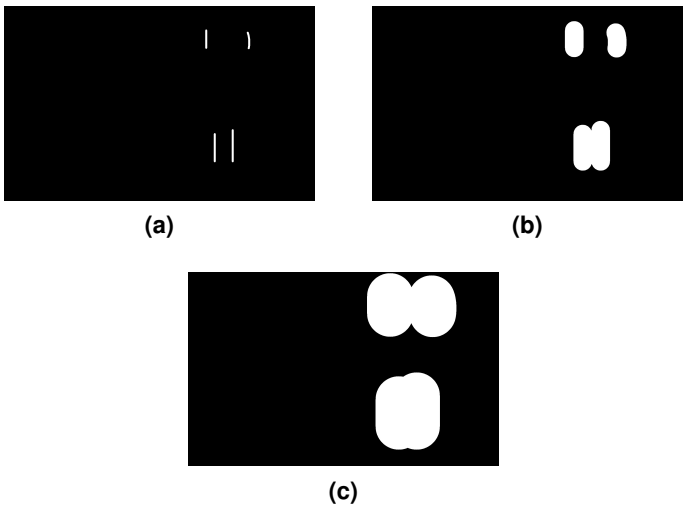


Figure 9.10: Masks of the Occlusion Boundary: (a) shows the occlusion boundaries with a thickness of 10 pixels. (b) shows the use of 100 pixels and (c) the use of 250 pixels for the thickness of the occlusion boundaries.

until a border size of 80 pixels. However, for a higher boundary thickness, the VS curve rises more strongly compared to that of NVS since the participants explored the whole scene more thoroughly in case of NVS. The same effect is apparent in the number of fixations, as shown in Fig. 9.11b.

The difference of both curves is also specified in Fig. 9.12a and Fig. 9.12b. In order to calculate the delta, NVS was subtracted from VS. A peak is visible for a thickness of 52–58 pixels. The foveal vision area can explain this effect, which leads to the conclusion that the participants did not directly focus on the boundary, but rather recognized it within their foveal areas. Afterwards, the difference between VS and NVS increases in both plots. This is because the area of the occlusion boundary covers fixations at the occluding object in case of VS,

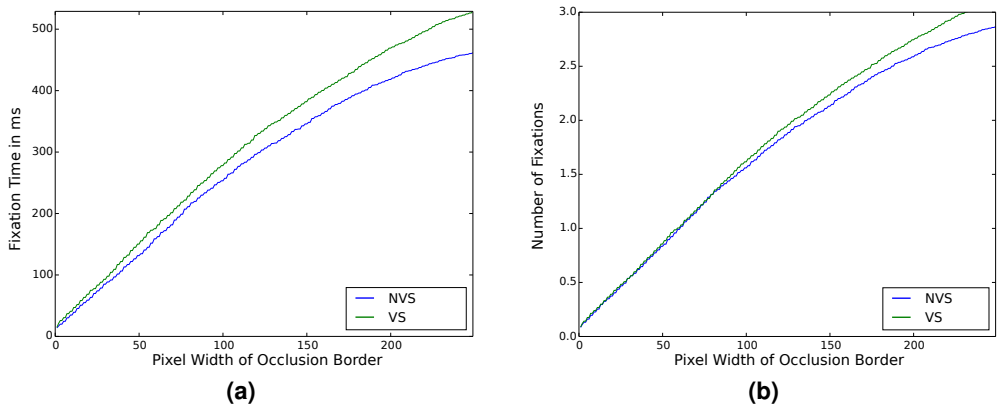


Figure 9.11: Fixations along the Occlusion Borders: (a) depicts the summed fixation time for VS and NVS. (b) indicates the corresponding number of fixation.

whereas fewer fixations are located within this area for NVS.

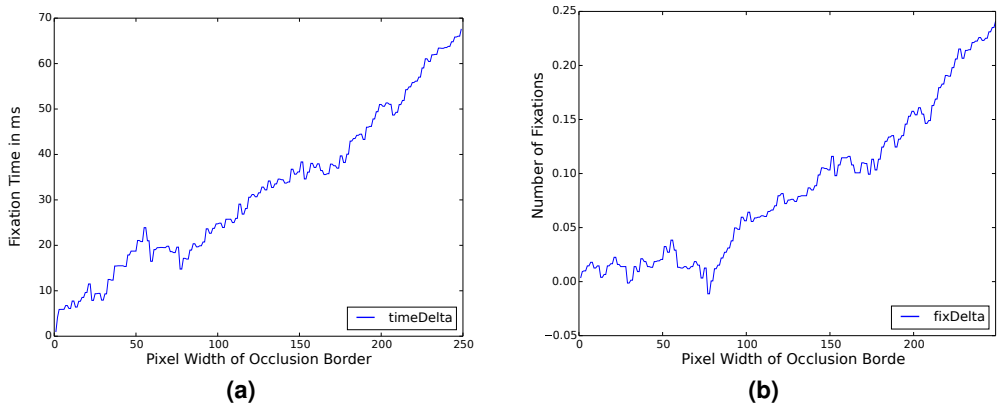


Figure 9.12: Fixations along the Occlusion Borders: (a) charts the difference of time of the fixations for VS and NVS. (b) represents difference in number of fixations for VS and NVS.

The evaluation of the recorded data reveals that the participants made use of the occluders in the scene. If participants recognized an occluding object, then they focused on the area surrounding this object, as evidenced by the increased number of fixations on this area. In contrast with this, participants explored the scene more extensively if the occluder was not visible, which is deduced from the increasing number of fixations on the background. Although the number of fixations on the car was consistent, it is apparent that the number of fixations on the intersection area decreased. This suggests that the participants had some

difficulties with combining the visible parts of the occluded object and systematically searching in the vicinity for more object information. If the occluder was visible, then the participants made a targeted search. The described effects are also demonstrated by the summed time of fixations for the different object types. In general, the simple presence of an occluder led participants to engage in a more systematic search and made it easier to join objects parts together.

9.5 Conclusion

This chapter has presented the design, experimental setup, and results of an eye-tracker study with 10 participants. Additionally, it has mentioned the incorporated restrictions for the dataset generation. The observed use of the occluder can be integrated into occlusion-handling strategies. One idea is to take into account occluded objects in the surroundings of occluding objects. Thereby, it is not necessary to recognize the occluding object. Such strategies can also be found in literature in which depth information serves an important role (Makris et al. [2013], Stückler and Behnke [2013]). Chapter 7 has demonstrated that the use of the mask information of an occluding object can significantly improve detection performance.

However, it is not possible to confirm the observation of Fukushima [2001] and [Johnson and Olshausen, 2005] that participants need more time for the detection task when the occluder is not visible.

10 Conclusion

This thesis has presented several methods to improve the recognition performance of car detectors during occlusion. The approaches can be distinguished according to two main directions. The first type of approaches for handling occlusion integrate occlusion-handling strategies at the training step or in the fixed compositional structure of the classifier. Such solutions exhibit limited generalization if new occlusion constellations appear during testing. However, it is the only way to integrate occlusion-strategies in holistic discriminative approaches. The second direction uses an explicit modeling of the occluder influence on the detection and is thus independent of the trained occlusion constellations. Hence, the system does not have to be trained again if new occlusion constellations must be recognized. Such occlusion-handling can be most easily integrated in parts-based detection approaches.

At the beginning of this thesis, the focus was on holistic discriminative detection approaches. The reference model in Chapter 3 indicates superior detection performance for non-occluded car views, but fails for more occluded views. Motivated by an analysis of the occlusion configurations of the annotated video streams, Chapter 4 presents a new hybrid approach combining detections trained on unoccluded and occluded cars. This approach can exploit the strong dominance of mutual car-car occlusions. It was shown that the classifier improves the detection performance. However, the framework is strongly specialized to one class of car-car constellations. Although, the framework can in principle be extended to any other object-object constellation that is known during training, the framework requires taking into account all possible configurations during training.

In view of this limitation, Chapter 5 develops a two-stage detection framework that can make use of more general occlusion patterns. A vertical split of the receptive field of the holistic car template was performed, and each resulting part-classifier was trained with un-occluded car views. With this, each classifier is forced to make a more local decision about the presence of the car, and is later not affected by occlusion of another part. It was demonstrated that the classifier improves the detection performance for non- to medium occlusion rates. However, for strongly occluded car views, no improvements have been seen. A performance increase for this most difficult condition could be reached only by taking a complementary feature like stereo depth into account. Using mean depth values as features enabled the detector to reject more implausible feature configurations. The framework can transfer to other application areas that exhibit structured occlusion that can be modeled by a sub-division of the holistic template. Despite the integration of all described occlusion-handling strategies, holistic discriminative approaches suffer from bad generalization to occlusion constellations not seen in the training data. Because of this insight, the second half of this thesis

concentrates on parts-based detection approaches.

One big problem with available ground truth training data is that the labeling of hand-annotated video streams often does not deliver accurate object instance information for a scene, which is a drawback if occlusion-handing strategies must be analyzed. To address this, Chapter 6 has presented a rendered benchmark data set. The data set provided pixel-level object instance information.

Chapter 7 first presented all detection results of a parts-based object detector. The framework exhibited some detection problems at higher rates of occlusion. A simple occlusion-handing strategy that makes use of the mask of the occluding object to reinforce possible object hypotheses was reviewing in order to overcome this limitation. The occlusion-handling improved the detection performance at all rates of occlusion. Furthermore, the chapter presented an extended occlusion-handling strategy, which especially indicated an improved detection performance for strongly occluded car views by taking the amount of missing features into account. Therefore, mask information of the occlusion constellation was integrated in the accumulation step of the framework. For the two parts-based approaches with occlusion-handling, no re-training of the classifiers was necessary because the occlusion-handling strategies were integrated at the detection step. This is a substantial benefit compared to holistic discriminative approaches, which require re-training to include occlusion-handling. Due to this flexibility, occlusion-handing can transfer to any other object recognition model that includes an accumulation step.

The quality of the feature descriptors is essential for parts-based models of recognition. Therefore, Chapter 8 performed an in-depth an analysis of two state-of-the-art feature descriptors. Thereby, the focus was on robustness against rotation and illumination changes. This chapter has also provided an analysis of how discriminative the features are in comparison. This revealed that SIFT [Lowe, 2004] features are more suitable for generating a car template than SURF [Bay et al., 2006] features.

In order to understand the strategies that humans use for their superior recognition performance also for occluded objects Chapter 9 described a psychophysical study to explore how human beings deal with occlusion during recognition. The results suggest that humans make use of the presence of an occluding object to reason about object parts during detection.

Overall, the novel work in this thesis has shown that standard recognition methods can be enhanced by an explicit reasoning about objects and their occluders. This is also in good agreement with findings about human recognition in cluttered conditions. The greatest remaining challenges are the considerable variability of objects under threedimensional rotation and illumination changes, which was not

covered in this thesis.

Bibliography

Bay, H., Tuytelaars, T., and Gool, L. V. (2006). Surf: Speeded up robust features. In *2006 European Conference on Computer Vision (ECCV)*, pages 404–417.

Bo, L., Ren, X., and Fox, D. (2014). Learning hierarchical sparse features for RGB-(D) object recognition. *I. J. Robotics Res.*, 33(4):581–599.

Borenstein, E., Sharon, E., and Ullman, S. (2004). Combining top-down and bottom-up segmentation. In *2004 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, page 46.

Bushnell, B., Harding, P., Kosai, Y., and Pasupathy, A. (2011). Partial occlusion modulates contour-based shape encoding in primate area V4. In *J Neurosci, 31*, pages 4012–4024.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: binary robust independent elementary features. In *2010 European Conference on Computer Vision (ECCV)*, pages 778–792.

Caron, L., Song, Y., Filliat, D., and Gepperth, A. (2014). Neural network based 2d/3d fusion for robotic object recognition. In *European Symposium on Artificial Neural Networks (ESANN)*, pages 127–132.

Ciresan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12):3207–3220.

Collet, A., Martinez, M., and Srinivasa, S. S. (2011). The MOPED framework: Object recognition and pose estimation for manipulation. *I. J. Robotic Res*, 30(10):1284–1306.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893.

Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2009). Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 304–311.

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

Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D. (2013). Visual object detection with deformable part models. *Commun. ACM*, 56(9):97–105.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. A., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645.

Fidler, S., Mottaghi, R., Yuille, A. L., and Urtasun, R. (2013). Bottom-up segmentation for top-down detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3294–3301.

Fukushima, K. (2001). Recognition of partly occluded patterns: a neural network model. *Biological Cybernetics*, 84(4):251–259.

Galerne, B. and Gousseau, Y. (2012). The transparent dead leaves model. In *Advances in Applied Probability*, pages 1–20.

Gao, T., Packer, B., and Koller, D. (2011). A segmentation-aware object detection model with occlusion handling. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1361–1368.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361.

Gould, S., Baumstarck, P., Quigley, M., Ng, A., and Koller, D. (2008). Integrating visual and range data for robotic object detection. In *2008 European Conference on Computer Vision (ECCV) - Workshop on Multi-camera and Multimodal Sensor Fusion Algorithms and Applications*, pages 26–29.

Hasler, S., Wersing, H., Kirstein, S., and Körner, E. (2009). Large-scale real-time object identification based on analytic features. In *International Conference on Artificial Neural Networks (ICANN)*, pages 663–672.

Hasler, S., Wersing, H., and Körner, E. (2007). A comparison of features in parts-based object recognition hierarchies. In *International Conference on Artificial Neural Networks (ICANN)*, pages 210–219.

Heisele, B., Serre, T., Pontil, M., Vetter, T., and Poggio, T. (2001). Categorization by learning and combining object parts. In *Neural Information Processing Systems*, pages 1239–1245.

Higa, K., Iwamoto, K., and Nomura, T. (2013). Multiple object identification using grid voting of object center estimated from keypoint matches. In *IEEE International Conference on Image Processing (ICIP)*, pages 2973–2977.

Hoiem, D., Stein, A. N., Efros, A. A., and Hebert, M. (2007). Recovering occlusion boundaries from a single image. In *International Conference on Computer Vision (ICCV)*, pages 1–8.

Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Synthetic data and artificial neural networks for natural scene text recognition. *CoRR*, abs/1406.2227.

Johnson, J. S. and Olshausen, B. A. (2005). The recognition of partially visible natural objects in the presence and absence of their occluders. In *Vision Research 45 (2005)*, pages 3262–3276.

Leibe, B., Leonardis, A., and Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *2004 European Conference on Computer Vision (ECCV)- workshop on statistical learning in computer vision*, pages 17–32.

Leibe, B. and Schiele, B. (2006). Interleaving object categorization and segmentation. In *Cognitive Vision Systems*, pages 145–161.

Liu, Y. Z., Chen, X. L., Yao, H. X., Cui, X. Y., Liu, C., and Gao, W. (2009). Contour-motion feature (CMF): A space-time approach for robust pedestrian detection. *Pattern Recognition Letters*, 30(2):148–156.

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

Makris, A., Perrollaz, M., and Laugier, C. (2013). Probabilistic integration of intensity and depth information for part-based vehicle detection. *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 14(4):1896–1906.

Meng, M. and Potter, M. (2008). Detecting and remembering pictures with and without visual noise. *Journal of Vision*, 8(9):1–10.

Minsky, M. and Papert, S. (1969). Perceptrons. *MIT Press.*

Peng, X., Sun, B., Ali, K., and Saenko, K. (2014). Exploring invariances in deep convolutional neural networks using synthetic images. *CoRR*, abs/1412.7122.

Pepik, B., Stark, M., Gehler, P. V., and Schiele, B. (2013). Occlusion patterns for object class detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3286–3293.

Pourtaherian, A., Wijnhoven, R. G. J., and de With, P. H. N. (2013). TROD: tracking with occlusion handling and drift correction. In *IEEE International Conference on Image Processing (ICIP)*, pages 2440–2444.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386.

Rozantsev, A., Lepetit, V., and Fua, P. (2015). On Rendering Synthetic Images for Training an Object Detector. *Computer Vision and Image Understanding*, 137:24–37.

Schmaltz, C., Rosenhahn, B., Brox, T., Weickert, J., Cremers, D., Wietzke, L., and Sommer, G. (2007). Occlusion modeling by tracking multiple objects. In *DAGM Symposium*, pages 173–183.

Schulz, H. and Behnke, S. (2012). Learning object-class segmentation with convolutional neural networks. In *European Symposium on Artificial Neural Networks (ESANN)*, pages 151–156.

Serre, T., Wolf, L., Bileschi, S. M., Riesenhuber, M., and Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(3):411–426.

Stückler, J. and Behnke, S. (2013). Hierarchical object discovery and dense modelling from motion cues in RGB-D video. In *International Joint Conference on Artificial Intelligence*, pages 2502–2509.

Torralba, A., Murphy, K. P., and Freeman, W. T. (2004). Contextual models for object detection using boosted random fields. In *Neural Information Processing Systems*, pages 1401–1408.

Torrent, A., Lladó, X., Freixenet, J., and Torralba, A. (2011). Simultaneous detection and segmentation for generic objects. In *IEEE International Conference on Image Processing (ICIP)*, pages 653–656.

van Gastel, J. S., Zwemer, M. H., Wijnhoven, R. G. J., and de With, P. H. N. (2015). Occlusion-robust pedestrian tracking in crowded scenes. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 919–924.

Vedaldi, A. and Zisserman, A. (2009). Structured output regression for detection with partial truncation. In *Neural Information Processing Systems*, pages 1928–1936.

Wersing, H. and Körner, E. (2003). Learning optimized features for hierarchical models of invariant recognition. In *Neural Computation*, pages 1559–1588.

Winn, J. M. and Shotton, J. (2006). The layout consistent random field for recognizing and segmenting partially occluded objects. In *2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 37–44.

Yan, J., Lei, Z., Wen, L., and Li, S. Z. (2014). The fastest deformable part model for object detection. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2497–2504.

Yi-Hsin, L., Tz-Huan, H., Tsai, A., Wen-Kai, L., Jui-Yang, T., and Yung-Yu, C. (2010). Pedestrian detection in images by integrating heterogeneous detectors. In *IEEE Computer Symposium*, pages 252–257.

Yu, J., Farin, D., Krüger, C., and Schiele, B. (2010). Improving person detection using synthetic training data. *IEEE International Conference on Image Processing (ICIP)*, pages 3477–3480.

Zhang, X., Fu, Y., Zang, A., Sigal, L., and Agam, G. (2015). Learning classifiers from synthetic data using a multichannel autoencoder. *CoRR*, abs/1503.03163.

Zia, M. Z., Stark, M., and Schindler, K. (2013). Explicit occlusion modeling for 3D object class representations. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3326–3333.