

ROBUST DETECTION AND LOCALISATION OF CARS IN AIRBORNE IMAGERY

MOHAMED H. A. ELMIKATY

BSc(Hons), MSc, DIC

Supervisor:

Dr. T. STATHAKI

A Thesis submitted in fulfilment of requirements for
the degree of Doctor of Philosophy and Diploma
of Imperial College London

Communications and Signal Processing Group
Department of Electrical and Electronic Engineering
Imperial College London

April, 2016

I hereby declare that the intellectual content of this thesis is the product of my own research work. Some parts of this thesis have appeared before in the Early-Stage Assessment and Late-Stage Review reports and in publications. Where information and assistance received from other sources, they have been acknowledged appropriately in text and a list of references is given in the bibliography.

The present work would constitute an original attempt to develop, improve and exploit pertinent techniques of image processing, computer vision and machine learning for the detection and localisation of cars in airborne imagery. It will be suggested that some success has been achieved along the lines adopted and the findings can be considered as a guide for further studies.

Mohamed ElMikaty, London

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Abstract

With the ever-increasing demand in the analysis and understanding of aerial images, this work is focused on the use of implicit models for the detection and localisation of small targets in static airborne imagery of typical urban scenes with associated ground sampling distance of few centimetres. These scenes are characterised by the existence of complex structures and heavy clutter. More specifically, it focuses on the detection of non-occluded cars as it has many useful applications in security, surveillance, remote sensing and other domains. This is a tremendously-challenging problem because of the huge inter-class similarity among car and other non-car targets in urban scenes. Nevertheless, it will be demonstrated that high precision rates can be achieved using robust ensembles of image descriptors alongside linear classification techniques, e.g., linear Support Vector Machines. The original contributions of this work include: i) novel methods to reduce search areas by evaluating the likeliness of detection windows to contain targets, (ii) accurate methods to estimate the dominant orientation of detection windows, (iii) robust ensembles of image descriptors of low dimensionality relative to the state of the art that depict the distributions of gradients, texture, colours, spectral information and the second-order statistics of each and (iv) a new method to extract regions that are placed around roads using a Gaussian Mixture Model classifier. The performance of the proposed frameworks was evaluated against two benchmark datasets of static images, namely, the Vaihingen and the Overhead Imagery Research datasets. Results show that the proposed frameworks are superior to and outperform the state of the art.

Acknowledgment

The author would like to thank Dr. Tania Stathaki for her valuable suggestions and her guidance as well as her enthusiasm and continuous encouragement. Her skills and ability to explain complex concepts that are based on her strong technical background, knowledge and experience helped the author develop an insight for analysing, tackling and solving research problems.

The author would also like to thank Dr. David Parker of the Research and Development team at BAE Systems Military Air and Information, UK, for his useful suggestions and the fruitful discussions.

The author is grateful to all the research colleagues, administrative and academic staff within the Communications and Signal Processing Group at the Department of Electrical and Electronic Engineering, Imperial College for providing an encouraging work environment.

This work has been funded through an Industrial Cooperative Awards in Science and Technology (CASE) between the Engineering and Physical Sciences Research Council (EPSRC) and BAE Systems Military Air and Information, UK.

Contents

Abstract	4
Acknowledgment	5
List of Figures	9
List of Tables	12
Mathematical Notation	14
Abbreviations	15
Chapter 1. Introduction	16
1.1 Problem Definition	17
1.2 Motivation	18
1.3 Objectives	18
1.4 Proposed General Methodology	19
1.5 Summary of Contributions	20
1.6 Thesis Outline	23
1.7 Publications	24
Chapter 2. Literature Review	26
2.1 Detection of Target Instances	26
2.1.1 Detection of Targets using Flat Models	27
2.1.2 Detection of Targets using Deep Models	30
2.2 Detection of Car Instances in Airborne Imagery	31
2.2.1 Detection of Cars using Explicit Models	32
2.2.2 Detection of Cars using Implicit Models	35
2.2.3 Car Detection Using Road Maps	40
2.3 Summary	43

Chapter 3. Datasets and Evaluation	44
3.1 Datasets	45
3.1.1 The Vaihingen Dataset	45
3.1.2 Overhead Imagery Research Dataset (OIRDS)	48
3.2 Evaluation Methodology	49
3.3 Baseline Frameworks	52
3.4 Summary	55
Chapter 4. Detection of Cars using Fixed Cell Distributions	56
4.1 Overview of the Proposed FCD Framework	57
4.2 Window Evaluation	60
4.3 Extraction and Encoding of Features	62
4.3.1 Rotation-Invariant Histogram-of-Oriented-Gradient Descriptor	64
4.3.2 Fourier Descriptor	67
4.3.3 Truncated-Pyramid-Colour-Self-Similarity Descriptor	70
4.4 Post-Processing	71
4.5 Datasets and Classification	72
4.6 Results	73
4.6.1 Qualitative Results	73
4.6.2 Quantitative Results	76
4.7 Summary	80
Chapter 5. Detection of Cars Using Adaptive Cell Distributions	81
5.1 Overview of the Proposed ACD Framework	82
5.2 Window Evaluation	83
5.3 Estimation of the Orientations of Cars	86
5.4 Extraction and Encoding of Features	88
5.4.1 Second-Order Statistics of Histogram-of-Oriented-Gradient Descriptor	89
5.4.2 Truncated-Pyramid-Colour-Self-Similarity Descriptor and its Second-Order Statistics	90
5.4.3 Gabor Descriptor and its Second-Order Statistics	92
5.5 Post-Processing	94
5.6 Datasets and Classification	94
5.7 Results	96

5.7.1	Qualitative Results	96
5.7.2	Quantitative Results	98
5.8	Summary	105
Chapter 6. Detection of Cars in Road Areas		106
6.1	Overview of the Proposed FCD-GMM and ACD-GMM Frameworks .	107
6.2	Extraction of Road Areas	110
6.3	Datasets and Classification	111
6.4	Results	111
6.4.1	Qualitative Results	112
6.4.2	Quantitative Results	112
6.5	Summary	118
Chapter 7. Conclusions and Future work		119
7.1	Conclusions	120
7.2	Avenues of Future Research	124
Bibliography		126
Appendix A. Suppor Vector Machines		137
Appendix B. Gaussian Mixture Models		140

List of Figures

1.1	Flow diagram of the general adopted four-stage methodology.	19
2.1	Examples of polyhedra structures.	29
2.2	Examples of wireframe models.	29
2.3	Distribution of log-polar histogram bins.	29
2.4	Examples of explicit models.	34
2.5	Flow diagram of computing the boosting HOG descriptor.	37
2.6	Segmentation of a car patch into superpixels.	38
3.1	Samples of the patches that were extracted from the Vaihingen dataset and were used for training.	47
3.2	Sample regions of the areas specified by DGPF in the Vaihingen dataset for testing.	48
3.3	Sample regions of the images of the OIRDS dataset containing various cars in scenes with different backgrounds.	54
4.1	Extraction of edges in target-class and non-target-class patches.	59
4.2	Computation of the maximum gradient in target-class and non-target-class patches.	61
4.3	Hue channel histograms of target-class and non-target class patches.	62
4.4	Application of the window-evaluation stage on a sample region of the training dataset.	63
4.5	Fixed distributions of evenly-spaced cells of the same size.	63
4.6	Estimation of the dominant orientation in target-class and non-target-class patches.	69
4.7	Computation of the Fourier descriptor in target-class and non-target-class patches.	69
4.8	Sample regions of the test areas of the Vaihingen dataset after applying the pro- posed FCD framework.	74

4.9	Sample regions of the test images of the OIRDS dataset after applying the proposed FCD framework.	75
4.10	Performance of the proposed FCD framework in different test areas of the Vaihingen dataset represented by precision rates at different recall rates.	76
4.11	Performance of the proposed FCD framework on the OIRDS dataset represented by precision rates at different recall rates.	77
4.12	Performance of the proposed FCD framework and the baseline framework on the Vaihingen dataset represented by precision rates at different recall rates.	78
4.13	Summary of the FCD framework for the detection of cars in airborne imagery using fixed cell distributions.	80
5.1	Application of the window-evaluation stage on a sample region of the training dataset.	87
5.2	Estimation of the dominant orientation in target-class and non-target-class patches.	88
5.3	Distribution of cells over which descriptors are computed for different orientations.	89
5.4	Magnitude of the 64×64 Gabor kernels at different orientations.	93
5.5	Sample regions of the test areas of the Vaihingen dataset after applying the ACD framework.	97
5.6	Sample regions of the test images of the OIRDS dataset after applying the ACD framework.	99
5.7	Performance of the ACD framework in different test areas of the Vaihingen dataset represented by precision rates at different recall rates.	100
5.8	Performance of the ACD framework on the OIRDS dataset represented by precision rates at different recall rates.	100
5.9	Performance of the FCD and ACD frameworks on the Vaihingen dataset represented by precision rates at different recall rates.	102
5.10	Performance of the FCD and ACD frameworks on the OIRDS dataset represented by precision rates at different recall rates.	102
5.11	Performance of the ACD framework and the baseline framework on the Vaihingen dataset represented by precision rates at different recall rates.	103

5.12	Summary of the ACD framework for the detection of cars in airborne imagery using adaptive cell distributions.	105
6.1	Hard negatives that were claimed by the FCD and ACD car-detection frameworks as true positives.	108
6.2	Sample regions of the test areas of the Vaihingen dataset after applying the proposed frameworks.	113
6.3	Performance of the proposed integrated frameworks in different test areas of the Vaihingen dataset represented by precision rates at different recall rates.	114
6.4	Performance of the proposed FCD-GMM and ACD-GMM frameworks in the three test areas of the Vaihingen dataset represented by precision rates at different recall rates.	115
6.5	Performance comparison of the proposed frameworks with and without extracting regions of interest represented by precision rates at different recall rates.	116
6.6	Block diagram of the proposed FCD-GMM and ACD-GMM frameworks for the detection of cars in airborne imagery using automatic extraction of roads.	118

List of Tables

2.1	Summary of previous notable frameworks for the detection of cars in aerial images that were based on the use of explicit models.	35
2.2	Summary of previous notable frameworks for the detection of cars in aerial images that were based on the use of implicit models.	40
2.3	Summary of previous notable frameworks for the detection of cars in aerial images that were based on the use of road maps.	42
3.1	Performance of baseline frameworks.	52
4.1	Performance of the proposed FCD framework on the Vaihingen and OIRDS datasets represented by the Mean Average Precision MAP and recall RC rates at different false positive per image $FPPI$ values.	77
4.2	Performance of the proposed FCD framework and the baseline frameworks represented by the 11-point Average Precision AP	79
5.1	Threshold values used in the window-evaluation stage of the ACD framework.	86
5.2	Performance of the ACD framework on the Vaihingen and OIRDS datasets represented by the Mean Average Precision MAP and recall RC rates at different false positive per image $FPPI$ values.	101
5.3	Performance of the two FCD and ACD frameworks on the Vaihingen dataset represented by the Mean Average Precision MAP and recall RC rates at different false positive per image $FPPI$ values.	101
5.4	Performance of the ACD framework and baseline frameworks by the 11-point Average Precision AP	104

6.1	Performance of the proposed frameworks on the Vaihingen dataset represented by the Mean Average Precision MAP and recall RC rates at different false positive per image $FPPI$ values.	115
6.2	Performance of the proposed frameworks and the baseline framework.	117

Mathematical Notation

X, x	Scalar
$\mathbf{x}$	Column Vector
$[x_1, \dots, x_K]$	Row Vector
$(\cdot)^T$	Transpose Operator
$(\cdot)^H$	Hermitian Transpose Operator
$[\cdot]$	Truncation Operator
$\mathbf{X}$	Matrix
$y(x)$	Function
$\mathbb{E}\{\cdot\}$	Expectation Operator
$\mathbb{R}_{\mathbf{x}\mathbf{x}}$	Covariance Matrix across $\mathbf{x}$
$\#\mathbf{x}$	Cardinality of $\mathbf{x}$
d	Eigenvalue
D	Dimensionality
λ	Wavelength
θ	Orientation of the Normal to the Parallel Strips of Gabor Function
ψ	Phase Offset
σ_g	Standard Deviation of a Gaussian Envelop
γ	Spatial Aspect Ratio

Abbreviations

ACD:	Adaptive Cell Distribution
EM:	Expectation Maximisation
FCD:	Fixed Cell Distribution
GMM:	Gaussian Mixture Model
GSD:	Ground Sampling Distance
HOG:	Histogram of Oriented Gradients
IKSVM:	Intersection-Kernel Support Vector Machines
OIRDS:	Overhead Imagery Research Dataset
PCA:	Principal Component Analysis
RBF:	Radial Basis Functions
RGB:	Red, Green and Blue
ROI:	Region of Interest
SIFT:	Scale Invariant Feature Transform
SVM:	Support Vector Machines
tPCSS:	truncated Pyramid Colour Self-Similarity

Chapter 1

Introduction

OVER the past three decades, the tasks of robust and accurate detection and localisation of small targets in airborne images of urban scenes of complex heavily-cluttered backgrounds have attracted many researchers in the fields of image processing and computer vision, more specifically in the fields of image analysis and understanding.

Small targets, e.g., cars, shipping containers, aeroplanes and military installations, are usually depicted within small areas of the order of few centimetres that are represented by a small number of pixels in high-resolution aerial images of low ground sampling distance (GSD). Such high-resolution images are usually captured by sensors mounted on aeroplanes, earth-observation satellites [1] and unmanned aerial vehicles (UAVs) from a vertical or a slightly-oblique viewpoints.

Detection and localisation of targets can be defined as the ability to decide the existence or absence of a particular target of interest in the image under examination and the ability to identify its position relative to the rest of this given image, respectively. The term "detection" in this thesis hereafter involves both tasks of detection and localisation of targets.

1.1 Problem Definition

Cars, among other categories of small targets, are very commonly found in different locations, such as roads, parkings and possibly other areas. These areas constitute one of the main components of urban and suburban regions. The present work is concerned with cars as targets and the main goal is to automatically detect and locate non-occluded cars relative to the rest of a given image of a typical heavily-cluttered urban scene with an associated GSD of $8cm$. Images of concern are captured during daylight. They are static and are almost free from noise and motion blur.

This is a tremendously-challenging task because of the huge intra-class variability that cars possess and the significant inter-class similarity among car and non-car targets, such as buildings' roofs and windows, that possess very similar visual appearance to that of car targets.

The first problem of intra-class variability stems from the significant differences in the visual appearance among cars, as far as various features, such as colour intensities, size, shape, contours and boundaries, are concerned. Furthermore, it is obvious that cars' appearance explicitly depends on the viewing angle. In order to overcome this problem, many researchers have focused on the task of the detection of cars in aerial images, wherein a collection of images are taken from vertical or slightly-oblique viewpoints. These constrained viewpoints facilitate the detection task by decreasing the intra-class variability.

The second problem that usually limits the performance of a car-detection system is the existence of many non-car objects that have very similar structures of edges, boundaries and colour distributions to that of cars, such as buildings' roofs and windows in urban scenes or parts of different objects form together similar structures to those of cars.

1.2 Motivation

With the ever-increasing demand in the analysis of airborne imagery and the availability of high-resolution images that are captured over urban scenes, the perspective of this work is built around the detection of cars in airborne imagery. This particular application of the image analysis and computer vision techniques is crucial to numerous useful domains, such as security, surveillance, defence, remote sensing, urban planning and many others.

1.3 Objectives

The primary focus of this work is to develop and design a robust system that has the capacity to reliably perform the task of the detection of cars in high-resolution aerial images of typical urban scenes by using novel robust ensembles of local image descriptors. These descriptors would have the capacity to capture visual information about cars' shape, distribution of boundaries, colours, salient points and/or other cues. Such system would possess desirable properties, such as accuracy, rotation invariance, dependence on low dimensional feature vectors and consequently, low computational cost.

The objectives of the present work are three-fold: (i) to narrow search areas by the development of methods that evaluate the likeliness of detection windows to be placed around cars, (ii) to accurately estimate the orientation of cars and (iii) to develop robust ensembles of descriptors that characterise cars distinctively from other non-car objects using linear classification techniques so that high true-positive and low false-positive rates can be achieved.

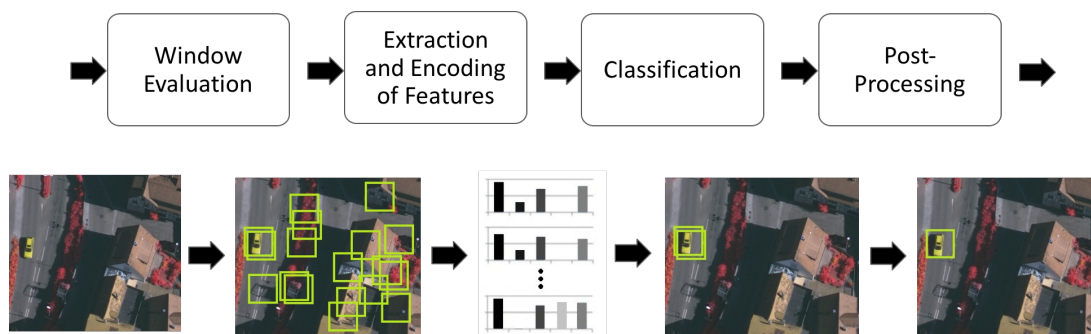


Figure 1.1: Flow diagram of the general adopted four-stage methodology. Top: Block diagram of the four main stages of the adopted general methodology. Bottom: From left to right: Original image, processed image using the window-evaluation stage, computed image descriptors for each successful candidate window, output of the classification stage and final detection after post processing.

1.4 Proposed General Methodology

The present work adopt a flat implicit-model type of approaches for the detection of cars because of its well-known efficiency to characterise distinctively fine details of targets of interest. The definition of implicit models for the detection of targets of interest will be given in Chapter 2 alongside its advantages and limitations. A general methodology that consists of four main stages has been adopted in this work and is shown in Figure 1.1. Each stage plays an important role in the pipeline as follows:

- **Window-Evaluation Stage:** the main goal of the window-evaluation stage is to narrow search areas by eliminating detection windows that are very unlikely to be placed around cars based on the computation of different image cues. The input to this stage is the original image under examination and the expected output is a number of windows of high probability to contain cars to be further examined in the pipeline.
- **Extraction-and-Encoding-of-Features Stage:** local image descriptors that depicts distinctively the visual information of cars are computed over overlapping

sliding detection windows in this stage. The input to this stage is a number of detection windows and the output is feature vectors that correspond to each detection window. These feature vectors will be passed to the classifier.

- **Classification Stage:** image descriptors that belong to detection windows that contain cars are identified in this stage using an appropriate classifier. The input to this stage is the computed feature vectors and the output is a corresponding confidence score by which the detection window is claimed to be placed around a car.
- **Post-Processing Stage:** as an overlapping-sliding-window approach has been adopted, multiple detections per single true detection are expected and the main aim of this stage is to keep only one detection. The input to this stage is a number of detection windows of high confidence score around the same target and the output is a single detection window that will be considered the final output of the pipeline.

1.5 Summary of Contributions

The primary contribution of the present work is the design and development of three car-detection frameworks that possess robust and reliable performance, when applied in heavily-cluttered environments. These three frameworks outperform the state of the art as it will be shown later. Detailed account of the main contributions is given below.

As urban scenes contain areas, wherein cars cannot be found, such as flat areas and textured surfaces corresponding to empty roads, vegetation and buildings' roofs, processing these areas are unnecessary. Therefore, eliminating detection windows that are placed over these areas early in the detection-process pipeline

will help narrow the search areas and consequently, it reduces the processing time. This is different from most of the previously adopted methodologies, wherein all detection windows undergo the same detection process. To achieve this, two novel methods to evaluate the likeliness of detection windows to contain cars are introduced. They take into account the physical and radiometric properties of cars. The first method eliminates detection windows based on the distribution of gradients and colours, whereas the second method uses the distribution of colours, texture and the second-order statistics of the distribution of local gradients.

Candidate detection windows that are thought to be placed around cars can contain cars at any orientation making it hard to exploit rotation-variant image descriptors. Although some descriptors can be modified to accommodate for rotation-invariance properties as it will be shown later for the case of Histogram-of-Oriented-Gradient descriptor, the majority of descriptors cannot be. Accordingly, detection windows must be rotated so that the dominant orientation would be always the same. In both cases, an accurate estimation of this dominant orientation is required. Two reliable methods to estimate the dominant orientation of detection windows based on the global or the local distribution of gradients are introduced. The benefit of estimating this orientation is that image descriptors will be only computed for a single orientation, unlike some works that compute descriptors at all orientations.

Another way to solve the aforementioned problem is the use of adaptive distributions of cells over which image descriptors are computed. This new introduced method of dividing the detection window into ordered cells, wherein the position of cells is no longer fixed and is a function of the dominant orientation of the detection window, helps achieve rotation-invariance properties and overcomes most of the problems that are encountered if detection windows are rotated.

In order to assign a label to a candidate detection window, the detection problem is transformed to a hyper-dimensional space by the means of image descriptors.

The main advantage of this transformation process is that it facilitates the classification process. Two robust ensembles of image descriptors that are encoded in a low-dimensional extended feature vector to depict distinctively the physical, spectral and/or radiometric properties of cars are developed. The first ensemble is composed of modified versions of Histogram-of-Oriented-Gradient, Fourier and Colour Self-Similarity image descriptors, whereas the second ensemble is composed of modified versions of Histogram-of-Oriented-Gradient, Colour Self-Similarity and Gabor image descriptors alongside the second order statistics of the distribution of each of these descriptors represented by the principal components of the associated covariance matrices.

These two novel ensembles are characterised by the ability to transform the classification problem to a hyperspace, where the car class and non-car class are almost linearly separable allowing the use of simple classifiers, such as Support Vector Machines (SVM). In addition, they have lower dimensionality than the descriptors used by the state of the art.

In an attempt to further narrow search areas to roads as they are the areas with the highest probabilities to contain cars in urban scenes, a Gaussian Mixture Model (GMM) classifier is used in a novel fashion to extract such areas based on the radiometric properties of areas that are paved with asphalt.

The main points of strength of the proposed methods include: (i) efficient encoding of several image cues with low dimensionality, (ii) high detection precision and low false-positive rates, (iii) short processing time and (iv) the ability to generalise and achieve acceptable performance, when used to detect cars in unseen data that are captured under different acquisition conditions to those of the training data and with lower resolution.

1.6 Thesis Outline

This chapter introduced the definition of the general problem that this thesis is concerned with. Furthermore, it described main goals, outlined key applications and summarised the contributions of the work presented here. The rest of the thesis is organised as follows:

- Chapter 2 begins with a brief discussion of the previous and the state-of-the-art research ideas and trends on the detection of general target instances. Then, it presents a more detailed discussion of the research ideas that were proposed to solve the problem of the detection and localisation of cars in airborne imagery. It also states the advantages and disadvantages of the main research methodologies that were adopted by researchers, who addressed the same problem.
- Chapter 3 describes two benchmark datasets that were used in training and testing the performance of the proposed frameworks followed by the explanation of the performance metrics that were used for assessment and evaluation. The chapter ends with outlining baseline frameworks that used same datasets and their obtained results.
- Chapter 4 introduces a framework that uses standard fixed cell distributions (FCD). It is based on the computation of modified versions of Histogram-of-Oriented-Gradient, Fourier and Colour-Self-Similarity descriptors. It begins with an overview of the method followed by giving the implementation details of each stage in the pipeline. Then, it presents and analyses the obtained results. After that, a comparison between the proposed framework and the state of the art is carried out. It ends with presenting and discussing the results of the conducted running-time analysis that was carried out.

- Chapter 5 introduces a framework that uses adaptive cell distributions (ACD). It is based on the computation of modified versions of Histogram-of-Oriented-Gradient, Colour-Self-Similarity and Gabor descriptors and the eigenvalues of the associated covariance matrices. It begins with an overview of the method followed by giving the implementation details of each stage in the pipeline. Then, it presents and analyses the obtained results. After that, a comparison between the proposed FCD and ACD frameworks is carried out followed by a comparison with the state of the art. It ends with presenting and discussing the results of the conducted running-time analysis that was carried out.
- Chapter 6 discusses the possible use of road maps in reducing the false positive rate. It introduces a simple, yet an effective method to extract automatically regions that are placed around roads using a Gaussian Mixture Model as road maps of the used datasets are not available. The performance of the FCD and ACD frameworks for the detection of cars alongside the proposed method of the extraction of roads is then evaluated and results are given and discussed.
- Chapter 7 gives a brief summary followed by the discussion of conclusions drawn from the conducted experiments and the attained results. Moreover, it discusses the advantages and limitations of the developed frameworks. It ends with outlining possible directions for further future work.

1.7 Publications

The results of this research work have already been published or are currently under review in top-tier conferences and journals in the fields of Image Processing, Computer Vision and Remote Sensing:

-
- The work of Chapter 4 was published in the IAPR International Conference on Pattern Recognition (ICPR'14) [2].
 - The work of Chapter 5 has been submitted to the IEEE Transactions on Geoscience and Remote Sensing [3] and is currently under review.
 - The work of Chapter 6 has been submitted to the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing [4] and is currently under review.

Chapter 2

Literature Review

ROBUST and accurate detection of cars in aerial images have drawn the attention of researchers in the image-processing and computer-vision communities for many years, because it can be exploited in a wide range of applications. However, it is a very challenging task as previously discussed in Chapter 1. Moreover, the difficulty of acquiring such images with high resolution formed another obstacle for researchers in the past.

This chapter covers a review of the open literature on related works and it is divided into two main parts: (i) a review on the detection of general target instances is presented in Section 2.1 and (ii) a review on the detection of cars as a particular target of interest in airborne imagery is presented in Section 2.2. A summary of the main research trends that were proposed to tackle the problem of car detection is given in Section 2.3.

2.1 Detection of Target Instances

Detection of targets has been addressed extensively in the literature [5], [6], [7]. It is a tremendously-challenging task as the methods that have been previously devel-

oped to carry out this task encountered major problems. For instance, one of the main problems is that in many cases, targets of interest do not exhibit a distinctive visual appearance. In addition, it was hard to encode fine visual details that are very useful in these cases due to the limited computational power. Moreover, in real-life scenarios, targets are usually surrounded with heavily-cluttered complex backgrounds making it hard to separate the foreground from the background [8]. Another major problem was the unavailability of datasets that were captured under different image acquisition conditions, e.g, different viewpoints, illumination, occlusion and deformation conditions. These problems limited the generalisation ability of the proposed methods, hindered the system design and affected dramatically their overall performance.

Past and recent techniques for the detection of targets can be generally categorised into flat and deep models [9] according to the type of the used image-feature-extraction and image-descriptor-encoding techniques. An image feature is defined as a distinctive characteristic of a given image, whereas an image descriptor is a mathematical function that is used to encode single or multiple extracted image features in a vector format.

2.1.1 Detection of Targets using Flat Models

The majority of the past proposed techniques that are found in the literature were based on the use of hand-crafted local and global image descriptors [10], [11], [12] that are used to depict the most distinctive features of the target of interest encoding various image cues. For example, the spatial distribution of gradients, edges, colour intensities, texture and other cues or combinations of these were used in conjunction with some measure for the cost of matching or with an appropriate classifier. They did not use deep neural network and hence they are named ‘flat’. Flat models can be further grouped into explicit models and implicit models.

Detection of Targets using Explicit Models

Detection of targets using explicit models usually follows a “bottom-up” approach, wherein hypotheses about image features are built and then they are tested using matching based on the distance between descriptor vectors, such as the Euclidean, Chi-square and Mahalanobis distances [13]. In other words, image-descriptor templates of two or three dimensions that resemble the visual appearance of the target of interest are designed and saved prior to the detection process. Hence, such models are named explicit. Targets are assumed to be found in regions of the image, where the cost of matching between the patch under examination and that pre-designed template is lower than a pre-defined threshold, which is usually found empirically.

Early works of explicit models used different polyhedra structures on constant backgrounds [14], such as those shown in Figure 2.1. Later, wireframe models were used as templates and the correspondence between these models and the target’s contours and edges was measured. Examples of wireframe models, such as the models shown in Figure 2.2, are the work of Lowe [15] and the work of Huttenlocher and Ullman [16].

More recently, the detection of targets was performed using more advanced templates that characterise local distribution of image cues. For instance, shape contexts [17], [18], shown in Figure 2.3, shape signatures [19] and speeded-up robust features [20]. They use the local distribution of image gradients, their associated strength and different matching cost functions.

Among the major advantages of explicit models is that they do not require a training dataset, which is hard to collect and annotate for some targets. Nonetheless, the matching process is usually computationally expensive and the associated performance degrades in scenes with complex cluttered backgrounds as templates are usually matched to many non-target objects.

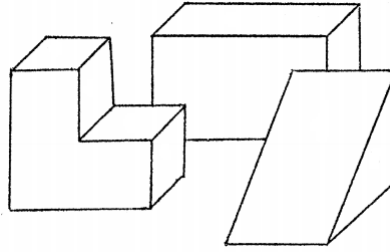


Figure 2.1: Examples of the polyhedra structures that were used in [14].

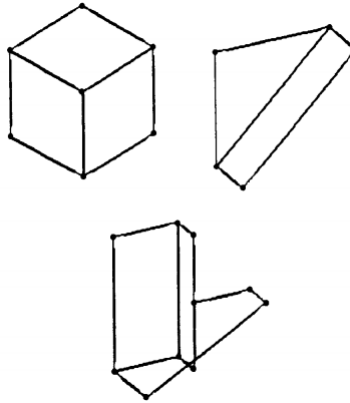


Figure 2.2: Examples of the wireframe models that were matched to the images in [16].

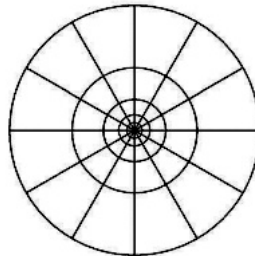


Figure 2.3: Distribution of the log-polar histogram bins that were used to describe the shape context by encoding the distribution of gradients in [17].

Detection of Targets using Implicit Models

Unlike explicit models, implicit models do not use pre-saved image-descriptor templates. Instead, they use a single or multiple image descriptors that are engineered to capture and encode different cues of targets and they usually follow a “top-down” approach, wherein hypotheses are built using a classifier that is trained on a set of

particular features for the target of interest [13]. In other words, the detection problem is transformed from the spatial domain to a hyper-dimensional space, wherein a classifier is trained to divide this space into two or more subspaces: one corresponds to target-class objects and the other corresponds to non-target-class objects. The trained classifier is later used to decide whether the patch under examination belongs to the target-class subspace or the non-target-class subspace.

Examples of implicit models are the work of Lowe [21] and the work of Dalal and Triggs [22], where they used image descriptors that encode the local distribution of gradients extracted by the Difference of Gaussian (DoG) filter and the first-derivative filter, respectively. More complex representation and encoding schemes of image features were also used, for example the bag-of-visual-word model [23] and its succeeding variants.

The performance of implicit models is acceptable in many real-life scenarios [24]. Nonetheless, the performance is significantly affected by more than a factor: the choice of the image cues and image descriptors and no solid theoretical foundations about this choice are found in the open literature [10], the dimensionality of the descriptors used and how rich the training dataset is.

2.1.2 Detection of Targets using Deep Models

The second category of techniques is deep models, wherein deep neural networks [25], [26], [27] of multiple serial and parallel layers of neurones, i.e, non-linear processing units, are used to extract image features, which are then passed to a classifier or the deep network carries out both the task of extraction of features and classification. Deep networks can be categorised into cascaded-architecture and hierarchical-architecture networks [28]. The cascaded networks consist of successive layers of neurones and each layer uses the output from the layer preceding it, whereas the

hierarchical networks depend on learning high-level features from low-level features.

Although deep networks have outperformed the state-of-the-art in many computer vision tasks, for example in image classification, they require a very rich training dataset with a very huge number of training samples to perform well. Moreover, there is a large number of parameters to tune, which are usually set empirically [29]. In addition, they can be easily fooled and produce high-confidence scores for images that are completely unrecognisable for humans [30]. This makes them unreliable for many domains that require high reliability in the output, e.g., military and security domains.

2.2 Detection of Car Instances in Airborne Imagery

First methodologies that were proposed to tackle the problem of detection of cars from perpendicular or slightly-oblique viewpoints in aerial images used raw pixel values and/or very simple features, such as straight lines owing to the unavailability of high-resolution aerial images, where fine details of cars are obvious. For instance, in the work by Ruskoné *et al.* [31], the authors used a dataset of a ground sampling distance (GSD) of 30 – 40cm. This means that the length of the car was expected to be represented by approximately ten pixels making it very difficult to extract structural features, such as the relative distribution of edges, boundaries and contours.

With the availability of higher resolution aerial images of GSD of 15cm, more recent methodologies that have been proposed to solve the problem of the detection of cars in airborne imagery used more complex explicit and/or implicit models to depict and encode structural and radiometric features of cars. A recent survey can

be found in the open literature [32], wherein the authors carried out a comparison among four recent works [33], [34], [35] and [36] on the problem of car detection in static and dynamic aerial images with different adopted methodologies. These four works belong to the group of target detection techniques based on flat models that was described earlier in Section 2.1.1 and more specifically, detection of targets using implicit models, wherein image descriptors are used alongside a suitable classifier.

Deep models for the detection of cars in airborne imagery can hardly be found in the literature as they require very huge training datasets and high processing power. To best of the author's knowledge, only one work was proposed by Chen *et al.* [37] to solve the problem of car detection using deep models. In this work, the authors used a hierarchal deep model to extract features exploiting convolutional layers and max-pooling layers in a hybrid fashion. Although their method achieved high accuracy, their testing dataset was rather simple as it has vast areas containing large number of cars, e.g., large parking areas in urban cities.

In the following sections, a review on the past and recent methodologies about the detection of car instances in airborne imagery is presented and representative works are given accompanied by brief demonstrations of their basic ideas. These past proposed methodologies can be broadly categorised into three main groups: the first group depends on the use of explicit models, the second group exploits implicit models and the third group uses road maps alongside either of these two models.

2.2.1 Detection of Cars using Explicit Models

Following the idea of explicit models in the detection of general target instances, researchers tried to solve the problem of the detection of cars using these models. Most of these models used a fixed or an adaptive wireframe model and an intensity model that describes the casted shadows by the bodies of cars.

For example, Zhao and Nevatia [38] assumed that cars were aligned with road directions and they used a coarse generic car model that consisted of a wireframe model, shown in Figure 2.4, and a surface-reflectance model to search for locations in the image of high probability to contain cars. They designed their generic model template incorporating the boundaries of the cars' body, the front windscreen and the surrounding shadow area in addition to the intensity of the shadow area. After projecting this generic model to the ground plan, features were fed to a Bayesian network [39] in a "bottom-up" matching approach followed by a Bayesian minimum risk classifier to decide the existence of a car target in a given image.

In a similar fashion, Hinz and Schlosser [40], [41] exploited an adaptive tree-like 3-D model that resembled the geometric structure of cars' contours and the colour constancy between the cars' bonnet and roof taking into account the orientation of the car and the direction of sunlight. In their model, such information was given as prior knowledge to compute the expected shadow projection. Figure 2.4 shows an example of their model. Matching between the 3-D model and the image was carried out in a "top-down" approach as the authors claimed that grouping features and matching the whole model in a "top-down" approach would yield to the achievement of better performance than adopting a "bottom-up" approach.

On the contrary, Choi and Yang [42] adopted a "bottom-up" approach to solve the problem of the detection of cars. Alike Zhao and Nevatia [38], they assumed that cars are aligned with the directions of roads. Moreover, they assumed that the surface of roads is almost homogenous and constitutes the biggest surface in a given urban image and that cars have symmetric structures. In their method, blobs of pixels that possessed similar geometric and radiometric properties to that of cars were formed using a mean-shift clustering algorithm [43]. A log-polar shape descriptor was then used to filter successful candidates of the preceding stage based on the measure of the similarity among edge and contour distributions using the

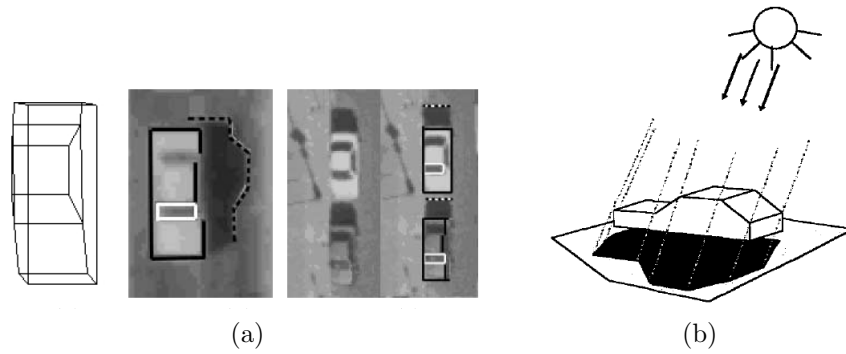


Figure 2.4: Examples of the explicit models used in previous frameworks: (a) The wireframe model used in [38]; (b) the 3-D model used in [40] and [41] taking into account the direction of sunlight.

idea of shape context discussed in Section 2.1.1.

Table 2.1 summarises features that were used by the aforementioned frameworks, their requirements and the ground sampling distance of the dataset used. Problems of these approaches arose from the designed explicit model as researchers, in most cases, had to design generic models in order to cope with the high intra-class variability. Such generic models are usually simple and do not incorporate detailed description of the image features that distinguish cars and as a result they usually fit to many positions in the image. Beside the common known problems associated with the use of explicit models, such as the high computational cost, which the matching process incur, the huge amount of human intervention in modelling such generic or fixed templates and the performance usually degrades, when applied in urban scenes with heavily cluttered backgrounds [44]. Constraints about the orientation of cars and about the direction of sunlight and the angles of shadows in some cases require the availability of prior knowledge. Definition of the performance metrics used in Table 2.1 will be given later in Chapter 3.

Table 2.1: Summary of previous notable frameworks for the detection of cars in aerial images that were based on the use of explicit models.

Framework	Features	GSD	Requirements	Performance
Zhao and Nevatia [38]	Distribution of boundaries and distribution of colours	$\approx 17.3cm$	Cars' alignment with roads	$PR \approx 62\%$, $RC \approx 95\%$
Hinz [40] and Schlosser <i>et al.</i> [41]	Distribution of contours and colour constancy	$\approx 15cm$	Prior knowledge of orientation and sunlight direction	-
Choi and Yang [42]	Symmetry of blobs, distribution of edges and symmetry of phase	$\approx 25cm$	Cars' alignment with roads and big homogenous road surface	$PR \approx 91.8\%$, $RC \approx 86.5\%$

2.2.2 Detection of Cars using Implicit Models

More recently, many car detection techniques have used either a single image descriptor or an ensemble of image descriptors alongside some classifier aiming to have a better description and encoding of the visual properties of cars, to overcome the problem of simple description of the features of cars and to be free from the constraints imposed by the use of explicit models.

A possible example of this class of methodologies of using implicit models is the work that was conducted by Nguyen *et al.* [44], who used an ensemble of image descriptors, namely, Haar Wavelets [45], Histogram of Oriented Gradients (HOG) [22] and Local Binary Patterns (LBP) [46]. They used an adaptive boosting (AdaBoost) [47] classifier with an online-training approach, wherein the training data is available in sequential order and is used to update the classifier at each step [48], in order to save the effort for labelling training data. For post-processing, they exploited a non-parametric clustering algorithm [49] that involved a mean-shift clustering [43] to group multiple detections around the same position.

Some past works used a single image descriptor to carry out the task of the

detection of cars. For instance, in [35], Sahli *et al.* used only the scale-invariant feature transform (SIFT) [21] descriptor and a non-linear support-vector-machine (SVM) classifier with radial-basis-function (RBF) kernel [39] as the authors reported that the problem was not separable in the linear hyperspace. In order to discard multiple detections, the authors used an affinity-propagation algorithm [50] in their post-processing stage. Their method is efficient for the detection of cars in scenes with simple backgrounds, e.g., rural areas.

Following the same idea of using only one single descriptor, Cao *et al.* [36] used a modified version of the HOG descriptors. They first computed HOG on local cells and then trained an AdaBoost classifier [51] transforming the weak classification abilities of the gradient histogram bin into a strong classifier. A flow diagram of how they computed the boosting HOG descriptor is shown in Figure 2.5. The output of the AdaBoost classifier was used to construct a feature vector with a lower dimensionality than that of the original HOG implementation by [22]. Feature vectors were then passed to a linear SVM classifier [39] to give a decision on whether it would be a true car target. Their proposed framework is only able to detect cars oriented horizontally as the HOG descriptor is a rotation-variant descriptor.

Kembhavi *et al.* [52] adopted an implicit multiple-descriptor approach. They exploited Colour Probability Maps (CPM) to characterise the distribution of colours of cars and parts of the surrounding background, Pairs of Pixels (PoP) to describe cars' structures and the standard HOG descriptor to depict the colour and geometric structural properties. Then, they used a Partial Least Squares (PLS) algorithm [39] to reduce the dimensionality of their resulting hybrid 70000-D descriptor, which is characterised as a very high-dimensional feature vector. After that, a quadratic discriminant classifier [39] was used to carry out the classification between car and non-car targets.

Also, Sahli *et al.* [53] developed a method that classifies superpixels, i.e.,

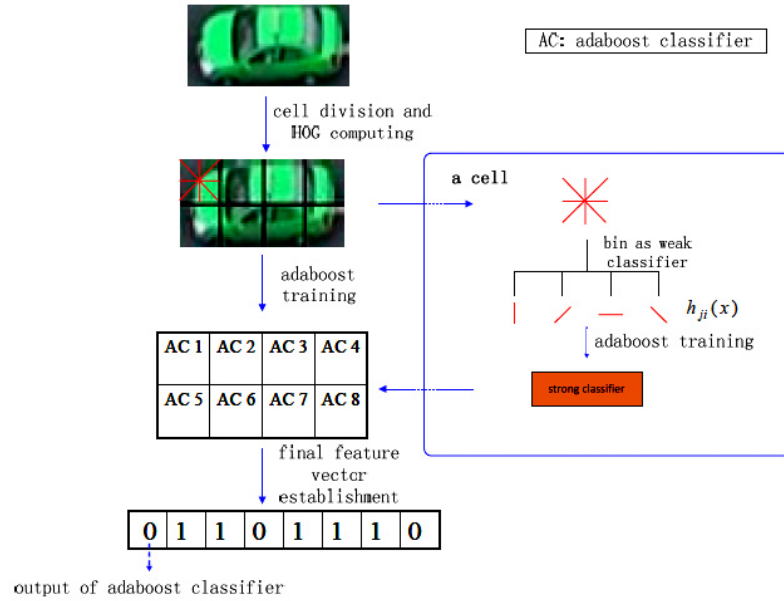


Figure 2.5: Flow diagram of computing the boosting HOG descriptor by [36].

clusters of pixels that are consistent in their colour distribution. First, they used a non-parametric method [54] to extract salient regions of dense edge distributions assuming that these are the regions, where cars were most likely to be found. Second, they segmented salient regions to superpixels using the fast simple linear iterative clustering (SLIC) algorithm [55] as shown in Figure 2.6. Third, they characterised each superpixel using concatenated feature vector that consists of SIFT and LBP descriptors. An RBF-SVM classifier was then used to classify superpixels that belong to car and non-car targets. The performance of their proposed framework is limited by the dependence on extracting salient regions of dense edge distributions as it would not work efficiently in cluttered environments.

Shao *et al.*, in their work [56], used a concatenated multiple-descriptor feature vector that incorporated HOG to depict gradients, LBP to encode texture and opponent histograms to capture the distribution of colours. They formed a single descriptor of 6760 dimensions and used an IKSVM classifier [57] to separate the car subspace from the non-car subspace. Among the limitations of their proposed

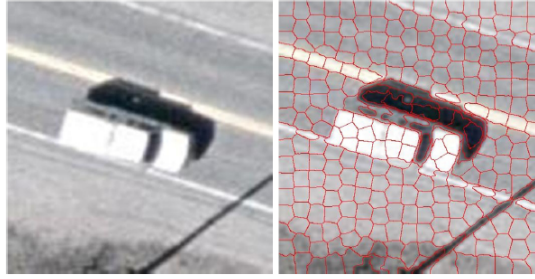


Figure 2.6: Segmentation of a car patch into superpixels by [53]. Shown images were scaled for viewing.

framework, are the use of a non-linear classifier, which is computationally expensive, and the produced high rate of high-confidence false positives.

Razakarivony and Jurie [58] divided the problem of the detection of cars into two problems. They trained two different classifiers, one for the foreground and the other for the background. They tested their method using raw pixel values, gradients or HOG features and different classifiers (autoencoders [25], principal-component-analysis (PCA) based manifold [59] and SVM [39]) for each of the background and foreground data. Although the authors achieved acceptable detection rates, they used two separate models, which means that classification is performed twice on new data samples.

Recently, Madhogaria *et al.* [60] have used HOG to detect cars in a two-stage process. In their first stage, HOG descriptors were computed over rotated images and a linear SVM classifier was used to decide whether the detection window under examination might have high probability to be placed around a car. In the second stage, a causal Markov Random Field classifier [39] was used to refine the output of the linear SVM classifier by discarding most of the false positives. According to the authors, their method achieved lower false-positive rate than in the case of using discriminative models. Nonetheless, their method failed to achieve similar recall rates to those of discriminative models. In addition, their method output

false-positives with very high confidence scores compared to the confidence scores of the true positives.

Table 2.2 gives a brief summary on the image descriptors used by previous researchers who adopted different implicit-model approaches to detect cars in airborne imagery. It can be observed from this table that with the increase in the resolution of the available datasets, which corresponds to the decrease in the ground sampling distance, researchers have addressed the problem using more fine accurate models that have the ability to capture fine structural details and that were specifically tailored to cars. Also, the majority of the proposed frameworks did not imply any constraints about the orientation of cars or the direction sunlight, i.e., did not require prior knowledge, unlike past works that used explicit models. Definition of the performance metrics used in Table 2.2 will be discussed later in detail in Chapter 3.

Furthermore, most of the works have used some sort of gradient descriptors, such as SIFT and HOG, and incorporated them with descriptors for other image cues. Few works used one single image descriptor. For instance, the work of Sahli *et al.* [35] used SIFT descriptor and the work of Cao *et al.* [36] used boosted HOG descriptor. Nonetheless, they used alongside this single descriptor a non-linear or a complex classifier as the problem is not separable in the case of using one image descriptor in the linear hyperspace. On the other side, works that exploited more than one image descriptor faced the problem of the curse of dimensionality [39], wherein the data become sparse in the hyperspace. Accordingly, they were forced to use a dimensionality-reduction algorithm, for instance the work of [52].

Table 2.2: Summary of previous notable frameworks for the detection of cars in aerial images that were based on the use of implicit models.

Framework	Features	GSD	Requirements	Performance
Nguyen <i>et al.</i> [44]	Haar wavelets, HOG and LBP	$\approx 8cm$	-	$PR \approx 76\%$, $RC \approx 88.5\%$
Sahli <i>et al.</i> [35]	SIFT	$\approx 11.2cm$	Simple backgrounds, such as rural areas.	$RC \approx 95.2\%$
Cao <i>et al.</i> [36]	Boosted HOG	$\approx 14cm$	Cars' alignment along same direction	$PR \approx 90\%$, $RC \approx 90\%$
Kembhavi <i>et al.</i> [52]	CPM, HOG and PoP	$\approx 15cm$	-	$PR \approx 22\%$, $RC \approx 89\%$
Sahli <i>et al.</i> [53]	SIFT and LBP for superpixels	$\approx 11.2cm$	Simple backgrounds, such as rural areas.	$RC \approx 90.91\%$
Shao <i>et al.</i> [56]	HOG, LBP and opponent histograms	$\approx 8cm$	-	$PR \approx 78\%$, $RC \approx 50\%$
Razakarivony and Jurie [58]	Raw pixels, gradients or HOG	$\approx 15cm$	-	$AP \approx 48.9\%$
Madhogaria <i>et al.</i> [60]	HOG, raw pixels and intensity differences among neighbouring pixels	$\approx 15cm$	-	$PR \approx 49\%$, $RC \approx 95\%$

2.2.3 Car Detection Using Road Maps

In order to restrict search areas to regions that are more likely to contain cars, several previous works have exploited road maps to extract roads and parking areas from the image under examination as it can provide a double-benefit solution. First, in heavily-cluttered environments, such as urban areas, using road maps decreases the rate of false positives. Second, as the scanning process does not cover the whole image, it ensures a shorter processing time allowing the use of more complex hybrid

features that can depict the characteristics of cars more distinctively.

For instance, the work made by Moon *et al.* [61], [62], wherein they made use of the locations of roads and buildings and assumed that cars possess 2-D rectangular shapes and they generalised their model to parallelograms using orthogonal projection to deal with the camera capturing directions. Accordingly, they utilised the derivative of the double-exponential (DODE) filter to extract gradients, the Canny edge detector [63] and the generalised Hough Transform (GHT) [64] to extract rectangle-like shapes.

Similarly, Zheng *et al.* [65] extracted roads and parking areas and used morphological shared-weighted neural network (MSNN) that contained morphological structuring elements [66] that extracted features and a classifier stage to distinguish actual car pixels using a feed-forward neural network [67].

Eikvil *et al.* [68] proposed a car-detection approach that consisted of a segmentation stage [69] that filtered trees and over-shadowed regions in the image and a two-stage object classification framework exploiting multi-spectral images, geometric properties, such as the geometric area, the compactness, the angle deviation and the spatial spread, colour properties, such as region mean and standard deviation, distribution of gradients as well as road networks.

Following the same idea, Tuermer *et al.* [70], proposed filtering areas with very low probability to contain cars using road databases, disparity maps and a pre-processing stage that discarded large homogeneous areas before the application of an object detector. They computed HOG features at different scales resulting in a 6148-dimension descriptor.

Table 2.3 summaries most of the past works that used road maps alongside either an explicit or implicit model. Whilst using road maps has shown in these works that it boosts the performance in terms of high true-positive and low false-

Table 2.3: Summary of previous notable frameworks for the detection of cars in aerial images that were based on the use of road maps.

Framework	Features	GSD	Requirements	Performance
Moon <i>et al.</i> [61] and [62]	Edges, gradients, and GHT	$\approx 26.4cm$	Works in car parking areas only	$PR \approx 41\%$, $RC \approx 61\%$
Zheng <i>et al.</i> [65]	Morphological structuring elements and neural networks	$\approx 100cm$	Exact road maps	$PR \approx 94.4\%$, $RC \approx 75.8\%$
Eikvil <i>et al.</i> [68]	Spatial, geometric and radiometric properties	$\approx 60cm$	Multi-spectral images and road maps	$PR \approx 82.8\%$, $RC \approx 84.5\%$
Tuermer <i>et al.</i> [70]	HOG	$\approx 13cm$	Disparity maps	$PR \approx 53\%$, $RC \approx 94\%$

positive rates, it requires accurate road maps to be known *a priori* and a precise map-projection method, which is computationally expensive, and the use of extra hardware, such as a global positioning system (GPS). Moreover, such methods that depend on the use of road maps are not able to detect cars that are placed in areas other than roads. For instance, cars in urban scenes are sometimes found parked between houses or between gardens' fences. As mentioned before, performance metrics used in Table 2.3 will be discussed later.

It is worth to mention that the presented results in Table 2.1, Table 2.2 and Table 2.3 can just give a general idea about the performance of previously-developed methods and they cannot be used to make a fair comparison among these methods as each method was tested on a different dataset.

2.3 Summary

This chapter discussed briefly main research trends to solve the problem of the detection of general target instances. It also presented a summary of the previous notable frameworks that were proposed to solve the problem of the detection of cars in aerial images that adopted different models, namely, explicit models, implicit models and models that use road maps alongside one of these models.

Among recent efficient methodologies in detection of cars is the detection of cars using implicit models because of its superiority in capturing and encoding fine structural details cars. It requires the availability of high-resolution aerial images. However, such images are becoming more available with the existence of earth-observation satellites and low cost UAVs.

Chapter 3

Datasets and Evaluation

WHILST there are various proposed methodologies to detect cars in airborne imagery, there is no common framework to compare their performance on some benchmark data. The performance of most of the previous works was evaluated on different datasets that contain images of different scenes and no information about the acquisition conditions and the used sensor was available. Furthermore, those datasets are not publicly available to other researchers to test the performance of their new ideas.

Therefore, in order to validate the effectiveness of a new proposed method and to allow its reproducibility, its performance must be evaluated on benchmark datasets. In this thesis, two benchmark datasets, namely, the Vaihingen dataset [71] and the Overhead Imagery Research Dataset (OIRDS) [72], were exploited to assess and evaluate the performance of the proposed methods. Both datasets were captured so that cars would have perpendicular or slightly-oblique points of view. In the present work, the Vaihingen dataset was used for the purposes of training and testing, whereas the OIRDS dataset was used as an unseen dataset to test the ability of the proposed methods to generalise on images captured under different acquisition conditions.

This chapter is divided into three parts: (i) a description of the two benchmark datasets is given in Section 3.1, (ii) performance metrics that were used to evaluate the proposed methods are discussed in Section 3.2 and (iii) baseline frameworks that have used these two datasets and their obtained results are given in Section 3.3.

3.1 Datasets

As aforementioned, the Vaihingen and OIRDS datasets were chosen for training and testing purposes. Acquisition information of these two benchmarks is given in the following sections and how the training dataset was generated as well as a description of the testing datasets.

3.1.1 The Vaihingen Dataset

The Vaihingen dataset comprises a set of high-resolution aerial images captured over the city of Vaihingen - Germany on the 24th of July and on the 6th of August 2008 by the German Society for Photogrammetry, Remote Sensing and Geo-information (DGPF) [71]. Images were captured using an Intergraph Z/I digital mapping camera (DMC) [73].

The associated ground sampling distance (GSD) is equal to $8cm$ and the dataset has a radiometric resolution of $11bits$. Images of the Vaihingen dataset are pan-sharpened coloured infra-red images with spectral bands IR, R and G and they were provided in 16-bit TIFF RGB files. This radiometric resolution was converted to be of $8bits$. The focal length of the used camera is $120mm$. Images were acquired at a flying height above the ground of $900m$. This means that the viewpoint is at a perpendicular or a slightly-oblique angle.

Training Dataset

Cars in the Vaihingen dataset are almost of the same size and they can be contained inside square windows of about 56×56 pixel size as the GSD is 8cm . Square patches were chosen instead of rectangular patches as it was found experimentally that square patches result in better performance. They incorporate some of the surrounding contextual information around cars. Furthermore, they facilitate the computation of image descriptors as cars can have any orientation in the image.

The DGPF did not provide a specific dataset for training. Instead, they marked testing areas and left other areas for training. Therefore, square patches of fixed size of 64×64 pixels that contain cars with different colours and at different orientations were chosen from areas other than test areas. They were used as the positive training data. It was found during the training and validation processes that adding these extra eight pixels around the car in both the horizontal and vertical directions is essential for the efficient computation of image descriptors over cars and a small part of the surrounding background. In order to increase the diversity among the training samples, data augmentation techniques were exploited by the inclusion of mirrored and horizontally-flipped versions of square patches that were chosen from various regions.

In a similar way, negative training data was created by the random selection of 64×64 size patches from different backgrounds as well as bootstrapped hard negatives that were detected as false positives during the validation process. The validation process was not carried out on the testing areas. The number of positive and negative training examples differs from one proposed method to another. The actual number used in each method will be mentioned in the description of its implementation in the following chapters and it differs from one method to the other. Sample patches of the training dataset are shown in Figure 3.1.

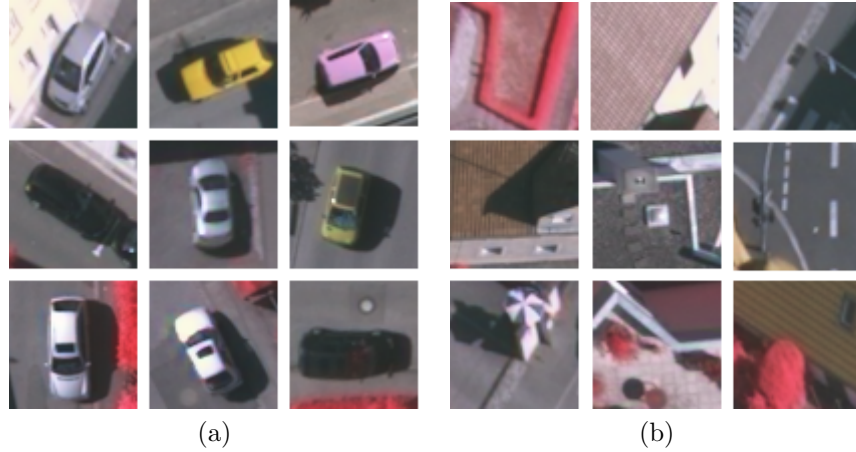


Figure 3.1: Samples of the patches that were extracted from the Vaihingen dataset and were used for training: (a) positive training samples of car-class targets of different orientations and colours; (b) negative training samples of non-car-class targets. Shown images were scaled for viewing.

Testing Dataset

DGPF specified three different testing areas for the assessment and evaluation of new proposed methods for the detection, segmentation and recognition of a particular target of interest. Test areas contain typical urban backgrounds with different building styles, roads and vegetation as shown in Figure 3.2. Description of the three areas is as follows:

- Area 1 was captured over the city centre and it contains historic buildings having complex shapes and some trees.
- Area 2 has a few modern high-rising residential buildings that have flat roofs and are surrounded by trees.
- Area 3 was captured over a suburban residential area with typical detached houses.

Ground truth data for these areas was not provided. Consequently, it was created manually by marking the co-ordinates of bounding boxes placed around 128

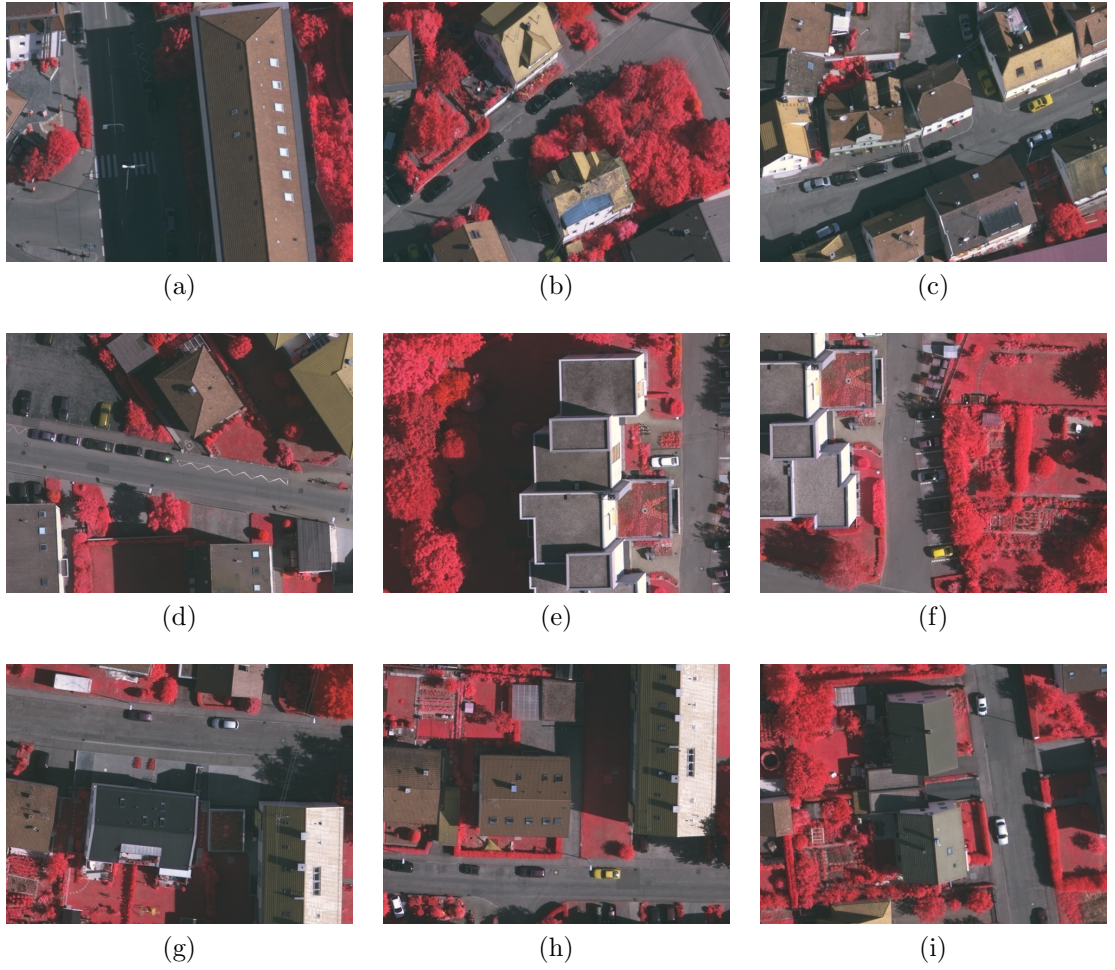


Figure 3.2: Sample regions of the areas specified by DGPF in the Vaihingen dataset for testing [71]: (a),(b) and (c) sample regions of test area 1; (d),(e) and (f) sample regions of test area 2; (g),(h) and (i) sample subregions of test area 3. Shown images were scaled for viewing.

cars, mostly sedan and hatchback cars. Ambiguous cars, for example cars that were partially occluded were not taken into account, when measuring the performance of the proposed methods in this work following previous works [52], [58] and [60].

3.1.2 Overhead Imagery Research Dataset (OIRDS)

The OIRDS dataset was taken from non-copyrighted sources and it is a part of the freely-available aerial imagery for the public. It was captured by the United States Geological Survey (USGS). Over 95% of the provided images has a GSD of

$\approx 15\text{cm}$ under a variety of operating characteristics. They were provided as a set of images of size 256×256 pixels [72]. Images were up-sampled by a factor of 1.875 so that the final GSD is $\approx 8\text{cm}$. This means that the dataset has a lower resolution compared to the training dataset described in Section 3.1.1. The dataset has more than 900 annotated images that contain approximately 1800 cars, mostly sedan and hatchback cars. Examples of the test images are given in Figure 3.3.

The OIRDS dataset was used for testing only, i.e., no examples were included in the training dataset. Ground truth data for these areas was provided in the format of the average target centroids. Nonetheless, partially-occluded cars were not taken into account when measuring the performance of the proposed methods as previously described.

As a fixed-size window has been adopted, datasets with aerial images of lower GSD can be down-sampled so that the final GSD is around 8cm . Performance may degrade for datasets with higher GSD. No further pre-processing was performed on both dataset, i.e, no colour enhancement or shadow removal.

3.2 Evaluation Methodology

Evaluation methods for target-detection algorithms depend on the adopted approach to carry out the task of detection [74]. As a window-based approach was adopted in the proposed methods alongside a classifier that outputs a confidence score, two widely-used quantitative metrics that are well known for their effectiveness were chosen to evaluate the performance and to compare it with the performance of the state-of-the-art methods that constitute the baseline performance. The two quantitative metrics are the Precision-Recall curves and the Average Precision. In addition, the Mean Average Precision and Recall rates at different False-Positive-Per-Image rates will be given. These quantitative metrics work on ranked outputs

according to the confidence score in a descending order. They are representations to the true-positive, the true-negative, the false-positive and the false-negative rates.

In the adopted evaluation methodology, a bounding box B_D , i.e., a detection window, was considered to be a true positive if it matched a corresponding ground truth bounding box B_{GT} using Equation 3.1 and there is no other bounding boxes matching the same ground truth [24]. In the case of having more than one bounding box matching the same ground truth, only one of them was considered a true positive and others were considered false positives.

$$\frac{Area(B_D \cap B_{GT})}{Area(B_D \cup B_{GT})} \geq 0.5 \quad (3.1)$$

Having defined the threshold of matching a detection window to one element of the ground truth $\mathbf{b}_{GT}$, detection windows can be classified to one of the following four cases:

- *True Positive (TP)*: a detection window was considered to be a true-positive element ($\in \mathbf{b}_{TP}$) if it contained a target and it matched a ground-truth target.
- *True Negative (TN)*: a detection window was considered to be a true-negative element ($\in \mathbf{b}_{TN}$) if it did not contain a target and there was not any ground-truth target.
- *False Positive (FP)*: a detection window was considered to be a false-positive element ($\in \mathbf{b}_{FP}$) if it was claimed to contain a target and there was not any ground-truth target.
- *False Negative (FN)*: a detection window was considered to be a false-negative element ($\in \mathbf{b}_{FN}$) if it was claimed to not to contain a target and there was a ground-truth target.

Precision (PR) and Recall (RC) were computed at several levels according to the rank of the confidence score associating a detection window. The term “recall” is used here to indicate the proportion of truly detected cars to the total number of ground truth cars above a given confidence score, whereas the term “precision” indicates the proportion of truly detected cars to the total number of detections above that confidence score [24].

$$PR = \frac{\#\mathbf{b}_{TP}}{\#\mathbf{b}_{TP} + \#\mathbf{b}_{FP}} \quad (3.2)$$

$$RC = \frac{\#\mathbf{b}_{TP}}{\#\mathbf{b}_{GT}} \quad (3.3)$$

The second metric that was used to evaluate the performance is the 11-point interpolated Average Precision (AP). The aim of the interpolation process is to overcome the problem of having small variations in the confidence scores output by the used classifier. This metric is defined in Equation 3.4 and is computed at 11 evenly-spaced recall points $[0, 0.1, \dots, 1]$. According to [24], the AP metric penalises methods that are not able to retrieve all the ground truth data, i.e, methods that do not have an overall recall rate of 100%.

$$AP = \frac{1}{11} \sum_{k \in \{0, 0.1, \dots, 1\}} \mathbf{PR}_{interp}(k) \quad (3.4)$$

$$\mathbf{PR}_{interp}(k) = \max_{\tilde{k}: \tilde{k} \geq k} \mathbf{PR}(\tilde{k})$$

The third metric that was used is the Mean Average Precision (MAP) and it is computed using Equation 3.5 by adding the average precision of all of the true positives. This metric is different from the 11-point AP as it is not computed at equally-spaced discretised points and it does not penalise detection frameworks that

Table 3.1: Performance of baseline frameworks represented by the 11-point Average Precision AP .

Framework	Dataset	AP
Shao <i>et al.</i> [56]	Vaihingen	58.92%
Razakarivony and Jurie [58]	OIRDS	48.9%
Madhogaria <i>et al.</i> [60]	OIRDS	54.45%

do not detect all the ground truth targets. Consequently, AP gives more information about the achieved precision rates at low and high recall rates, unlike MAP that may provide only information at a limited range of recall rates.

$$MAP = \frac{1}{\#\mathbf{b}_{TP}} \sum_{k \in \{1, 2, \dots, \#\mathbf{b}_{TP}\}} \mathbf{PR}(B_{TP,k}) \quad (3.5)$$

The fourth metric that was used is the recall rate (RC) at different False-Positive-Per-Image ($FPPI$) rates and it is computed using Equation 3.3 by calculating the recall rate if 0.01, 0.1 and 1 false positives are allowed to exist in one image. For example, if the total number of images is 100. Recall rates are computed at three points, the first, when only one false positive is allowed, the second, when only ten false positives are allowed and the third, when only 100 false positives are allowed in the whole test dataset.

In addition to the quantitative metrics, a qualitative analysis was carried out for each proposed framework by visual inspection of sample regions of the test areas that were processed by the proposed frameworks.

3.3 Baseline Frameworks

In order to have a better evaluation of new devised methods, their achieved performance must be compared with the performance of the state-of-the-art frameworks. Three recent frameworks were chosen for this task of performance comparison as

they used the benchmark datasets that were described in Section 3.1 and not their own individual datasets. These three frameworks are the works of Shao *et al.* [56], Razakarivony and Jurie [58] and Madhogaria *et al.* [60]. They all adopt some form of implicit models and an overview of the image descriptors and classification algorithm used in these three frameworks was given in Section 2.2.2 of Chapter 2. Each of the baseline methods used a different training dataset.

Table 3.1 shows the computed 11-point average precisions AP of the framework of [56] on the Vaihingen dataset, their method achieved 58.92%. The framework of [58] achieved 48.9% on the OIRDS dataset using a 10-fold training-testing approach, whereas the framework of Madhogaria *et al.* [60] achieved 54.45% on the OIRDS. AP values were computed using the provided precision-recall curves setting the precision value to zero to penalise the devised method, if there is no given recall value.

It should be noted that due to the unavailability of the source code of the baseline frameworks, it was not possible to evaluate their performance on both datasets.



Figure 3.3: Sample regions of the images of the OIRDS dataset containing various cars in scenes with different backgrounds: (a) motorway area; (b) cars over imposed by shadow; (c) flat area; (d) black cars; (e) urban area; (f) parking area; (g) cars partially imposed by shadow; (h) parking area; (i) flat area; (j) urban area; (k) suburban area; (l) complex-shape area. Shown images were scaled for viewing.

3.4 Summary

This chapter described the two benchmark datasets, namely, the Vaihingen and the OIRDS datasets that were used for training and testing. It also discussed the evaluation methodology and performance metrics that were adopted to measure and compare the performance of the proposed frameworks. In addition, it presented the obtained results of baseline methods.

Chapter 4

Detection of Cars using Fixed Cell Distributions

THIS chapter focuses on the detection of cars in high-resolution urban scenes using flat implicit models based on the computation of an ensemble of image descriptors, namely, modified Histogram of Oriented Gradients, Fourier and truncated Pyramid Colour Self-Similarity. These descriptors are computed over fixed distributions of local cells, i.e., detection windows are divided into evenly-spaced equal-size cells. Hence, this proposed framework is called the Fixed-Cell-Distribution (FCD) framework.

The main contributions of the work that will be presented in this chapter include: i) the development of an efficient and simple pre-processing stage that is based on the physical and radiometric characteristics of cars and that is tailored to discard detection windows of low probabilities to contain cars in order to narrow search areas, ii) the development of rotation-invariant Histogram-of-Oriented-Gradients descriptor for targets that possess a dominant orientation for edges, iii) the introduction of truncated-Pyramid-Colour-Self-Similarity descriptor and iv) the design of a novel ensemble of image descriptors that robustly detects cars in high-

resolution urban scenes and that incorporates the local distribution of gradients, spectral information and local distribution of colours. The main points of strength of the FCD framework include: (i) efficient encoding of image cues with lower dimensionality and (ii) lower false-positive rate compared to the state of the art.

This chapter is organised as follows: in Section 4.1, an overview of the FCD framework is presented followed by the illustration of the implementation details of each stage of the adopted general four-stage methodology in Section 4.2, Section 4.3 and Section 4.4. Information about the training and testing datasets and the used classifier is given in Section 4.5. Obtained results are given and discussed in Section 4.6. A summary of the FCD framework is given in Section 4.7.

4.1 Overview of the Proposed FCD Framework

The first step to design a system for the detection of a particular target of interest is usually to analyse some patches that contain that target of interest and patches that contain other non-target objects in order to understand what makes this particular target of interest visually-different from non-target objects and from the surrounding background as it is considered a good practice. In other words, to search for visual differences between the foreground and the background. Visual differences can belong to various image cues, such as gradients and texture. This first step facilitates the challenging process of the choice of which image cues to rely on and encode using image descriptors. Theoretical foundations about this choice are still immature and only some guidelines are found in the open literature [10]. Consequently, in most cases, decisions are made by experimentation using empirical results.

In airborne images, patches that contain cars are expected to have parallelogram-like shapes that correspond to the external boundaries of the bodies of cars. This can be seen in Figure 4.1, where edges were extracted from 64×64

pixel patches from the Vaihingen dataset using the well-known Canny edge detector [63].

Image patches that are placed around a car usually reveal two small parallelograms that correspond to the front and rear windscreens and several additional almost parallel and/or almost perpendicular lines that resemble the external boundaries. These rectilinear structures form together a considerable number of points of interest, e.g., high-curvature points and corners, which are generally crucial to the detection process. Nevertheless, they are incapable of discriminating cars from other objects because other non-car objects, which have similar edge maps, such as buildings' roofs and windows, can be easily misidentified as cars. This can be seen in Figure 4.1 (g), where the patch that was placed around a window revealed a very similar edge structure to that of cars. Based on this, implicit models are superior to explicit models as they can capture fine details. In addition, a simple linear classifier can learn to distinguish well-encoded image descriptors that correspond to image cues related to cars from a large number of training samples as previously mentioned in Chapter 2.

Accordingly, to design an implicit model, a well-established gradient type of descriptors can be used as gradient descriptors have proved their effectiveness in many target detection applications in various real-life scenarios. For instance, it has been used for the detection of pedestrians [22]. However, it must be combined with another type of descriptors in order to be able to distinguish cars. By looking at Figure 4.1 again, it can be realised that cars exhibit some specific distribution of colours. Their body parts, the car bonnet, hood and roof, usually have the same colour and windscreens are almost always of black colour due to the reflection of sunlight. Hence, a colour descriptor can be combined with a gradient descriptor to form a more robust combined descriptor to distinguish cars as proposed later in this chapter.

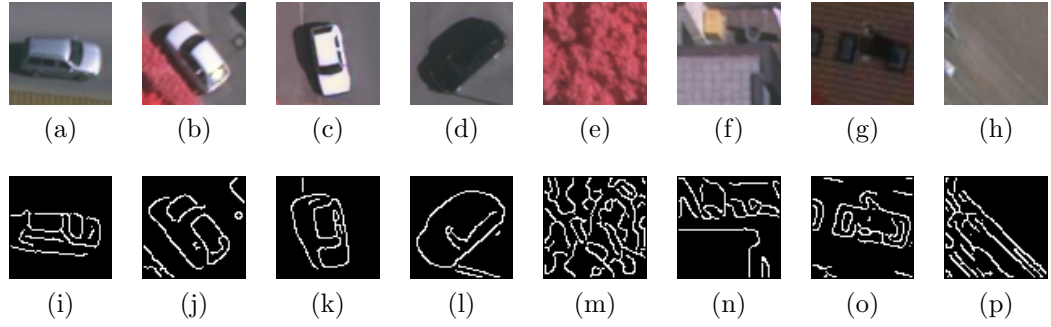


Figure 4.1: Extraction of edges in target-class and non-target-class patches using Canny edge detectors with an associated Gaussian filter of standard deviation of $\sqrt{2}$: Target-class patches are shown in (a), (b), (c) and (d); their corresponding edges in (i), (j), (k) and (l) respectively. Non-target-class patches are shown in (e), (f), (g) and (h); their corresponding edges in (m), (n), (o) and (p) respectively. Shown images were scaled for viewing.

The proposed FCD framework follows the adopted general four-stage methodology discussed in Chapter 1 that consists of a window-evaluation stage followed by the computation of three image descriptors, namely, rotation-invariant HOG, Fourier and tPCSS descriptors over a 64×64 pixel detection window. The associated stride length is eight pixels. These descriptors characterise the spatial distribution of pixels, spectral information and local colour distribution, respectively. Reasons for the choice of these descriptors will be discussed later in this chapter. A fixed-size overlapping sliding detection window has been used as cars in the Vaihingen dataset are of almost same size as it was captured from relatively high altitude. Feature descriptors are concatenated to form a single hybrid descriptor that is fed to a linear SVM classifier, which discriminates the car-class subspace from the non-car-class subspace. Finally, a non-maximum suppression technique is applied to eliminate multiple detections per single true target. Implementation details of this four-stage process are given in the following sections.

4.2 Window Evaluation

Airborne images of urban scenes usually contain large flat areas with constant or slowly-varying colour distribution that correspond to empty roads and vegetation. Such areas can be eliminated before the computation of image descriptors and before the classification process saving computational time. Therefore, how likely a detection window might contain car (have a high level of objectness [75]) is evaluated based on several image cues in a novel two-stage evaluation process and those detection windows with low probability to contain cars are eliminated.

First, for a given image $\mathbf{I}$, a matrix of the magnitudes of intensity gradients $\mathbf{M}_g(x, y)$ is computed over the detection window. This is done by the computation of the first-order discrete derivatives $\frac{\partial \mathbf{I}}{\partial x}$ and $\frac{\partial \mathbf{I}}{\partial y}$ of $\mathbf{I}$ using a non-smoothing 1-D filter with kernel of $[-1, 0, 1]$ vertically and horizontally for each colour channel $\mathbf{I}_1$, $\mathbf{I}_2$ and $\mathbf{I}_3$ separately and only the maximum gradient is kept using Equation 4.1 and Equation 4.2. If the number of gradients that have magnitude close to zero (less than 5) is greater than 1365 (one third of the detection window size), the detection window is ignored because this indicates that the detection window does not contain sufficient amount of gradients and most probably it belongs to a flat area, such as empty roads. Computed gradients over sample patches are shown in Figure 4.2. It is worth mentioning that this matrix of gradients does not incur extra computational cost as it will be used later in the computation of the rotation-invariant HOG descriptors.

$$\mathbf{M}_k(x, y) = \sqrt{\left(\frac{\partial \mathbf{I}_k(x, y)}{\partial x}\right)^2 + \left(\frac{\partial \mathbf{I}_k(x, y)}{\partial y}\right)^2} \quad (4.1)$$

$$\mathbf{M}_g(x, y) = \max_{k \in \{1, 2, 3\}} [\mathbf{M}_k(x, y)] \quad (4.2)$$

Second, a typical detection window that is placed around a car is expected to have two dominant greyscale intensities. The first grey intensity corresponds to the

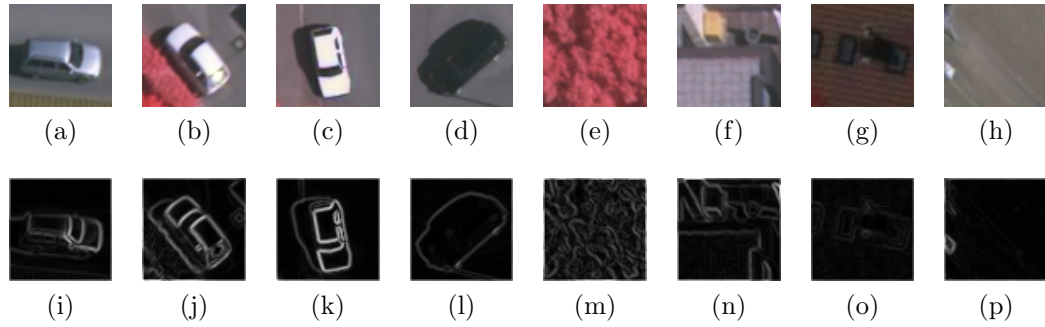
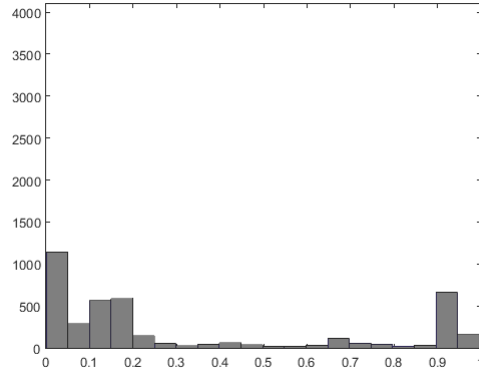


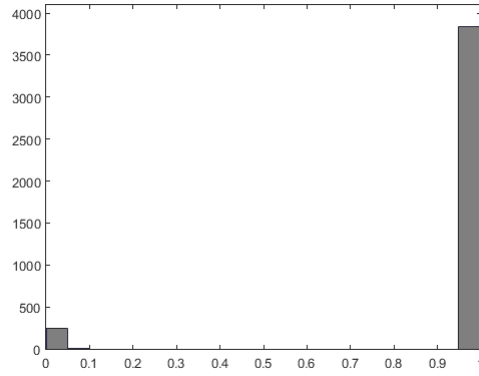
Figure 4.2: Computation of the maximum gradient in target-class and non-target-class patches: Target-class patches are shown in (a), (b), (c) and (d); their corresponding gradient matrices $\mathbf{M}_g$ in (i), (j), (k) and (l) respectively. Non-target-class patches are shown in (e), (f), (g) and (h); their corresponding gradient matrices $\mathbf{M}_g$ in (m), (n), (o) and (p) respectively. In some patches, gradients are not clear as they have low strength. Shown images were scaled for viewing.

colour of the car's body, whereas the second grey intensity corresponds to the colour of the surrounding background as can be observed from the colour distribution of the sample patches containing cars in Figure 4.2. Therefore, its colour histogram is expected to have a bi-modal shape as shown in Figure 4.3. In addition, the score of the bins of very dark shades, which correspond to the grey levels of front and rear windcreens, must exceed a pre-defined threshold. To evaluate this colour distribution property in a simple cost-effective way, the entire 8-bit RGB image is converted to an HSV image for which a 20-bin intensity histogram is created for the Hue channel. Detection windows that have a single mode with a very high score (greater than 2500) are discarded as these are of high probabilities to be parts of roads, vegetation areas or roofs of buildings. Detection windows that do not have a sufficient number (less than 500) of dark pixels of intensity less than 0.2 of the normalised Hue intensities are discarded as well.

Despite that this window-evaluation stage may seem to be simple, results of the experiments that were carried out showed its effectiveness in reducing regions that would be checked by the classifier and consequently reducing the processing time, as can be clearly seen in Figure 4.4, and the rate of false positives. Although a



(a)



(b)

Figure 4.3: Hue channel histograms of target-class and non-target class patches: (a) 20-bin colour histogram of bi-modal shape for a target-class patch; (b) 20-bin colour histogram of uni-modal shape for a non-target class patch of flat vegetation area.

substantial amount of regions has been eliminated, the number of successful window candidates is still high requiring a very robust descriptor to eliminate the high amount of clutter present in the used dataset as it can be seen in Figures 4.4 (b) and (d).

4.3 Extraction and Encoding of Features

As previously mentioned, three types of image descriptors are computed to encode three image features. They are computed over fixed cell distribution, i.e., the square



Figure 4.4: Application of the window-evaluation stage on a sample region of the training dataset: (a) and (b) original images; (c) and (d) corresponding processed images. Areas that have been eliminated by the window-evaluation stage as they are highly unlikely to contain cars are marked with black. Shown images were scaled for viewing.

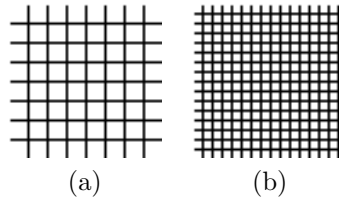


Figure 4.5: Fixed distributions of evenly-spaced cells of the same size: (a) cells of 8×8 pixels; (b) cells of 4×4 pixels.

detection window is divided into evenly-spaced cells with fixed sizes of 8×8 pixels and 4×4 pixels regardless of the orientation of cars. Cell distributions are shown in Figure 4.5. Implementation details of how the chosen image descriptors are computed are given in the following sections.

4.3.1 Rotation-Invariant Histogram-of-Oriented-Gradient Descriptor

Histogram-of-Oriented-Gradient descriptors were first introduced by Dalal and Triggs [22] mainly to solve the problem of pedestrian detection, i.e, people standing in upright position. Nonetheless, it has been recently exploited in many detection algorithms for targets of interest other than pedestrians [24]. It is a powerful single-feature descriptor that captures the spatial distribution of edge pixels [76].

HOG is computed in a very similar way to that of SIFT [21]. The major difference between the two descriptors is that HOG is computed over a dense grid. In order to compute the HOG descriptor for an image $\mathbf{I}$, the detection window is subdivided into evenly-spaced cells of 8×8 pixels as shown in Figure 4.5 (a). For each cell, magnitudes of gradients $\mathbf{M}_g(x, y)$ are computed using Equation 4.1 and Equation 4.2. Then, associated orientations $\mathbf{O}_g(x, y)$ are computed using Equation 4.3. Magnitudes $\mathbf{M}_g$ are accumulated in a 1-D histogram of evenly-spaced nine-orientation bins $\mathbf{g}_{cell}$ with a vote equals to a weighted magnitude of the associated gradient as in Equation 4.4. Orientations of unsigned gradients are used. In other words, the orientation interval starts at 0° and ends at 180° .

$$\mathbf{O}_g(x, y) = \arctan \left(\frac{\frac{\partial \mathbf{I}(x, y)}{\partial y}}{\frac{\partial \mathbf{I}(x, y)}{\partial x}} \right) \quad (4.3)$$

$$\mathbf{g}_{cell}(k) = \sum_{x, y} w(\mathbf{M}_g(x, y)) \mathbf{1}[\mathbf{O}_g(x, y) \in \mathbf{o}_k], k \in \{1, 2, \dots, 9\} \quad (4.4)$$

where $\mathbf{1}$ is a binary function that indicates whether the associated orientation belongs to the orientation interval $\mathbf{o}_k$. The weight of each vote w is computed using trilinear interpolation [77] in both position and orientation according to the system of Equations 4.5 devised by [77].

$$\mathbf{G}_{3D}(x_1, y_1, o_1) \leftarrow \mathbf{G}_{3D}(x_1, y_1, o_1) + w \cdot \mathbf{M}_g(x, y) | w = \left(1 - \frac{x - x_1}{b_x}\right) \left(1 - \frac{y - y_1}{b_y}\right) \left(1 - \frac{\mathbf{O}_g(x, y) - o_1}{b_o}\right) \quad (4.5a)$$

$$\mathbf{G}_{3D}(x_1, y_1, o_2) \leftarrow \mathbf{G}_{3D}(x_1, y_1, o_2) + w \cdot \mathbf{M}_g(x, y) | w = \left(1 - \frac{x - x_1}{b_x}\right) \left(1 - \frac{y - y_1}{b_y}\right) \left(\frac{\mathbf{O}_g(x, y) - o_1}{b_o}\right) \quad (4.5b)$$

$$\mathbf{G}_{3D}(x_1, y_2, o_1) \leftarrow \mathbf{G}_{3D}(x_1, y_2, o_1) + w \cdot \mathbf{M}_g(x, y) | w = \left(1 - \frac{x - x_1}{b_x}\right) \left(\frac{y - y_1}{b_y}\right) \left(1 - \frac{\mathbf{O}_g(x, y) - o_1}{b_o}\right) \quad (4.5c)$$

$$\mathbf{G}_{3D}(x_2, y_1, o_1) \leftarrow \mathbf{G}_{3D}(x_2, y_1, o_2) + w \cdot \mathbf{M}_g(x, y) | w = \left(\frac{x - x_1}{b_x}\right) \left(1 - \frac{y - y_1}{b_y}\right) \left(1 - \frac{\mathbf{O}_g(x, y) - o_1}{b_o}\right) \quad (4.5d)$$

$$\mathbf{G}_{3D}(x_1, y_2, o_2) \leftarrow \mathbf{G}_{3D}(x_1, y_2, o_2) + w \cdot \mathbf{M}_g(x, y) | w = \left(1 - \frac{x - x_1}{b_x}\right) \left(\frac{y - y_1}{b_y}\right) \left(\frac{\mathbf{O}_g(x, y) - o_1}{b_o}\right) \quad (4.5e)$$

$$\mathbf{G}_{3D}(x_2, y_1, o_2) \leftarrow \mathbf{G}_{3D}(x_2, y_1, o_2) + w \cdot \mathbf{M}_g(x, y) | w = \left(\frac{x - x_1}{b_x}\right) \left(1 - \frac{y - y_1}{b_y}\right) \left(\frac{\mathbf{O}_g(x, y) - o_1}{b_o}\right) \quad (4.5f)$$

$$\mathbf{G}_{3D}(x_2, y_2, o_1) \leftarrow \mathbf{G}_{3D}(x_2, y_2, o_1) + w \cdot \mathbf{M}_g(x, y) | w = \left(\frac{x - x_1}{b_x}\right) \left(\frac{y - y_1}{b_y}\right) \left(1 - \frac{\mathbf{O}_g(x, y) - o_1}{b_o}\right) \quad (4.5g)$$

$$\mathbf{G}_{3D}(x_2, y_2, o_2) \leftarrow \mathbf{G}_{3D}(x_2, y_2, o_2) + w \cdot \mathbf{M}_g(x, y) | w = \left(\frac{x - x_1}{b_x}\right) \left(\frac{y - y_1}{b_y}\right) \left(\frac{\mathbf{O}_g(x, y) - o_1}{b_o}\right) \quad (4.5h)$$

$$\begin{cases} b_x = 8, b_y = 8, b_o = 20 \\ x_1 \leq x < x_2, y_1 \leq y < y_2, o_1 \leq \mathbf{O}_g(x, y) < o_2 \end{cases}$$

where $\mathbf{G}_{3D}$ denotes a three-dimensional histogram, b_x , b_y and b_o represent the width of the histogram cell in the horizontal direction, vertical direction and orientation,

respectively, and x_1 and x_2 are the two nearest bins in the horizontal direction, y_1 and y_2 are the two nearest bins in the vertical direction and o_1 and o_2 are the two nearest bins in orientation.

A 16×16 pixel square block that consists of four overlapping cells is then formed. The contrast of each block of cells is locally normalised using $L2$ normalisation as in Equation 4.6, where $\mathbf{g}_{block}$ is the unnormalised descriptor and ϵ is a small constant that has been set empirically to 0.01. This normalisation step is very important in order to have a better invariance to illumination conditions [22]. The final HOG descriptor is formed by the concatenation of the responses from all blocks in a single vector of $1764D$.

$$\mathbf{g}_{block} \leftarrow \frac{\mathbf{g}_{block}}{\sqrt{\|\mathbf{g}_{block}\|_2^2 + \epsilon^2}} \quad (4.6)$$

The above original implementation of the HOG descriptor does not accommodate for rotation invariance if the amount of rotation is greater than the width of the orientation-histogram bins. However, as mentioned before in Section 4.1, an edge map of a car consists of a set of almost parallel and/or almost perpendicular lines. Therefore, if the dominant edge orientation O_{edge} within the patch is computed, it will be perpendicular to the orientation of the car $O_{dominant}$ in the original image. A global nine-bin histogram of orientations $\mathbf{g}_{window}$ is formed to compute the dominant edge orientation without the trilinear interpolation and the bins of the local histograms of oriented gradients of each cell are circularly shifted so that rotation invariance is achieved.

$$\mathbf{g}_{window}(k) = \sum_{x,y} (\mathbf{M}_g(x,y)) \mathbf{1}[\mathbf{O}_g(x,y) \in \mathbf{o}_k], k \in \{1, 2, \dots, 9\} \quad (4.7)$$

$$O_{edge} = (10 + 20(k - 1))^{\circ} | \mathbf{g}_{window}(k) = \max_{k=1,\dots,9} [\mathbf{g}_{window}] \quad (4.8)$$

$$O_{dominant} = \begin{cases} O_{edge} + 90^{\circ} & O_{edge} \leq 90^{\circ} \\ O_{edge} - 90^{\circ} & O_{edge} > 90^{\circ} \end{cases} \quad (4.9)$$

4.3.2 Fourier Descriptor

Although the HOG descriptor is very robust as it has been proven in many target detection scenarios [76], it fails to perform well in some cases, where the strength of gradients is low. For example, in the case of very dark cars as it can be seen in Figure 4.2 (1), the gradients around the boundaries of windcreens are with very low strength. Moreover, these cars can be found very close to textured surfaces, such as vegetation areas and building roofs making it hard for the classifier to distinguish between gradient distribution of cars and gradient distribution of textured surfaces.

As aforementioned boundaries of cars have rectilinear structures. Encoding this characteristic of cars using another descriptor and incorporating it with HOG would result in a more robust combined image descriptor. For these reasons, the Fourier descriptor is exploited. It captures the spectral information of a given image. In addition, it has a very important feature that distinguishes rectilinear shapes. If an edge line exists in the original patch (spatial domain), it will appear as a line in the spectral domain but with an orientation perpendicular to its original orientation as shown in Figure 4.7. This distinctive feature can be exploited to differentiate cars as they mainly consist of dominant line edges with almost parallel/perpendicular orientations. On the contrary, other non-car objects are expected to have edges with different orientations as shown in Figure 4.7. As a result, the Fourier descriptor of a window containing a car has a very distinctive shape, wherein most of the

energy is concentrated along the vertical and horizontal axes. Hence, the feature vector of Fourier coefficients for a car patch will be different from that of a non-car patch, wherein in the later energy is expected to be distributed among most of the coefficients as in Figure 4.7 (p).

Fourier descriptors have been used in target detection with many variants and modifications [78], [79], [80], [81]. Nonetheless, the simplest form of the Fourier descriptors has been used in the proposed FCD framework as it is the least computationally expensive. The 2-D discrete Fourier Transform for an image $\mathbf{I}$ of dimensions X and Y can be defined as [82]:

$$\mathbf{F}(u, v) = \frac{1}{XY} \sum_x \sum_y \mathbf{I}(x, y) \cdot e^{-j2\pi\left(\frac{ux}{X} + \frac{vy}{Y}\right)} \quad (4.10)$$

where x and y are the spatial co-ordinates and u and v are two real frequency variables.

To compute Fourier descriptors, detection windows are first rotated so that the dominant orientation will be vertical using the global orientation calculated in Equation 4.7. By doing this, detection windows that have cars will have a vertical orientation. Rotated windows are then cropped keeping the central square region of 48×48 pixels such that extra padded parts, which are added because of the rotation process, are removed as shown in Figure 4.6.

After that, the Fast Fourier Transform, which is a fast implementation of Equation 4.10, of a grey-scale version of the image is computed. Finally, the obtained spectrum is shifted, wherein the two-dimensional frequency pair $\mathbf{F}(1, 1)$ is transferred from the top left corner of the given image patch to the middle (centre) of it. The Fourier feature vector $\mathbf{f}_{window}$ is formed by reshaping the matrix that contains the magnitudes of Fourier Transform coefficients into a single vector of $2304D$ and it is then normalised using $L2$ normalisation with ϵ equals to 0.01.

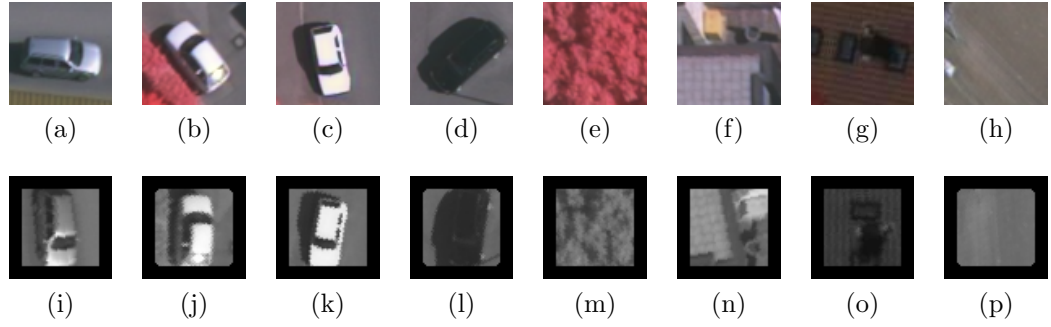


Figure 4.6: Estimation of the dominant orientation in target-class and non-target-class patches followed by rotation so that targets have vertical orientations: Target-class patches are shown in (a), (b), (c) and (d); their corresponding cropped rotated versions in (i), (j), (k) and (l) respectively. Non-target-class patches are shown in (e), (f), (g) and (h); their corresponding cropped rotated versions in (m), (n), (o) and (p) respectively. Estimated dominant orientations were: (a) $O_{dominant} = 170^\circ$; (b) $O_{dominant} = 130^\circ$; (c) $O_{dominant} = 110^\circ$; (d) $O_{dominant} = 50^\circ$; (e) $O_{dominant} = 110^\circ$; (f) $O_{dominant} = 170^\circ$; (g) $O_{dominant} = 170^\circ$; (h) $O_{dominant} = 130^\circ$. Shown images were scaled for viewing.

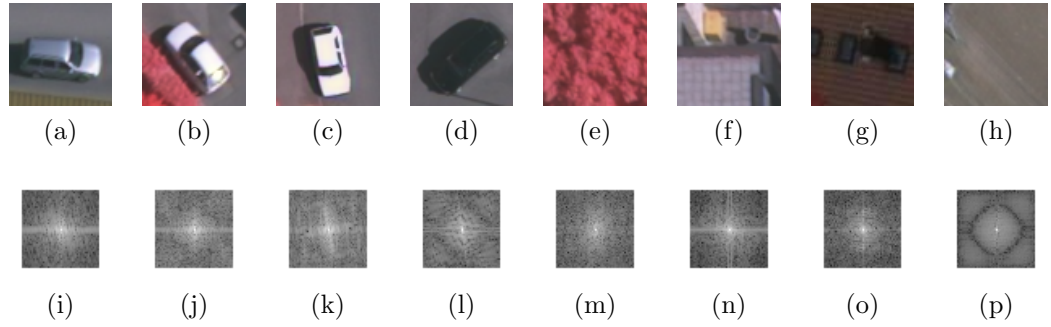


Figure 4.7: Computation of the Fourier descriptor in target-class and non-target-class patches: Target-class patches are shown in (a), (b), (c) and (d); their corresponding Fourier Transform coefficients in (i), (j), (k) and (l) respectively. Non-target-class patches are shown in (e), (f), (g) and (h); their corresponding corresponding Fourier Transform coefficients in (m), (n), (o) and (p) respectively. Fourier Transform was computed after rotating the original patch. Shown images were scaled for viewing.

$$\mathbf{f}_{window} \leftarrow \frac{\mathbf{f}_{window}}{\sqrt{\|\mathbf{f}_{window}\|_2^2 + \epsilon^2}} \quad (4.11)$$

4.3.3 Truncated-Pyramid-Colour-Self-Similarity Descriptor

The edge and gradient maps of buildings' roofs and windows, shown in Figures 4.1 (o) and 4.2 (o), respectively, reveal very similar maps to those of typical cars. In such cases, both HOG and Fourier descriptors fail to distinguish cars and the classifier would usually identify such non-car objects as true positives. In order to mitigate this problem, another image cue should be used alongside the gradients, such as the distribution of colours. Self-similarity descriptors are regularly used to encode the local or global similarity between pixel intensities in a given image. This allows the description of the colour distribution independently from the absolute pixel intensity value as cars can possess any colour.

Regarding a detection window that is placed around a car, it is expected that adjacent pixels that belong to the parts of the car body, such as the car bonnet, hood and roof, should exhibit a great similarity. Furthermore, there is always a local variation between pixels that belong to windcreens and pixels that belong to parts of the car body. Consequently, using a colour self-similarity descriptor will depict these distinctive features of local similarity and variation across intensity levels.

A modified version of the colour self-similarity descriptors used by [83] and [84] is proposed to complement HOG and Fourier descriptors and it is called truncated Pyramid Colour Self-Similarity (tPCSS). This new self-similarity descriptor depends on the calculation of the intersection of local single-colour histograms to measure the similarity between two histograms $\mathbf{h}_c$ and $\mathbf{h}_n$ for the central cell and a neighbouring cell, respectively.

$$\mathbf{s}_{cell} = [\mathbf{h}_c \cap \mathbf{h}_n] = \left[\left[\sum_k \min[\mathbf{h}_c(k), \mathbf{h}_n(k)] \right] \right], n \in \{1, 2, \dots, 8\}, k \in \{1, 2, \dots, 16\} \quad (4.12)$$

where $\lceil \cdot \rceil$ indicates that the result is truncated at the size of an unsigned integer of eight bits. This truncation was found empirically to be essential in order to have a high recall rate as it tolerates some variation among true-positive patches, e.g., black cars as they possess slightly different colour distribution from other cars. The intersection is computed with the eight neighbouring cells.

To compute tPCSS, the extracted rotated cropped grey-scale detection window in the previous stage is divided into a number of square cells. The descriptor is then computed on two levels separately. In level one, the cell size is 4×4 pixels and it has a fixed distribution, shown in Figure 4.5 (b), whereas in level two, the cell size is 8×8 and it has a fixed distribution, shown in Figure 4.5 (b). The tPCSS vector is formed by the concatenation of the $L2$ -normalised responses of all cells in the two levels of the pyramid with ϵ equals to 0.01 and it has $928D$.

$$\mathbf{s}_{cell} \leftarrow \frac{\mathbf{s}_{cell}}{\sqrt{\|\mathbf{s}_{cell}\|_2^2 + \epsilon^2}} \quad (4.13)$$

All values of thresholds and parameters used in the window-evaluation and descriptor-computation stages were found by testing excessively the properties of a large number of image patches from the training dataset.

4.4 Post-Processing

Since an overlapping sliding detection window has been adopted in our framework, multiple detections per one single true detection are expected. Bounding boxes that overlap by more than 50% using Equation 3.1 are eliminated except the one that has the highest output confidence score [24].

4.5 Datasets and Classification

The proposed overall extended feature vector is formed by the concatenation of the computed ensemble of descriptors: 1764 HOG features, 2304 Fourier features and 928 tPCSS features. The total length of this new combined feature vector is 4996 features. Since this problem of classification is characterised by having relatively high dimensionality, it is recommended to adopt an SVM classifier that is known to work efficiently in high-dimensional scenarios [39]. Based on that and following Occam’s razor “Among competing hypotheses, the one with the fewest assumptions should be selected”, a linear SVM classifier has been adopted in the proposed FCD framework to discriminate the car subspace from the non-car subspace with a regularisation parameter c equals 15, which was chosen using cross validation on the training dataset. The SVM classifier is fed by the formed descriptor vector to distinguish between the car-class subspace and the non-car-class subspace.

In order to classify a new data sample s , the feature vector is computed using the transformation $\phi(s)$ that transforms this new data sample to a hyperspace of dimensionality $4996D$. The distance from the separating hyperplane is calculated using the optimal weight $\mathbf{w}_{opt}$ and bias b using Equation 4.14. The sign of $z(s)$ determines on which side of the hyperplane the data sample s is, i.e, to which subspace the data sample belongs. The distance from the hyperplane represents its confidence score. A detailed mathematical analysis of how SVM classifier works is presented in appendix A.

$$z(s) = \mathbf{w}_{opt}^T \phi(s) + b \quad (4.14)$$

The classifier was trained on samples extracted from the training areas of the Vaihingen dataset. Sample examples are shown in Figure 3.1. For the positive training data, 754 positive examples that contained cars placed at different ori-

entations and on various backgrounds and their mirrored and horizontally-flipped versions were used to increase the diversity among the training samples. For the negative training data, a total of 3601 negative examples including patches extracted randomly from various backgrounds and bootstrapped hard negatives that were identified as true positives during the validation process on areas other than test areas.

4.6 Results

The proposed FCD framework was run on the test areas of the Vaihingen dataset that were described in Section 3.1.1 and such data was not included in the data over which the SVM classifier was trained. In addition, it has been run on the OIRDS dataset that constitutes an unseen dataset to the trained classifier. The performance was assessed and evaluated using qualitative analysis by the visual inspection of the bounding boxes that were produced as a result of the application the proposed framework on the data and various quantitative measures, namely, the precision-recall, the mean average precision, recall rates at different false-positive-per-image rates and the interpolated average precision that were described in Section 3.2. Achieved results were the outcome of a single experiment conducted on each of the two datasets.

4.6.1 Qualitative Results

Visual results of the experiments of applying the FCD framework on sample regions of the test areas of the Vaihingen dataset are shown in Figure 4.8. The proposed framework is able to detect cars of different colours at different orientations. The true-positive and false-positive rates are acceptable and this will be corroborated in the next section using quantitative analysis. Nonetheless, in some challenging cases,

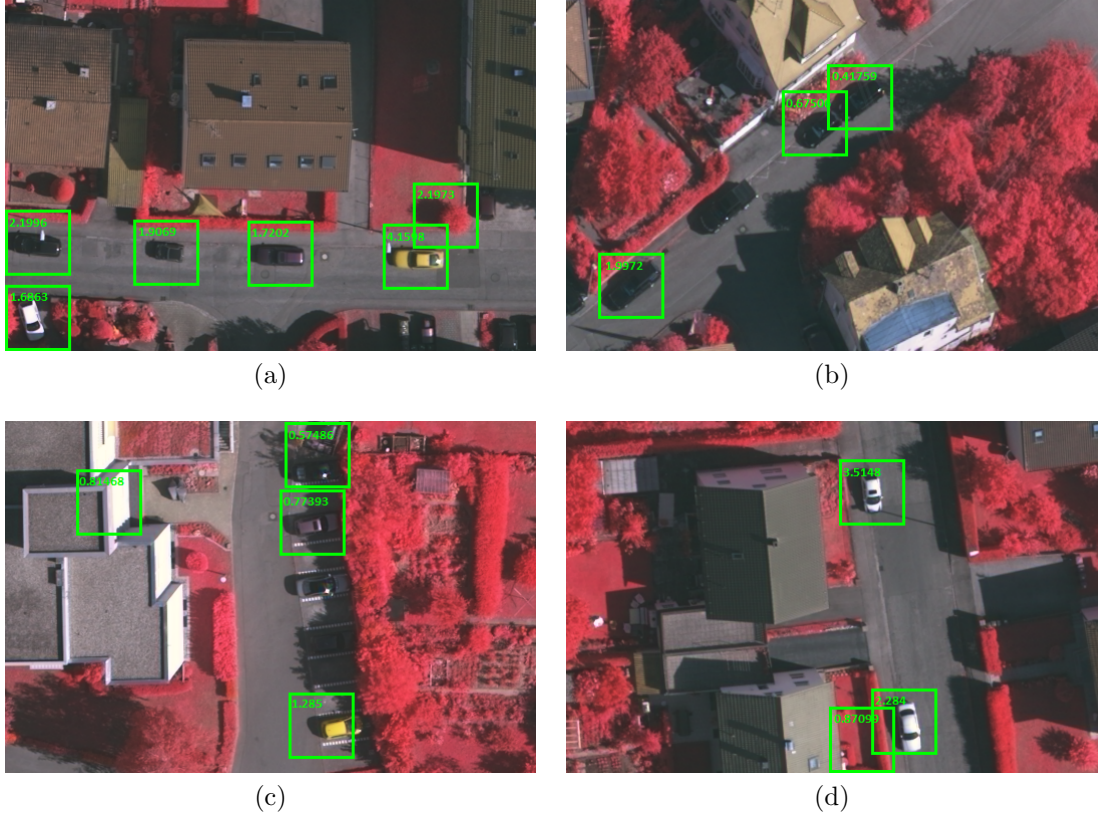


Figure 4.8: Sample regions of the test areas of the Vaihingen dataset after applying the proposed FCD framework. Bounding boxes represent detections and the number at the top left corner of each bounding box represents its confidence score. Shown images were scaled for viewing.

wherein non-car objects possess very similar structure and colour distribution to that of cars, such as the case in Figure 4.8 (a), the tree and its shadow, the proposed framework fails to distinguish cars from such objects. Moreover, in Figures 4.8 (b) and (c), cars of very dark colours or in areas over-imposed by strong shadow were missed.

Figure 4.9 shows the obtained results after applying the FCD framework on sample images of the OIRDS dataset. It can be deduced from this figure that the proposed FCD framework has acceptable generalisation in terms of the true detections and false positives, when applied to images with lower resolution and that were acquired under different conditions from those over which the classifier was trained.



Figure 4.9: Sample regions of the test images of the OIRDS dataset after applying the proposed FCD framework. Bounding boxes represent detections and the number at the top left corner of each bounding box represents its confidence score. Shown images were scaled for viewing.

Moreover, in Figure 4.8 and Figure 4.9, it can be noted that black cars tend to be detected with low confidence scores as opposed to white cars. This is because of two reasons: (i) the strength of gradients is lower and (ii) the differences and similarities in the colour distribution among different parts of the car body are slightly different from cars with bright colours.

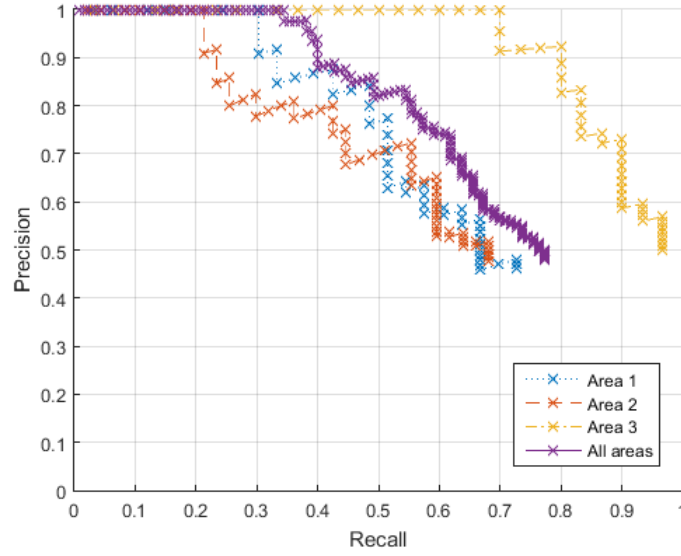


Figure 4.10: Performance of the proposed FCD framework in different test areas of the Vaihingen dataset represented by precision rate at different recall rates.

4.6.2 Quantitative Results

Performance of the Proposed FCD Framework

Figure 4.10 shows the obtained precision-recall curves that were computed using Equation 3.2 and Equation 3.3 discussed in Chapter 3, when applied to the three test areas of the Vaihingen dataset. It can be seen that it has the best performance, represented by an precision rate of 50.00% at a recall rate of 96.67%, in urban regions that contain small detached houses (area 3). Nevertheless, the performance degrades to a precision rate of 47.76% at a recall rate of 68.09% in urban regions that contain high-rising residential buildings (area 2) because in these areas, edge maps and colour distributions of the roofs of buildings, windows, patios and balconies are very similar to those of cars. The overall precision and recall rates in the whole testing areas are 48.02% and 77.72%, respectively.

Figure 4.11 shows the obtained precision-recall curves, when applied to the OIRDS dataset. The overall precision and recall rates in the whole testing areas

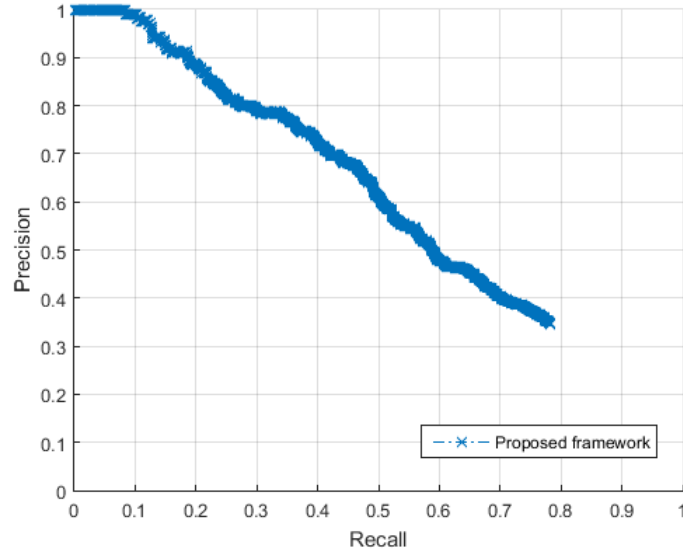


Figure 4.11: Performance of the proposed FCD framework on the OIRDS dataset represented by precision rates at different recall rates.

Table 4.1: Performance of the proposed FCD framework on the Vaihingen and OIRDS datasets represented by the Mean Average Precision MAP and recall RC rates at different false positive per image $FPPI$ values.

Dataset	MAP	RC		
		$0.01FPPI$	$0.1FPPI$	$1.0FPPI$
Vaihingen	75.10%	34.55%	49.09%	77.27%
OIRDS	70.41%	12.84%	29.95%	65.30%

are 34.55% and 78.23%, respectively. It can be observed that the obtained curve is smoother than the obtained precision-recall curve from the Vaihingen dataset. The reason for this is that the OIRDS has a larger number of cars.

Table 4.1 shows the mean average precision values of the obtained results of the proposed framework and recall rates at different false-positive-per-image rates for both the Vaihingen and the OIRDS datasets.

Accuracy of the Estimation of the Orientation of Cars

The FCD framework uses a global gradient histogram to estimate the orientation of the dominant edge as previously discussed in Section 4.3.1, which is expected to be

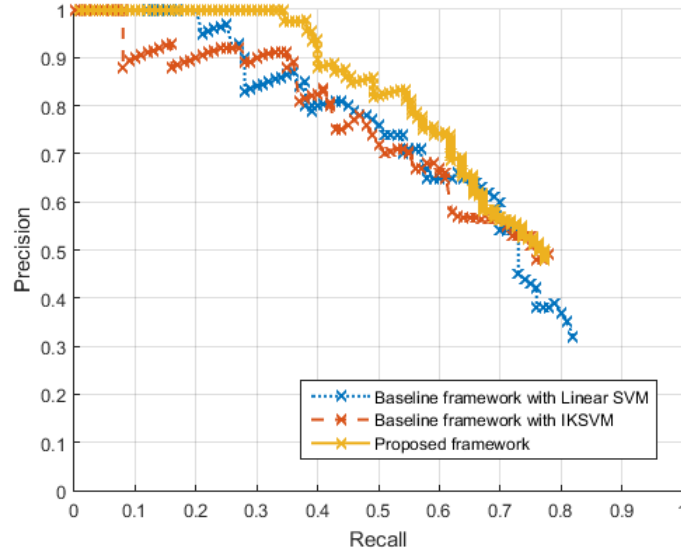


Figure 4.12: Performance of the proposed FCD framework and the baseline framework of [56] on the Vaihingen dataset represented by precision rates at different recall rates.

perpendicular to the orientation of car. The accuracy of estimating this dominant edge orientation is 90.52%, when it was tested on the Vaihingen dataset.

Comparison with the State of the Art

Figure 4.12 shows the obtained precision-recall curves for the proposed framework and the framework of Shao *et al.* [56]. It can be deduced from these curves that for the same recall the proposed framework achieves higher precision than that of framework [56] using linear classification, unlike the standard implementation of [56] that adopts an IKSVM classifier, which incurs longer computational time. Moreover the proposed algorithm is more efficient by using lower dimensional feature vector than that of [56] by 1764D.

Furthermore, the performance of the proposed framework has an overall 11-point average precision, described in Section 3.2, higher than that of framework [56] on the Vaihingen, despite the fact that this metric penalises methods that retrieve only a subset of the ground truth [24] and an 11-point average precision higher than

Table 4.2: Performance of the proposed FCD framework and the baseline frameworks represented by the 11-point Average Precision AP .

Framework	Dataset	AP
FCD	Vaihingen	64.26%
Shao <i>et al.</i> [56]	Vaihingen	58.92%
FCD	OIRDS	53.55%
Razakarivony and Jurie [58]	OIRDS	48.9%
Madhogaria <i>et al.</i> [60]	OIRDS	54.45%

that of framework [58] on the OIRDS dataset as shown in Table 4.2. Moreover, it is worth to mention that the framework of Razakarivony and Jurie [58] used a 10-fold training/testing methodology for the OIRDS dataset, whereas the whole data was completely unseen for the proposed FCD framework. The framework of Madhogaria *et al.* [60] achieved slightly higher precision - by less than 1% - than the FCD framework. However, the authors used a larger training dataset compared to the training dataset the FCD framework exploited and their method resulted in false-positives with very high confidence scores compared to the confidence scores of the true positives.

It is also worth mentioning that due to the unavailability of the source code of the baseline frameworks, it was not feasible to evaluate their performance on both datasets as previously mentioned.

The FCD framework has been implemented in a MATLAB environment. All of the experiments were run on a computer with the following specifications: a 2.5-GHz-speed Intel-Core-i5 CPU and a 6-GB Random Access Memory. The machine that was used to conduct the experiments has multiple processing cores. Nevertheless, the current implementation uses only a single core. The current implementation is able to process ≈ 76 detection windows per second, which indicates that it has a near real-time performance.

4.7 Summary

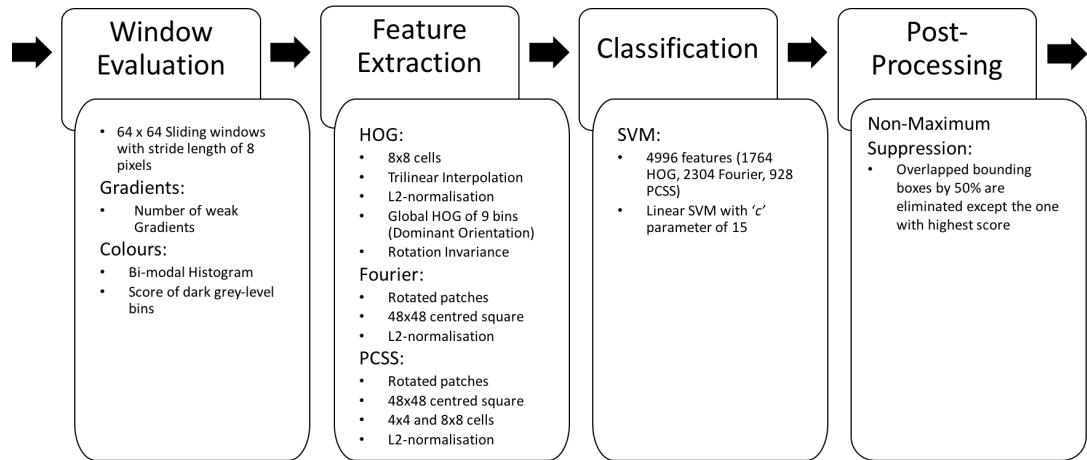


Figure 4.13: Summary of the FCD framework for the detection of cars in airborne imagery using fixed cell distributions.

This chapter introduced the first proposed framework for the detection of cars in aerial images of urban scenes, namely, the Fixed-Cell-Distribution framework. It consists of four main stages. Implementation details were given and discussed. A summary is given in Figure 4.13. Image descriptors that are used in the proposed framework are: rotation-invariant Histogram of Oriented Gradients, Fourier and truncated Pyramid Colour Self-Similarity. The proposed Fixed-Cell-Distribution framework outperforms baseline frameworks as it achieves higher average precision of 64.26% and 53.55% on the Vaihingen and OIRDS datasets, respectively.

Chapter 5

Detection of Cars Using Adaptive Cell Distributions

IN Chapter 4, it was shown that the boundaries of a car in a given aerial image are expected to have a rectilinear structure that corresponds to the edges of its body and front and rear windscreens. Furthermore, it was shown that cars exhibit specific distribution of colours. Accordingly, an ensemble of modified HOG, Fourier, tPCSS descriptors was proposed to depict these characteristic features of cars. Nevertheless, the FCD method failed to detect with high confidence scores cars that were fully covered by shadows and some cars that have black colour. Moreover, in some confounding situations of having a non-car class object that possess similar visual characteristics, it was classified as a true positive as it was shown in Figure 4.8 and Figure 4.9.

In order to solve the above-mentioned problems, this chapter introduces a new method to evaluate detection windows and exploits a novel ensemble of modified image descriptors, namely, Histogram of Oriented Gradients, truncated Pyramid Colour Self Similarity and Gabor and the eigenvalues of the covariance matrices of the local distribution of each of these descriptors. These three image descrip-

tors are computed over novel adaptive cell distributions using a clever method that changes the distribution of local cells according to the dominant orientation of the patch under examination and hence this new framework is called the Adaptive-Cell-Distribution (ACD) framework.

The main contributions of the work that will be presented in this chapter include: (i) introduction of a novel method to decrease search areas by evaluating the probability that detection windows may be placed around cars based on colour, gradient and texture image cues, (ii) very accurate estimation of the orientation of cars using the second-order statistics of local gradients and (iii) detection of cars more robustly with low false-positive rate by using a novel ensemble of image descriptors and the eigenvalues of their covariance matrices. This novel ensemble incorporates the local distribution of gradients, the local distribution of colours and the local distribution of texture. The main points of strength of the ACD framework include: (i) efficient encoding of image cues with lower dimensionality, (ii) lower false-positive rate and (iii) shorter processing time than the FCD framework.

This chapter is organised as follows: in Section 5.1, an overview of the proposed framework is presented followed by the illustration of the implementation details of each stage in Section 5.2, Section 5.3, Section 5.4 and Section 5.5. Information about the training and testing datasets and classification is given in Section 5.6. Obtained results are given and discussed in Section 5.7. A summary of the ACD framework is given in Section 5.8.

5.1 Overview of the Proposed ACD Framework

The ACD framework follows the adopted general four-stage methodology that was discussed in Chapter 1. It consists of a window-evaluation stage followed by the computation of three modified image descriptors, namely, HOG, tPCSS and Gabor

descriptors that are used to characterise the distribution of gradients, colours and texture, respectively. In addition, eigenvalues of their associated covariance matrices are incorporated. Image descriptors are computed over a grey-scale fixed-size window of 64×64 pixels and associated stride length of four pixels as cars have almost similar sizes in a set of aerial images captured at a given GSD, e.g., the GSD of the Vaihingen dataset is 8cm . Square patches were chosen to facilitate the computation of image descriptors, as discussed before in Chapter 4. Feature descriptors are concatenated to form a single hybrid descriptor that is fed to a linear SVM classifier, which divides the hyperspace into car-class subspace and non-car-class subspace. Finally, a non-maximum suppression technique is exploited to discard multiple detections per single true target. Details of the implementation of each stage are discussed in the following sections.

5.2 Window Evaluation

To reduce search areas, detection windows that are highly unlikely to contain cars are investigated during the window-evaluation stage based on several cues (colour distribution, texture and gradients) to shorten the processing time and help reduce the false-positive rate using a three-step filtering process that measures the objectness [75] of the detection windows and how likely that they would contain cars in a novel way.

First, the variation in the distribution of grey levels across a detection window is examined by the computation of two 32-bin colour histograms $\mathbf{h}_w$ and $\mathbf{h}_c$, where the first colour histogram is calculated for the whole detection window and the second is calculated for the 32×32 central square area. The cost of matching between these two intensity histograms is measured using the Manhattan distance $r_{\text{Manhattan}}$ according to Equation 5.1. If the distance is found to be below a pre-

defined threshold, the window is discarded as this highly indicates that it belongs to a flat area with a constant or a slowly-varying colour distribution, such as empty roads and vegetation areas.

$$r_{Manhattan} = \sum_{k=1}^{32} |\mathbf{h}_w(k) - \mathbf{h}_c(k)| \quad (5.1)$$

Second, the texture of the grey-scale detection window is examined in a new fashion. Pixels of each local cell of 4×4 pixels are divided into two groups according to whether the pixel value is larger than the average value of the cell or smaller. Then, the difference between the averages of the two groups is calculated. If the standard deviation $\sigma_{texture}$ of the responses of the computed differences is larger than a pre-defined threshold, the detection window will not be considered for further processing as a high standard deviation $\sigma_{texture}$ means that there is a highly textured surface that cannot be a car taking into account that cars tend to have smooth surfaces.

Third, for a given image $\mathbf{I}$, magnitudes of intensity gradients $\mathbf{M}_g(x, y)$ at each pixel and their orientations $\mathbf{O}_g(x, y)$ are calculated over the grey-scale detection window. This is done by the computation of the first-order discrete derivatives $\frac{\partial \mathbf{I}}{\partial x}$ and $\frac{\partial \mathbf{I}}{\partial y}$ of $\mathbf{I}$ using a non-smoothing 1-D filter with kernel of $[-1 \ 0 \ 1]$ in both vertical and horizontal directions. Then, the magnitudes of gradients of local cells of 4×4 pixels are accumulated independently in nine-bin histograms $\mathbf{g}_{cell}$ according to their unsigned orientation without interpolation using Equation 5.2, Equation 5.3 and Equation 5.4.

$$\mathbf{M}_g(x, y) = \sqrt{\left(\frac{\partial \mathbf{I}(x, y)}{\partial x}\right)^2 + \left(\frac{\partial \mathbf{I}(x, y)}{\partial y}\right)^2} \quad (5.2)$$

$$\mathbf{O}_g(x, y) = \arctan \left(\frac{\frac{\partial \mathbf{I}(x, y)}{\partial y}}{\frac{\partial \mathbf{I}(x, y)}{\partial x}} \right) \quad (5.3)$$

$$\mathbf{g}_{cell}(k) = \sum_{x, y} \mathbf{M}_g(x, y) \mathbf{1} [\mathbf{O}_g(x, y) \in \mathbf{o}_k], k \in \{1, 2, \dots, 9\} \quad (5.4)$$

where $\mathbf{1}$ is a binary function that indicates whether the associated orientation corresponds to the k^{th} bin. Then the covariance matrix $\mathbb{R}_{GG}$ of the $L2$ -normalised local cell responses $\mathbf{g}_{cell}$ is computed using an ϵ of value equal to 0.01.

$$\mathbf{g}_{cell} \leftarrow \frac{\mathbf{g}_{cell}}{\sqrt{\|\mathbf{g}_{cell}\|_2^2 + \epsilon^2}} \quad (5.5)$$

$$\mathbb{R}_{GG} = \mathbb{E} \{ \mathbf{G} \mathbf{G}^H \} \quad (5.6)$$

For a typical detection window containing a car, there should be a dominant orientation of the gradients that corresponds to the direction of the perpendicular to the orientation of the car in a given image. Therefore, if the covariance matrix $\mathbb{R}_{GG}$ is eigen-decomposed, the maximum eigenvalue $d_{G, max}$ must exceed a pre-defined threshold, otherwise the window is discarded. The intuition behind this is that covariance matrix $\mathbb{R}_{GG}$ represents the second-order statistics of the received signal. It can be used to determine the direction of the dominant emitting source using the order of its significant eigenvalues $d_{G, k}$ [85]. In other words, this orientation bin that corresponds to the dominant orientation will have the highest received power represented by the maximum eigenvalue.

$$d_{G, max} = \max_{k=1, \dots, 9} [eig(\mathbb{R}_{GG})] \quad (5.7)$$

The above mentioned three-step process of window filtering helps reduce the

Table 5.1: Threshold values used in the window-evaluation stage of the ACD framework.

Threshold	Value
$r_{Manhattan}$	≤ 2200
$\sigma_{texture}$	≤ 30
d_{max}	≥ 0.09

search areas and the rate of the false positives. Table 5.1 summarises the values of the thresholds used in this stage. All values of the thresholds and parameters used in the implementation of the window-evaluation stage were found empirically by testing excessively the properties of a large number of image samples from the training dataset.

Figure 5.1 shows a sample region of the training dataset that was processed using the proposed window-evaluation stage. It can be seen that a substantial amount of the search areas was eliminated and areas that are very likely to contain targets were kept. On comparing the output of the new proposed method for evaluating detection windows with the output of the window-evaluation stage in Chapter 4 that was shown in Figure 4.4, it can be deduced that the new proposed method is more effective in eliminating detection windows of low probability to contain targets. In addition, it does not use RGB colour information indicating that it can be further extended to work for images of different colour properties.

5.3 Estimation of the Orientations of Cars

Unlike the FCD framework that estimates the dominant orientation of the patch under examination using a global histogram of oriented gradients, the ACD framework proposes the use of the second-order statistics of local gradients in a novel, simple and robust way.

The orientation of cars $O_{dominant}$ is discretised into nine orientations at 10° ,

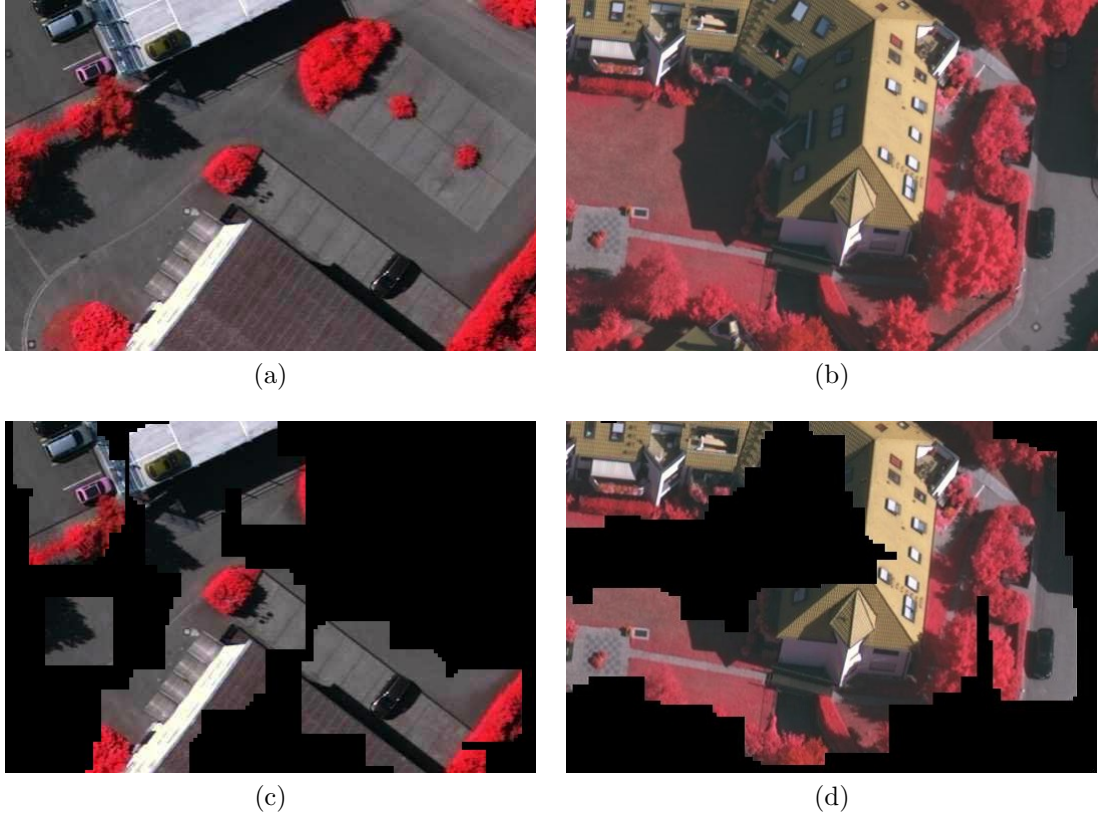


Figure 5.1: Application of the window-evaluation stage on a sample region of the training dataset: (a) and (b) original image; (c) and (d) corresponding processed image. Areas that have been eliminated by the window-evaluation stage as they were highly unlikely to contain are marked with black. Shown images were scaled for viewing.

$30^\circ, 50^\circ, 70^\circ, 90^\circ, 110^\circ, 130^\circ, 150^\circ$ and 170° . It is estimated to be in a perpendicular orientation $O_{d_{G,max}}$ to the bin orientation corresponding to maximum eigenvalue $d_{g,max}$ of the covariance matrix $\mathbb{R}_{GG}$ of locally accumulated gradients that were computed over 4×4 cells in the window-evaluation stage. Figure 5.2 shows the output of the proposed orientation-estimation stage over target and non-target class patches. This novel method of estimating the dominant orientation achieves very high accuracy. A comparison between the accuracy of using a global HOG and the second-order statistics will be carried out later in Section 5.7.2.

$$O_{d_{G,max}} = (10 + 20(k - 1))^\circ \mid d_{G,k} = \max_{k=1,\dots,9} [eig(\mathbb{R}_{GG})] \quad (5.8)$$

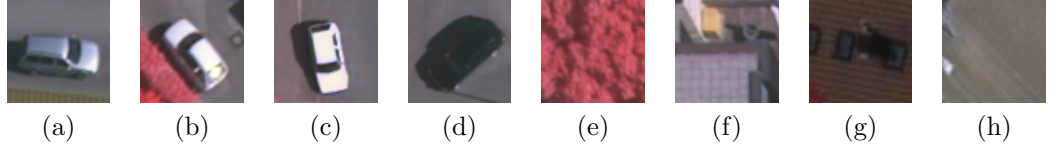


Figure 5.2: Estimation of the dominant orientation in target-class and non-target-class patches: Target-class patches are shown in (a), (b), (c) and (d). Non-target-class patches are shown in (e), (f), (g) and (h). Estimated dominant orientations were: (a) $O_{dominant} = 170^\circ$; (b) $O_{dominant} = 130^\circ$; (c) $O_{dominant} = 110^\circ$; (d) $O_{dominant} = 50^\circ$; (e) $O_{dominant} = 110^\circ$; (f) $O_{dominant} = 170^\circ$; (g) $O_{dominant} = 170^\circ$; (h) $O_{dominant} = 130^\circ$. Shown images were scaled for viewing.

$$O_{dominant} = \begin{cases} O_{d_{G,max}} + 90^\circ & O_{d_{G,max}} \leq 90^\circ \\ O_{d_{G,max}} - 90^\circ & O_{d_{G,max}} > 90^\circ \end{cases} \quad (5.9)$$

5.4 Extraction and Encoding of Features

In Chapter 4, it was proposed to divide the detection window into evenly-distributed square cells over which image descriptors are computed locally. For rotation-variant features, it was proposed to rotate the detection window so that all cars would have vertical orientation. However, this rotation process is time consuming and it affects the relative distribution of the original pixels due to the non-linear transformation between co-ordinates. To overcome this problem, adaptive cell distributions are proposed, wherein the shape of cells are no longer fixed-size squares and the distribution is a function of the dominant orientation.

According to the estimated orientation of a detection window containing a car, a suitable mask of cell distribution is chosen. Eighteen steerable masks are used in the ACD framework and they are shown in Figure 5.3. Masks in subfigures (a), (b), (c), (d), (e), (f), (g), (h) and (i) of Figure 5.3 were originated from the rotation of a mask oriented horizontally of size 64×50 pixels and divided into ordered cells of size 8×10 pixels to the previously mentioned discretised orientation, where as

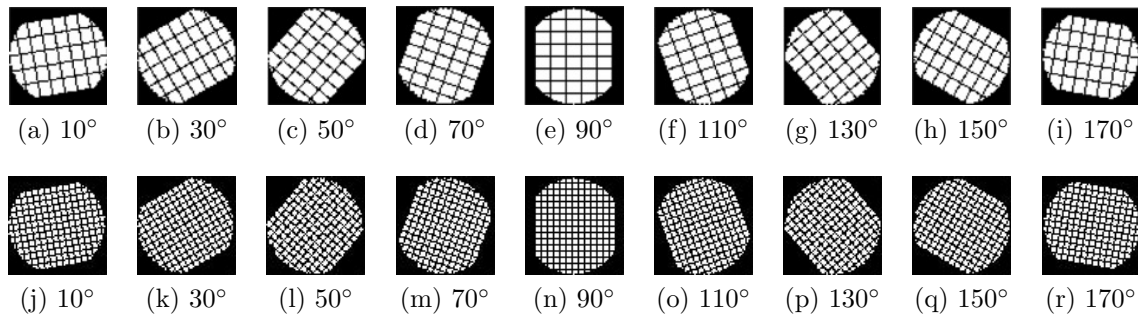


Figure 5.3: Distribution of cells over which descriptors are computed for different orientations: Cells of size 8×10 pixels are shown in (a), (b), (c), (d), (e), (f), (g), (h) and (i); cells of size 4×4 pixels are shown in (j), (k), (l), (m), (n), (o), (p), (q) and (r). Shown images were scaled for viewing.

masks in subfigures (j), (k), (l), (m), (n), (o), (p), (q) and (r) of Figure 5.3 were originated from the rotation of a mask oriented horizontally of size 64×52 pixels and divided into ordered cells of size 4×4 pixels. The corners of each mask were rounded by eliminating pixels that are located at a distance more than 31 pixels from the centre of the detection window. Rounding the corners helps eliminate extra padded parts that are resulted from the rotation process of a given mask. These 18 masks are computed and indices of pixels that belong to each cell are pre-stored saving computational cost.

5.4.1 Second-Order Statistics of Histogram-of-Oriented-Gradient Descriptor

The original implementation of the HOG descriptor was presented in Chapter 4 as well as how to modify HOG in order to be a rotation-invariant image descriptor. In this framework, in a novel fashion, the HOG descriptor is computed over one of the distributions of local cells of size 8×10 pixels, shown in Figure 5.3 according to the estimated orientation of the detection window using the method described in Section 5.3.

The magnitudes of gradients $\mathbf{M}_g$ of local cells are accumulated in nine-bin histograms $\mathbf{g}_{cell}$ according to their orientation $\mathbf{O}_g$ independently without the trilinear interpolation process recommended by [22] as it is a computationally-expensive process and it has a minimal effect on the overall performance. The response for each cell $\mathbf{g}_{cell}$ is normalised using $L2$ -normalisation so that illumination invariance properties can be achieved using Equation 5.2, Equation 5.3, Equation 5.4 and Equation 5.5. The final HOG descriptor is formed from the concatenation of the $L2$ -normalised eigenvalues $d_{k,block}$ of the covariance matrix $\mathbb{R}_{GG,block}$ of the cell responses of a block of four adjacent cells with $\epsilon = 0.01$ rather than the cell responses as in the original implementation because it was found empirically using the eigenvalues of the second order statistics results in a more robust image descriptor. This new modified HOG descriptor has much lower dimensionality of $216D$.

$$\mathbf{d}_{G,block} = [d_{G,1,block}, \dots, d_{G,9,block}]^T = eig(\mathbb{R}_{GG,block}) \quad (5.10)$$

$$\mathbf{d}_{G,block} \leftarrow \frac{\mathbf{d}_{G,block}}{\sqrt{\|\mathbf{d}_{G,block}\|_2^2 + \epsilon^2}} \quad (5.11)$$

5.4.2 Truncated-Pyramid-Colour-Self-Similarity Descriptor and its Second-Order Statistics

Cars exhibit a distinctive distribution of colours, as aforementioned, the colour of windscreens is always dark in aerial images and the colour of the car boot is usually the same as the colour of the car bonnet. Therefore, using a local CSS descriptor will encode the local similarity and variation of colours across pixels. Built on the Colour Self-Similarity descriptors proposed by [83], [84] and its modification in Chapter 4, it is proposed in this framework to improve them by encoding the eigenvalues d_s of the covariance matrix $\mathbb{R}_{SS}$ of the response of a block of adjacent cells using Equation

5.12 and Equation 5.13. This new colour-self-similarity descriptor is based on the similarity among adjacent cells and the second-order statistics.

$$\mathbf{s}_{cell} = [\mathbf{h}_c \cap \mathbf{h}_n] = \left[\left[\sum_k \min[\mathbf{h}_c(k), \mathbf{h}_n(k)] \right] \right], n \in \{1, 2, \dots, 8\} \quad (5.12)$$

$$\mathbf{d}_{S,block} = [d_{S,1,block}, \dots, d_{S,8,block}]^T = eig(\mathbb{R}_{SS,block}) \quad (5.13)$$

where $\mathbf{h}_c$ and $\mathbf{h}_n$ are the two grey-level histograms of the central cell and a neighbouring cell respectively, each has 16 bins, $\mathbf{h}_c(k)$ and $\mathbf{h}_n(k)$ are scores of the k^{th} bins of the two histograms and $\lceil \cdot \rceil$ indicates that the result is truncated at the size of an unsigned integer of eight bits as previously recommended.

Local CSS descriptors are computed on two levels in a pyramid approach. In the first level, descriptors are computed over one of the distributions of local cells of size 8×10 pixels according to the estimated orientation of the detection window and the block size is four cells, whereas in the second level, descriptors are computed over one of the distributions of local cells of size 4×4 pixels according to the estimated orientation of the detection window and the block size is nine cells.

The final modified tPCSS descriptor is formed by concatenating the $L2$ -normalised local responses from all the cells into a single vector using an $\epsilon = 0.01$ and the normalised eigenvalues of the covariance matrices of the colour histograms of each block in the two levels of the pyramid and it has $3190D$.

$$\mathbf{s}_{cell} \leftarrow \frac{\mathbf{s}_{cell}}{\sqrt{\|\mathbf{s}_{cell}\|_2^2 + \epsilon^2}} \quad (5.14)$$

$$\mathbf{d}_{S,block} \leftarrow \frac{\mathbf{d}_{S,block}}{\sqrt{\|\mathbf{d}_{S,block}\|_2^2 + \epsilon^2}} \quad (5.15)$$

5.4.3 Gabor Descriptor and its Second-Order Statistics

Texture is a common image cue that has been used, among other cues, to solve the problem of object detection [86]. Many researchers have used Gabor filters to extract and encode texture motivated by the similarity between 2-D Gabor filters and the receptive fields in the visual cortex of the vision systems of humans [87]. Although cars do not possess distinguishable textured surfaces, the Gabor texture descriptors play a great role in reducing the rate of false positives by discriminating textured surfaces that do not belong to cars, such as those of buildings' roofs and vegetation. Gabor filter kernels are formed by a 2-D Gaussian function modulated by a complex sinusoidal signal:

$$\mathbf{F}_{Gabor}(x, y; \lambda, \theta, \psi, \sigma_g, \gamma) = \exp\left(\frac{x'^2 + \gamma^2 y'^2}{2\sigma_g^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right) \quad (5.16)$$

$$\begin{cases} x' = x \cos \theta + y \sin \theta \\ y' = -x \sin \theta + y \cos \theta \end{cases}$$

Nine Gabor kernels of size 64×64 are used in the ACD framework and they are shown in Figure 5.4, with a wavelength λ of 10 pixels, nine orientations $\theta_k = [10^\circ, 30^\circ, \dots, 170^\circ]$, phase offset ψ of -90° , aspect ratio γ of 10 and the variance of the Gaussian envelop σ_g^2 of 13.671. All values of the previous parameters were found by experimentation by testing excessively the properties of a large number of image samples from the training dataset.

After filtering the detection windows using those nine Gabor kernels, windows are divided into one of the cell distributions of size 8×10 pixels shown in Figure 5.3

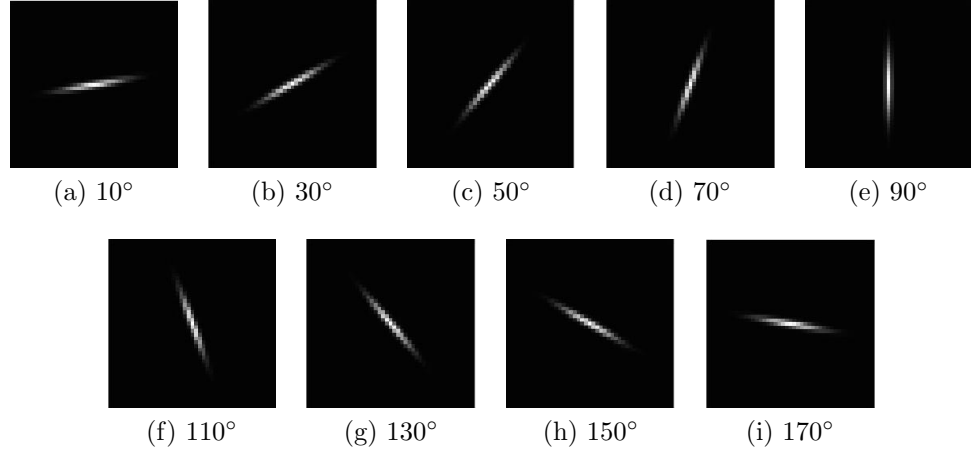


Figure 5.4: Magnitude of the 64×64 Gabor kernels at different orientations with associated wavelength λ of 10 pixels, phase offset ψ of -90° , aspect ratio γ of 10 and the variance of the Gaussian envelop σ_g^2 of 13.671. Shown images were scaled for viewing.

according to the estimated orientation of the detection window. The mean $t_{cell,mean}$ and energy $t_{cell,energy}$ of the response to the Gabor filters in each cell is computed at each orientation.

$$\mathbf{t}_{cell,mean}(k) = \frac{1}{size(cell)} \sum_{x,y} (|\mathbf{F}_{Gabor} * \mathbf{I}|) \mathbf{1} [\theta_k = (20k - 10)^\circ] \quad (5.17)$$

$$\mathbf{t}_{cell,energy}(k) = \frac{1}{size(cell)} \sum_{x,y} ((\mathbf{F}_{Gabor} * \mathbf{I})(\mathbf{F}_{Gabor} * \mathbf{I})^H) \mathbf{1} [\theta_k = (20k - 10)^\circ] \quad (5.18)$$

$$\mathbf{t}_{cell} = [\mathbf{t}_{cell,mean}, \mathbf{t}_{cell,energy}]^T \quad (5.19)$$

For rotation-invariance, the order of the nine orientations is changed such that rotation invariance is achieved following the same idea of circularly-shifting the histogram bins in Chapter 4. The final Gabor descriptor is formed from the

concatenation of the $L2$ -normalised eigenvalues of the covariance matrix of the cell responses of a block of four adjacent cells using an $\epsilon = 0.01$ and it has $432D$.

$$\mathbf{d}_{T,block} = [d_{T,1,block}, \dots, d_{T,18,block}]^T = eig(\mathbb{R}_{TT,block}) \quad (5.20)$$

$$\mathbf{d}_{T,block} \leftarrow \frac{\mathbf{d}_{T,block}}{\sqrt{\|\mathbf{d}_{T,block}\|_2^2 + \epsilon^2}} \quad (5.21)$$

5.5 Post-Processing

Since an overlapping sliding detection window has been adopted in our framework, multiple detections per one single true detection are expected. Bounding boxes that overlap by more than 50% using Equation 3.1 are eliminated except the one that has the highest output confidence score [24].

5.6 Datasets and Classification

The proposed hybrid feature vector is composed by the concatenation of the novel ensemble of descriptors: 216 HOG features, 3190 PCSS features and 432 Gabor features including the eigenvalues of the covariance matrices of each descriptor. The overall length of the feature vector is $3838D$. Although the new proposed feature vector has lower dimensionality than that of the FCD in Chapter 4 by $1158D$, the classification problem is still regarded as a relatively high-dimensional classification problem. Accordingly, an SVM classifier that is known to work efficiently in high-dimensional scenarios [39] has been adopted to discriminate the car subspace from the non-car subspace with a regularisation parameter c equals 15, which was chosen using cross validation on the training dataset. A linear classifier has been used to

show the robustness of the new proposed feature vector.

In order to classify a new data sample s , the feature vector is computed using the transformation $\phi(s)$ that transforms this new data sample to a hyperspace of dimensionality $3838D$. The distance from the separating hyperplane is calculated using the optimal weight $\mathbf{w}_{opt}$ and bias b using Equation 5.22. The sign of $z(s)$ determines on which side of the hyperplane the data sample s is and the distance from the hyperplane represents its confidence score. A detailed mathematical analysis of how SVM classifier works is presented in appendix A.

$$z(s) = \mathbf{w}_{opt}^T \phi(s) + b \quad (5.22)$$

The classifier was trained on samples extracted from the training areas of the Vaihingen dataset. Sample examples are shown in Figure 3.1. For positive training data, square patches of pixels that contain cars were chosen from areas other than the test areas so that image descriptors can be computed efficiently over cars and a small part of the surrounding background. Patches were chosen from different regions such that a great diversity in the orientation of cars and backgrounds could be achieved. Mirrored and horizontally-flipped versions of these square patches were included as well to richen the diversity of the training data. For negative training data, randomly selected patches from different areas of the background as well as bootstrapped hard negatives that were detected as false positives during the validation process on areas other than test areas were included. In total, 781 positive samples and 2473 negative samples were used in the training processing.

5.7 Results

The new proposed ACD framework was run on the test areas of the Vaihingen dataset that were described in Section 3.1.1 and such data was not included in the data over which the SVM classifier was trained. Furthermore, it was run over the OIRDS dataset that was described in Section 3.1.1 and it was a completely unseen set of data to the classifier to test the ability of the ACD framework to generalise. The performance was assessed and evaluated using qualitative analysis by the visual inspection of the bounding boxes that were produced as a result of applying the proposed framework on the data and various quantitative measures, namely, the precision-recall, the mean average precision, recall rates at different false-positive-per-image rates and the interpolated average precision that were described in Section 3.2. Obtained results are the outcome of a single experiment conducted on each of the two datasets.

5.7.1 Qualitative Results

Visual results of the experiments of applying the ACD framework on sample regions of the test areas of the Vaihingen dataset are shown in Figure 5.5, whereas results on sample images of the OIRDS datasets are shown in Figure 5.6. These two figures show that the ACD framework is able to robustly detect cars in both environments.

In Figure 5.5 (c), it can be seen that the ACD method is able to correctly classify detection windows that have poor contrast (fourth car from top), thanks to the local normalisation of the HOG responses of cells, discussed in Section 5.4.1, a case which is even hard for the human eye. Nevertheless, in very few situations, wherein detection windows contain black cars that are covered by very strong building shadows and the contrast is too low, the proposed method fails to correctly classify them as true positives, because the strength of gradients in such detection

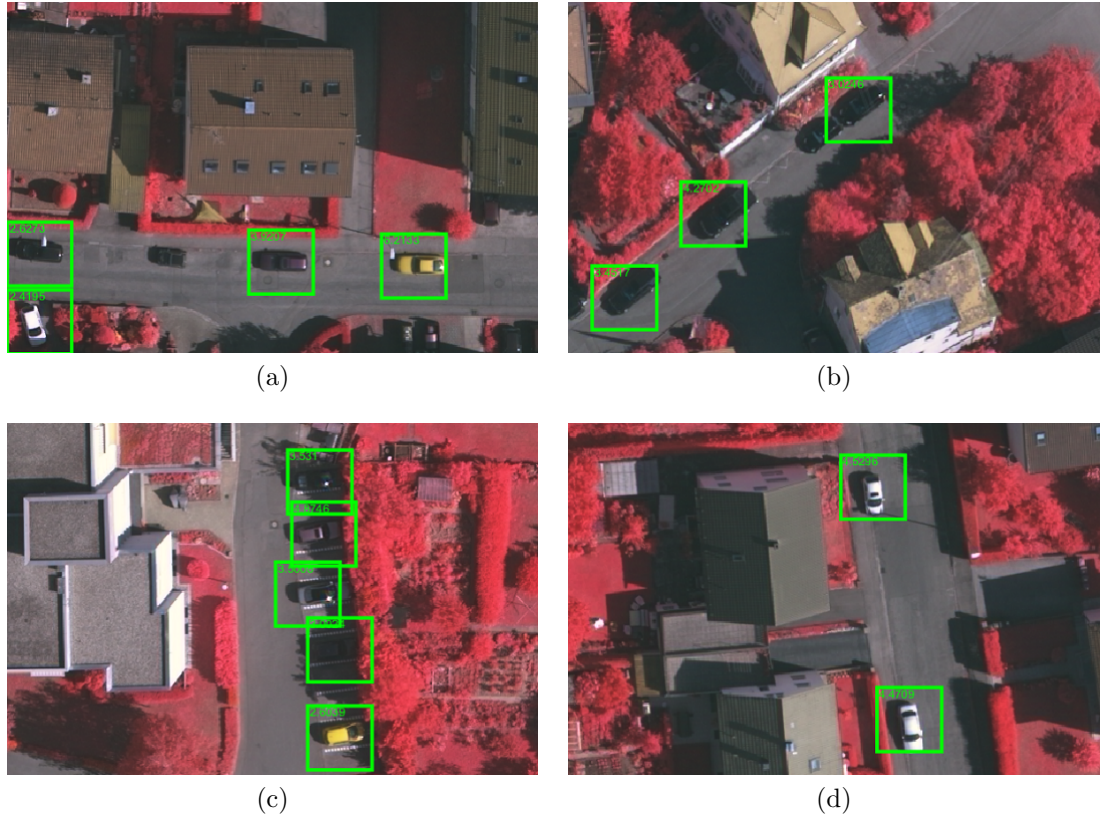


Figure 5.5: Sample regions of the test areas of the Vaihingen dataset after applying the ACD framework. Bounding boxes represent detections and the number at the top left corner of each bounding box represents its confidence score. Shown images were scaled for viewing.

windows is too low and the colour distribution is concentrated around the histogram bins that correspond to dark grey levels. In Figures 5.5 (a) and (b), two cars were not detected as they have a smaller size (mini cars) compared to other standard cars. However, the number of these cars in the whole test areas is too small (five cars) and this problem can be solved by extending the proposed method to multi-scale analysis, which is among the future research goals as it will be discussed in Chapter 7.

From Figure 5.6, it can be clearly deduced that the ACD framework has a very good generalisation as it achieves high true-positive rate, while having a low false-positive rate, when applied to images with different acquisition conditions and a lower resolution than those of the data on which the SVM classifier was trained.

More specifically, in challenging cases, where cars are partially over-imposed by shadows, e.g., in Figure 5.6 (b) and Figure 5.6 (e), the ACD framework is able to detect cars despite the fact that the shadow alters the distribution of edges and colours. Moreover, in the case of complex backgrounds, for example in Figure 5.6 (d) and Figure 5.6 (i), the proposed ACD algorithm did not produce a high rate of false positives.

Furthermore, in Figure 5.5 and Figure 5.6, it can be noted that black cars tend to be detected with confidence scores of similar values to those of cars with bright colours indicating that the ACD framework has detection consistency. This constitutes an advantage of the ACD framework over the FCD framework.

5.7.2 Quantitative Results

Performance of the ACD Framework

Figure 5.7 shows the obtained precision-recall curves for the ACD framework, when applied to the three test areas of the Vaihingen dataset. It can be seen that it has the best performance, represented by a high precision rate of 73.81% at a recall rate of 79.49%, in urban regions that contain small detached houses (area 3). Nonetheless, the performance degrades to a precision rate of 62.96% at a recall rate of 69.39% in urban regions that contain high-rising residential buildings (area 2) because in these areas edge maps and colour distributions of the different parts of buildings are very similar to that of cars as previously discussed. The overall precision and recall rates in the whole testing areas are 68.57% and 75% respectively.

Figure 5.8 shows the obtained precision-recall curve for the ACD framework, when applied to the OIRDS dataset. The overall precision and recall rates on the whole dataset are 40.04% and 77.77% respectively. These numbers indicate the acceptable performance of the ACD framework, when tested on a large number of



Figure 5.6: Sample regions of the test images of the OIRDS dataset after applying the ACD framework. Bounding boxes represent detections and the number at the top left corner of each bounding box represents its confidence score. Shown images are scaled for viewing.

images containing hundreds of cars. Table 5.2 shows the mean average precision values of the obtained results of the ACD framework. They also indicate the adequate performance on both datasets.

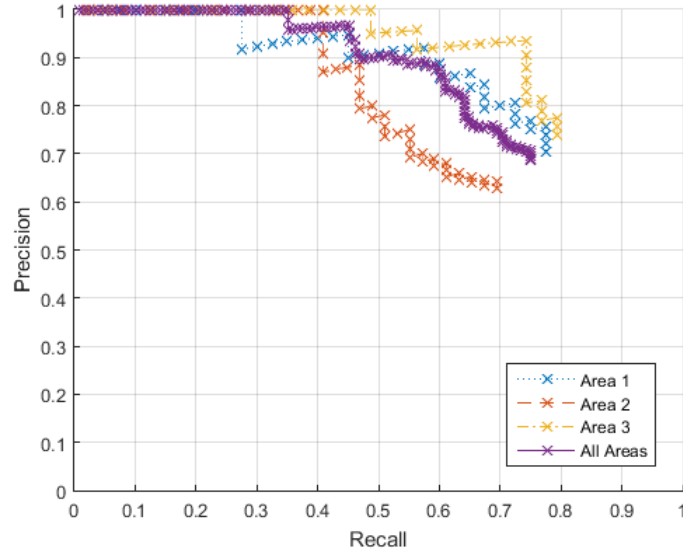


Figure 5.7: Performance of the ACD framework in different test areas of the Vaihingen dataset represented by precision rates at different recall rates.

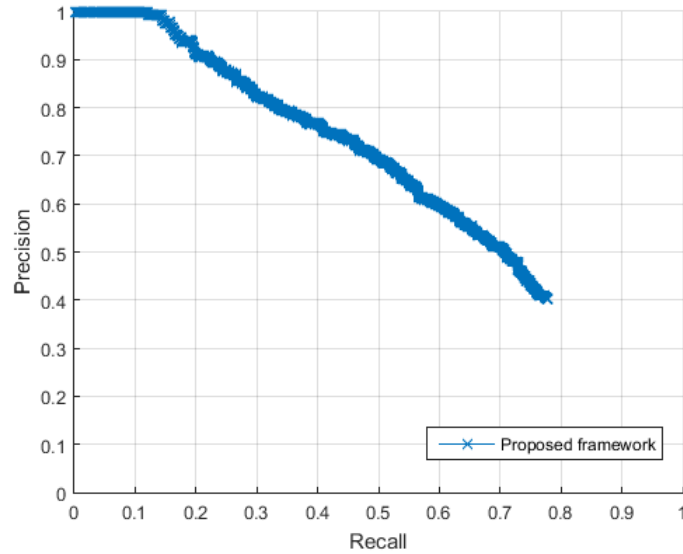


Figure 5.8: Performance of the ACD framework on the OIRDS dataset represented by precision rates at different recall rates.

Accuracy of the Estimation of the Orientation of Cars

The ACD framework uses the eigenvalue of the covariance matrix of the cell responses to HOG to estimate the orientation of the dominant edge as previously

Table 5.2: Performance of the ACD framework on the Vaihingen and OIRDS datasets represented by the Mean Average Precision MAP and recall RC rates at different false positive per image $FPPI$ values.

Dataset	MAP	RC		
		$0.01FPPI$	$0.1FPPI$	$1.0FPPI$
Vaihingen	92.79%	35.16%	60.16%	75.00%
OIRDS	76.1%	16.56%	32.84%	72.84%

Table 5.3: Performance of the FCD and ACD frameworks on the Vaihingen dataset represented by the Mean Average Precision MAP and recall RC rates at different false positive per image $FPPI$ values.

Framework	Dataset	MAP	RC		
			$0.01FPPI$	$0.1FPPI$	$1.0FPPI$
FCD	Vaihingen	75.10%	34.55%	49.09%	77.27%
ACD	Vaihingen	92.79%	35.16%	60.16%	75.00%
FCD	OIRDS	70.41%	12.84%	29.95%	65.30%
ACD	OIRDS	76.1%	16.56%	32.84%	72.84%

discussed in Section 5.3. The accuracy of estimating this dominant edge orientation is 98.34%.

Comparison between FCD and ACD frameworks

Figure 5.9 and Figure 5.10 show a comparison between the performance of the FCD and ACD frameworks. It can be deduced that the ACD framework outperforms the FCD framework on both datasets. It has a higher precision for a given recall rate. Furthermore, it has a higher overall MAP as indicated in Table 5.3. The recall rate of the ACD framework at 1.0 $FPPI$ is slightly lower than that of the FCD framework, as it can be noticed. The reason for this is that the ACD framework is not able to detect small cars (mini cars) as previously mentioned. The ACD framework is more robust and has a better generalisation than the FCD framework, when it was tested on the OIRDS dataset.

Moreover, the ACD framework achieves higher accuracy than the FCD framework in estimating the orientation of a given car by $\approx 8\%$.

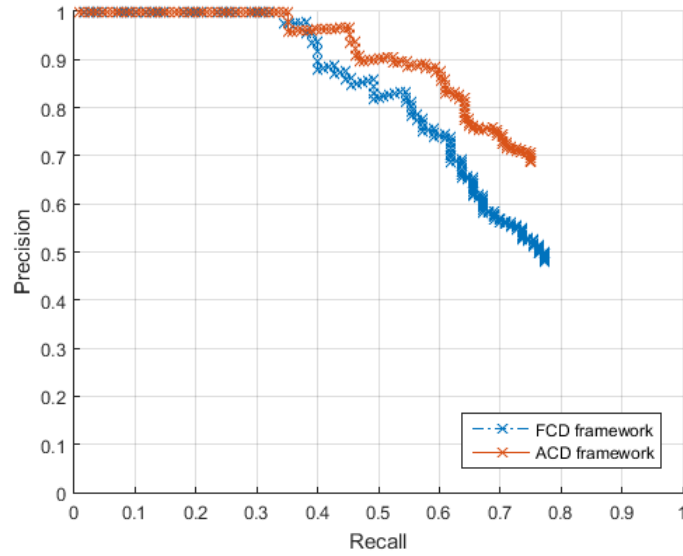


Figure 5.9: Performance of the FCD and ACD frameworks on the Vaihingen dataset represented by precision rates at different recall rates.

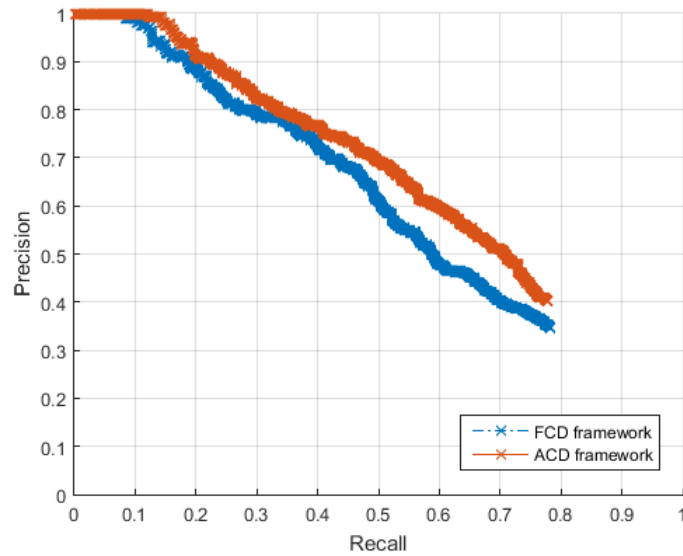


Figure 5.10: Performance of the FCD and ACD frameworks on the OIRDS dataset represented by precision rates at different recall rates.

Comparison with the State of the Art

Figure 5.11 shows the obtained precision-recall curves for the ACD framework, FCD framework and the framework of Shao *et al.* [56]. It can be deduced from these

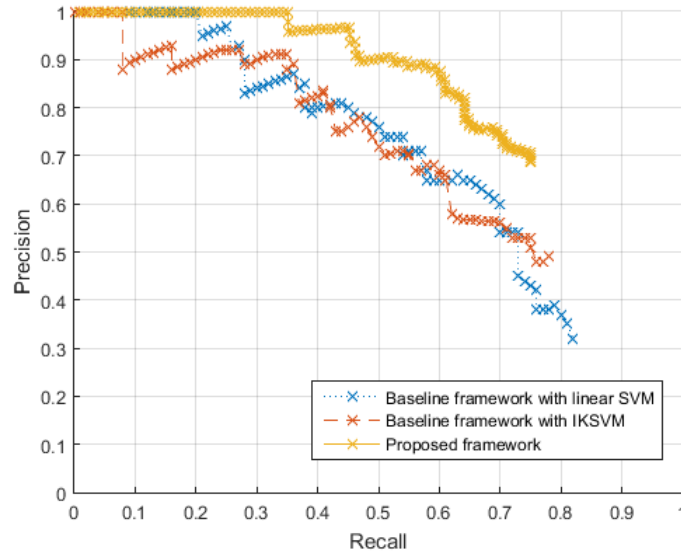


Figure 5.11: Performance of the ACD framework and the baseline framework of [56] on the Vaihingen dataset represented by precision rates at different recall rates.

curves that for the same recall the ACD framework achieves the highest precision using linear classification unlike the standard implementation of Shao *et al.* [56] that adopts a non-linear classifier. Although the framework of Shao *et al.* [56] achieves higher recall rate in the case of using a linear classifier, its associated precision is less than 40% indicating a very high rate of false positives. Moreover, the ACD algorithm has a precision of 100% for up to a recall rate of 35% and it has lower dimensional feature vector than that of Shao *et al.* [56] by 2922D.

Furthermore, the performance of the ACD framework was compared to the framework of Shao *et al.* [56], to the framework of Razakarivony and Jurie [58] and the framework of Madhogaria *et al.* [60] using the 11-point average precision AP , described in Section 3.2 and suggested by [24], and results are shown in Table 6.2. It achieves an overall 11-point average precision, higher than that of Shao *et al.* [56] on the Vaihingen dataset and higher than that of Razakarivony and Jurie [58] and of Madhogaria *et al.* [60] on the OIRDS dataset.

Obtained results show clearly that the ACD algorithm outperforms the state

Table 5.4: Performance of the ACD framework and baseline frameworks represented by the 11-point Average Precision AP .

Framework	Dataset	AP
ACD	Vaihingen	68.03%
FCD	Vaihingen	64.26%
Shao <i>et al.</i> [56]	Vaihingen	58.92%
ACD	OIRDS	57.34%
FCD	OIRDS	53.55%
Razakarivony and Jurie [58]	OIRDS	48.9%
Madhogaria <i>et al.</i> [60]	OIRDS	54.45%

of the art by achieving higher AP on the Vaihingen dataset using a linear classifier as opposed to the non-linear classifier used in [56]. It also outperforms the car-detection framework developed in [58] that uses two models and a 10-fold training/testing methodology on the OIRDS dataset. In addition, it is superior to the framework of Madhogaria *et al.* [60], when they were both tested on the challenging OIRDS dataset, despite the fact that the ACD used a smaller training dataset.

As previously mentioned due to the unavailability of the source code of the baseline frameworks, it was not possible to evaluate their performance on both datasets.

The ACD framework has been implemented in a MATLAB environment. All of the experiments were run on a computer with the following specifications: a 2.5-GHz-speed Intel-Core-i5 CPU and a 6-GB Random Access Memory. The machine that was used to conduct the experiments has multiple processing cores. Nevertheless, the current implementation uses only a single core. The current implementation has a near-real-time performance as it is able to process ≈ 421 detection windows per second. Thanks to the novel method of adaptive cell distribution that overcomes the associated high computational cost of rotating detection windows, the ACD framework is faster than the FCD framework by ≈ 5.5 times.

5.8 Summary

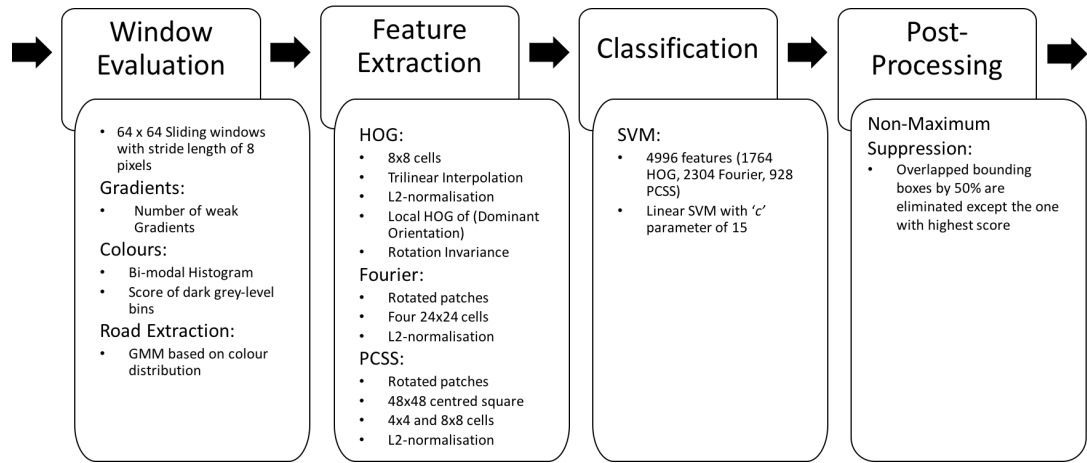


Figure 5.12: Summary of the ACD framework for the detection of cars in airborne imagery using adaptive cell distributions.

This chapter introduced the Adaptive-Cell-Distribution framework for the detection of cars in aerial images of urban scenes. Implementation details of the four main stages were given and discussed and a summary is given in Figure 5.12. Image descriptors that are used in the Adaptive-Cell-Distribution framework are: modified histogram of oriented gradients, pyramid colour self-similarity and Gabor and the eigenvalue of the covariance matrices of their local distributions. The Adaptive-Cell-Distribution framework outperforms baseline frameworks as well as the previously proposed Fixed-Cell-Distribution framework since it achieves higher average precision rates of 68.03% and 57.34% on the Vaihingen and OIRDS datasets, respectively.

Chapter 6

Detection of Cars in Road Areas

IN urban areas, cars are mostly found in road areas and areas paved with asphalt. Based on this, a novel method to reduce the false-positive rate of the FCD and ACD frameworks by eliminating detection windows that have low confidence scores and that are not placed around road areas is introduced. The proposed method is based on the use of a Gaussian Mixture Model (GMM) classifier that has been trained on colour features to divide the colour space into multiple subspaces in order to eliminate such detection windows. This new method will be incorporated in the window-evaluation and post-processing stages of the FCD and ACD frameworks. The overall performance of the resulted integrated frameworks, the FCD-GMM and the ACD-GMM, will be investigated over the Vaihingen dataset.

The main contributions of the work that will be presented in this chapter include the introduction of a novel method to extract regions that are very likely to belong to road areas using a Gaussian Mixture Model that divides the colour space and exploiting it for car detection. The strength of the proposed framework include: (i) the robust and accurate detection of cars in urban scenes with high precision and very low false-positive rates, (ii) no prior knowledge about the scene under examination is required and (iii) no extra hardware is needed as compared

to the previously-developed methods that used road maps, which are found in the open literature.

This chapter is organised as follows: in Section 6.1, an overview of the FCD-GMM and ACD-GMM frameworks is presented followed by the illustration of implementation details in Section 6.2. Obtained results of applying the integrated FCD-GMM and ACD-GMM frameworks as well as comparing the performance of the two combined frameworks are given in Section 6.4. A summary of the proposed FCD-GMM and ACD-GMM frameworks is given in Section 6.5.

6.1 Overview of the Proposed FCD-GMM and ACD-GMM Frameworks

In Chapters 4 and 5, it was shown that the use of a well-engineered ensemble of image descriptors results in a more robust and accurate system for the detection of particular targets of interest, such as cars. Nevertheless, in heavily-cluttered environments that contain many objects of various shapes other than the target of interest, target-detection systems may claim very hard negatives as true positives since these non-car objects or parts of them form together a very similar visual appearance to that of cars in terms of the structural and radiometric properties. For example, in Figure 6.1, both FCD and ACD frameworks produced some false positives, when they were applied to sample test areas of the Vaihingen dataset. These false positives have associated relatively-low confidence scores and accordingly they can be eliminated by setting a higher threshold for the confidence score. However, there are some cars that are detected with similar confidence score values, such as black cars. These false positives are called hard negatives. Moreover, some of them are even hard for the human eye to distinguish them from cars.

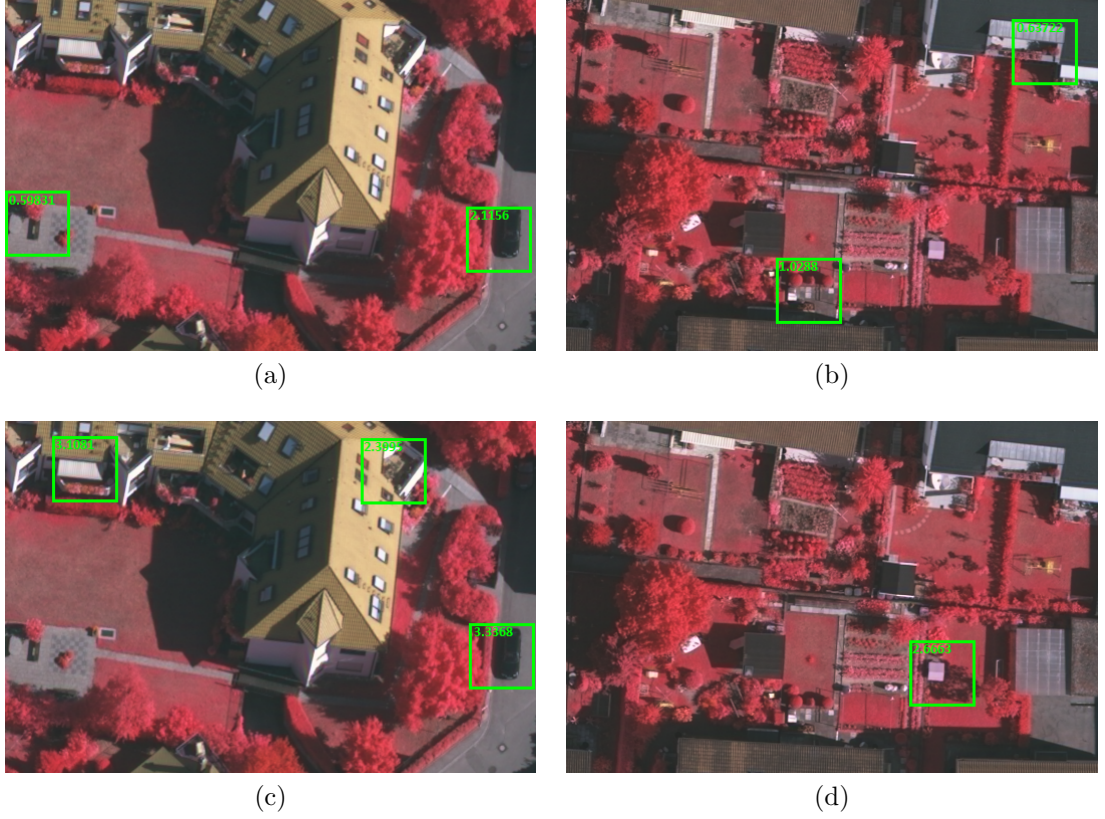


Figure 6.1: Hard negatives that were claimed by the FCD and ACD car-detection frameworks as true positives: (a) and (b) outputs of the FCD framework on a sample region of test area 1 and test area 3 of the Vaihingen dataset, respectively; (c) and (d) outputs of the ACD framework on a sample region of test area 1 and test area 3 of the Vaihingen dataset, respectively. Bounding boxes represent detections and the number at the top left corner of each bounding box represents its confidence score. Shown images were scaled for viewing.

A possible solution to tackle this problem of the existence of high amount of clutter without involving more computationally-expensive image descriptors is to extract regions of interest and eliminate detection windows that are not placed around these regions. As the target of interest in this work is cars, regions of interest (ROI) are thought to be roads and parking areas. Many researchers, as previously discussed in Chapter 2, used road maps and a map-projection method to extract these regions. However, this requires the knowledge of road maps *a priori* and a computationally-expensive method to project this map onto the image under examination.

It is proposed in this chapter to incorporate distinctive colour properties that belong to road areas by the use of a GMM classifier that divides the colour space into several subspaces represented by seven Gaussian components. Depending on the contribution of the pixel intensity to the three colour channels, a pixel is classified corresponding to road areas if it belongs to the Gaussian components that correspond to the colour subspace of roads. The number of Gaussian components was chosen based on the colour distribution of the training data. A binary mask that indicates whether this given pixel corresponds to the colour subspace of roads will be then created. The elimination process is performed as follows: (i) detection windows that do not overlap with areas, wherein the binary mask has an "on" response, by more than 20% are discarded, (ii) detection windows that have an overlap more than 20% and have high-confidence scores are kept, (iii) detection windows with low confidence scores and have an overlap more than 30% are considered as true detections and (iv) detection windows with low confidence scores and that do not overlap with ROIs are discarded. Here, the confidence score is used to resemble the distance from the separating hyperplane. The first condition is integrated with the window-evaluation stage, whereas the other three conditions are integrated with the post-processing stage. The reason for this is that the first condition is a general condition for all detection windows regardless of the associated confidence score. Consequently, it can be checked prior to the extraction-and-encoding-of-features stage, saving processing time.

This process of using a GMM classifier does not have a high computational cost as a given pixel is assigned to one of the Gaussian components according to largest posterior probability for this pixel intensity weighted by the component probability. Implementation details of this stage of the extraction of ROIs and how the GMM classifier was trained are given in the following sections.

6.2 Extraction of Road Areas

In urban scenes, the majority of cars are expected to be found in roads and parking areas. As discussed before, extracting these regions would eliminate a large number of false positives. Colour features have been exploited to extract these regions based on the fact that almost all roads are paved with asphalt that possess a very distinctive dark grey colour. An unsupervised GMM classifier with seven components was trained on the training dataset, which will be described in the following section. The probability density function pdf of the feature vector $\mathbf{c}$ can be modelled as a mixture of Gaussian components. This GMM can be defined as a weighted sum of multiple Gaussian components and can be expressed according to Equation 6.1 and Equation 6.2 [39].

$$pdf(\mathbf{c}) = \sum_k \pi_n \mathcal{N}(\mathbf{c} | \mu_{\mathbf{c},n}, \mathbb{R}_{\mathbf{c},n}), n \in \{1, 2, \dots, 7\} \quad (6.1)$$

where $\mathbf{c}$ is a 3-D feature vector, wherein each feature corresponds to the pixel intensity of each colour channel in the RGB colour representation, $\mathcal{N}(\mathbf{c} | \mu_{\mathbf{c},n}, \mathbb{R}_{\mathbf{c},n})$ is one Gaussian component with mean $\mu_{\mathbf{c},n}$ and covariance $\mathbb{R}_{\mathbf{c},n}$ and π_n is the mixing coefficient. A Gaussian component $\mathcal{N}(\mathbf{c} | \mu_{\mathbf{c},n}, \mathbb{R}_{\mathbf{c},n})$ is a multivariate Gaussian function and can be defined as:

$$\mathcal{N}(\mathbf{c} | \mu_{\mathbf{c},n}, \mathbb{R}_{\mathbf{c},n}) = \frac{1}{(2\pi)^{D/2} |\mathbb{R}_{\mathbf{c},n}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{c} - \mu_{\mathbf{c},n})^T \mathbb{R}_{\mathbf{c},n}^{-1} (\mathbf{c} - \mu_{\mathbf{c},n}) \right\} \quad (6.2)$$

Parameters $\mu_{\mathbf{c},n}$ and $\mathbb{R}_{\mathbf{c},n}$ are estimated from the training dataset using the iterative expectation-maximisation (EM) algorithm. Analysis about the mathematical theory of how the GMM classifier works is given in Appendix B.

6.3 Datasets and Classification

The proposed colour feature vector is formed by the concatenation of pixel intensities of the three colour channels in a vector format. The GMM classifier was trained on the RGB colour features of all the pixels of patches extracted from the training areas of the Vaihingen dataset. Sample examples are shown in Figure 3.1. A total of 4335 patches were used for training. They included randomly-selected patches that were placed around cars, road areas and various backgrounds. Since the GMM classifier is an unsupervised classifier, all colour features of the training patches were fed to the classifier together without their associated labels.

A detection window is claimed to belong to road areas according to the conditions that were described in Section 6.1. This process is integrated mostly with the post-processing stage in both the FCD and ACD frameworks.

6.4 Results

The proposed FCD-GMM and ACD-GMM frameworks were run on the test areas of the Vaihingen dataset that were described in Section 3.1.1. No further training was carried out. SVM classifiers that were trained for the original FCD and ACD frameworks were used. The performance was assessed and evaluated using qualitative analysis by the visual inspection of the bounding boxes that were produced as a result of applying the proposed framework on the data and various quantitative measures, namely, the precision-recall, the mean average precision, recall rates at different false-positive-per-image rates and the interpolated average precision that were described in Section 3.2.

6.4.1 Qualitative Results

Visual results of the experiments of applying the proposed FCD-GMM and ACD-GMM frameworks on sample regions of the test areas of the Vaihingen dataset are shown in Figure 6.2. Extraction of region of regions of interest is approximate, as can be inferred from Figure 6.2. It can be noticed that areas over-imposed by shadows have been extracted as they possess the same colour distribution in road and non-road areas. Nevertheless, it is good enough to reduce the false positive rate. This will be corroborated in the quantitative analysis section. In comparison to the output of FCD and ACD frameworks that were shown in Figure 6.1, it can be observed that the false positives have been successfully eliminated as these false positives were claimed in regions of the urban scene under examination that do not belong to road areas. Consequently, these false-positive detection windows did not overlap enough with the created binary mask and they were successfully discarded using the new FCD-GMM and ACD-GMM frameworks.

6.4.2 Quantitative Results

Figure 6.3 shows the obtained precision-recall curves for the FCD-GMM and ACD-GMM frameworks, when applied to the three test areas. It can be observed that both frameworks have shown the best performance on area 3 as expected since both the FCD and ACD frameworks achieved the best precision in this area. The overall precision and recall rates in the whole testing areas for the FCD-GMM framework are 75.22% and 77.27% respectively, whereas the overall precision and recall rates in the whole testing areas for the ACD-GMM framework are 89.52% and 74.02% respectively.

Moreover, it can be noted from Figure 6.3 that the recall rates are slightly lower than those of the FCD and ACD frameworks. This has occurred because there

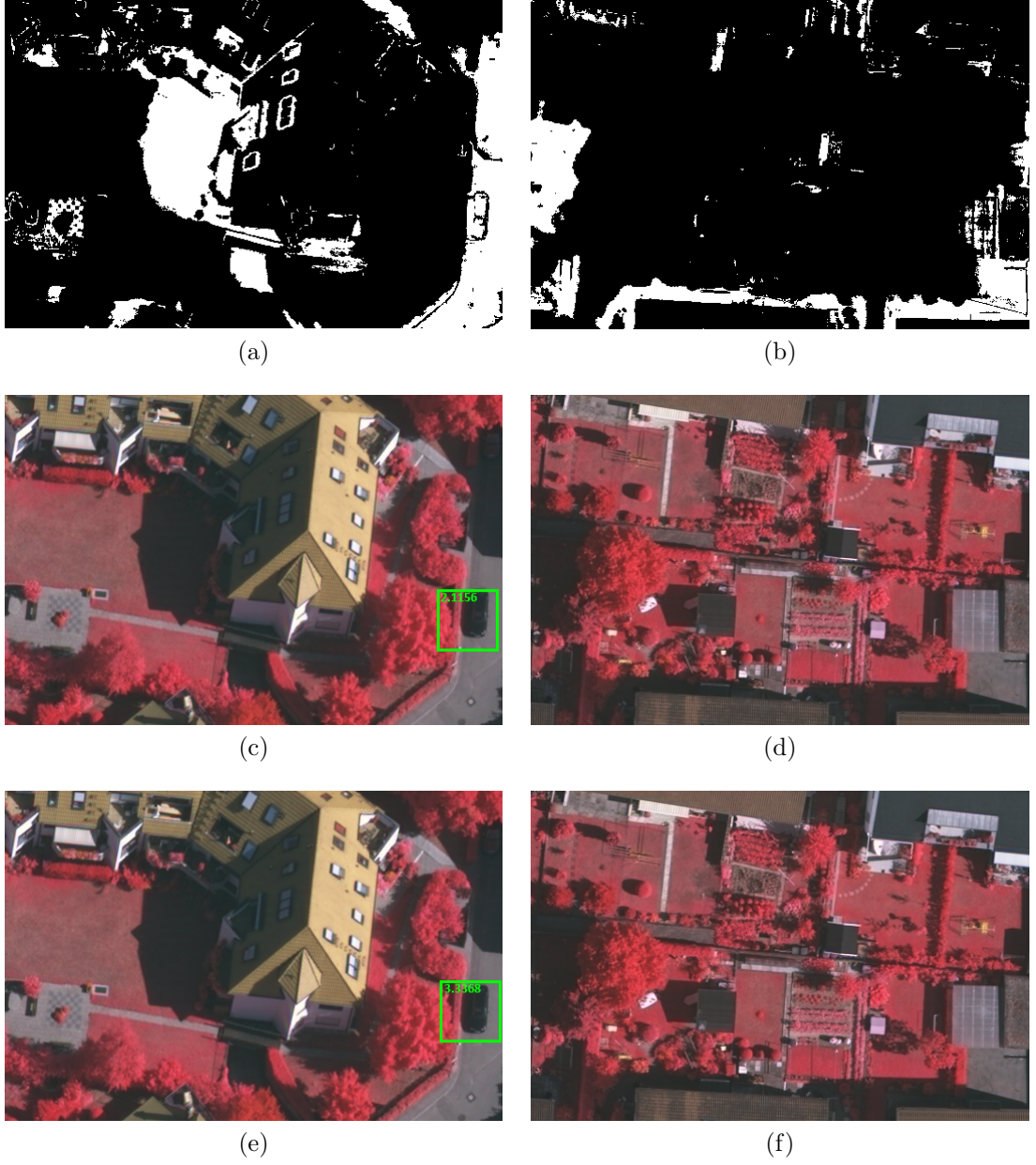
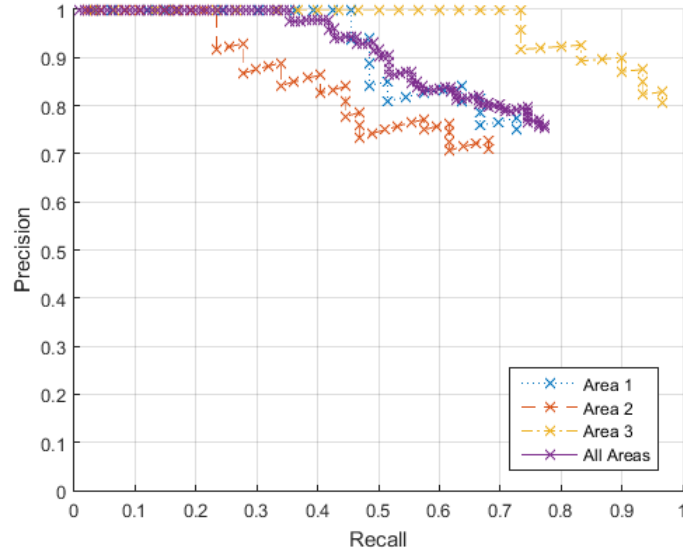
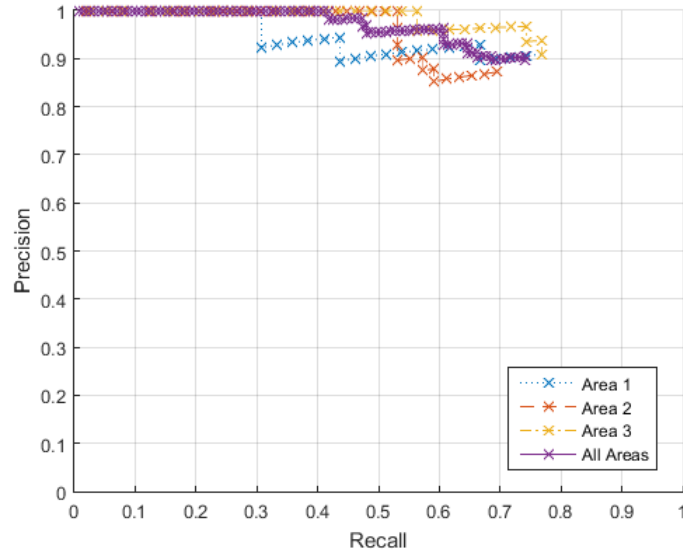


Figure 6.2: Sample regions of the test areas of the Vaihingen dataset after applying the proposed frameworks: (a) and (b) outputs of the GMM classifier; (c) and (d) outputs of the FCD-GMM framework on a sample region of test area 1 and test area 3 of the Vaihingen dataset; (e) and (f) outputs of the ACD-GMM framework on a sample region of test area 1 and test area 3 of the Vaihingen dataset. Bounding boxes represent detections and the number at the top left corner of each bounding box represents its confidence score. No false positives were produced as compared to the output in Figure 6.1. Shown images were scaled for viewing.

is a very small number of cars in the Vaihingen dataset that are not placed in road areas.



(a)



(b)

Figure 6.3: Performance of the proposed integrated frameworks in different test areas of the Vaihingen dataset represented by precision rates at different recall rates: (a) precision-recall curve for the FCD-GMM framework; (b) precision-recall curve for the ACD-GMM framework.

Figure 6.4 shows a comparison between the obtained precision rates for a given recall rate for FCD-GMM and ACD-GMM frameworks, when applied to the whole test areas of the Vaihingen dataset. It can be deduced that the ACD-GMM has a higher precision for detections with low confidence score than the FCD-GMM

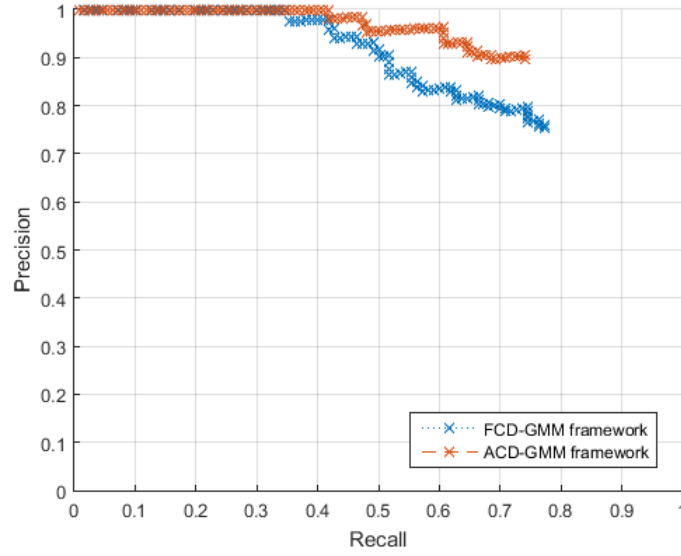


Figure 6.4: Performance of the proposed FCD-GMM and ACD-GMM frameworks in the three test areas of the Vaihingen dataset represented by precision rates at different recall rates.

Table 6.1: Performance of the proposed frameworks on the Vaihingen dataset represented by the Mean Average Precision MAP and recall RC rates at different false positive per image $FPPI$ values.

Framework	Dataset	MAP	RC		
			$0.01FPPI$	$0.1FPPI$	$1.0FPPI$
FCD	Vaihingen	75.10%	34.55%	49.09%	77.27%
ACD	Vaihingen	92.79%	35.16%	60.16%	75.00%
FCD-GMM	Vaihingen	92.95%	35.45%	56.36%	77.27%
ACD-GMM	Vaihingen	96.00%	41.73%	74.02%	74.02%

framework. However, it has a slightly higher rate of false negatives as it achieves slightly lower recall rate. The reason for this is that the original FCD framework is able to recall a slightly-larger number of cars from the ground truth than the ACD framework can.

Table 6.1 shows the mean average precision values of the obtained results of the FCD, ACD, FCD-GMM and ACD-GMM frameworks at different false-positive-per-image rates, when applied to the whole test areas of the Vaihingen dataset. Clearly, the ACD-GMM framework achieves the highest overall mean average precision on the whole dataset. This promotes its superiority among other frameworks.

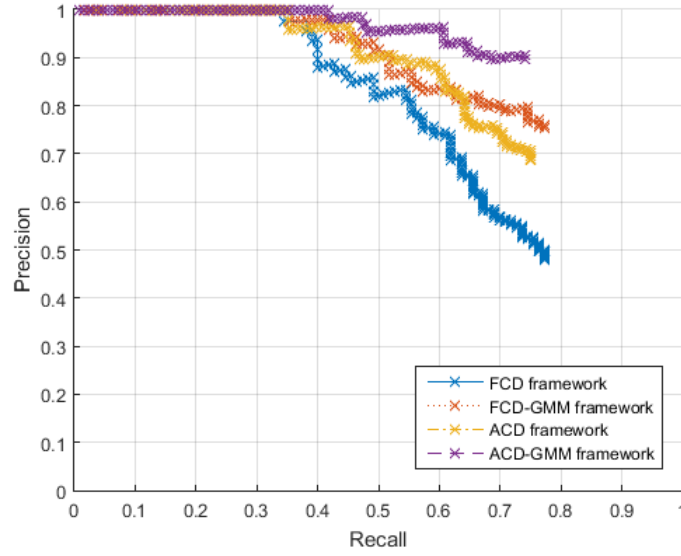


Figure 6.5: Performance comparison of the proposed frameworks with and without extracting regions of interest in the three test areas of the Vaihingen dataset represented by precision rates at different recall rates.

Figure 6.5 shows the obtained precision-recall curves for the FCD, FCD-GMM, ACD and ACD-GMM frameworks. Obviously, using a GMM classifier to extract regions of interest and eliminate detection windows that are not placed around road areas results in a higher precision for the same recall. Moreover, it can be realised that the FCD-GMM framework has the same recall rate as the FCD framework indicating that the added GMM classifier did not affect the recall rate. However, the ACD-GMM framework has slightly lower recall rate than that of the original ACD framework. The reason for this is that the ACD framework is able to detect cars that are found parked between gardens accurately, unlike the FCD framework. Nevertheless, such areas are not paved with asphalt and consequently, detection windows that were placed around them were eliminated.

Finally, the performances of the four proposed frameworks and the baseline method of [56] are compared using the 11-point interpolated average precision and results are shown in Table in 6.2. The ACD-GMM has the highest average precision.

Table 6.2: Performance of the proposed frameworks and the baseline framework of [56] represented by the 11-point Average Precision AP .

Framework	Dataset	AP
FCD	Vaihingen	64.26%
ACD	Vaihingen	68.03%
FCD-GMM	Vaihingen	68.47%
ACD-GMM	Vaihingen	69.70%
Shao <i>et al.</i> [56]	Vaihingen	58.92%

The performance of the FCD-GMM and ACD-GMM frameworks was not evaluated on the OIRDS dataset as it contains many cars that are not placed in roads and parking areas. This constitutes a limitation of the current implementation of both the FCD-GMM and the ACD-GMM frameworks.

The FCD-GMM and the ACD-GMM frameworks have been both implemented in a MATLAB environment. All of the experiments were run on the same computer that was used for testing the original FCD and ACD frameworks. It has the following specifications: a 2.5-GHz-speed Intel-Core-i5 CPU and a 6-GB Random Access Memory. The current implementation exploits only one processing core. The FCD-GMM is able to process ≈ 81 detection windows per second, whereas the ACD-GMM is able to process ≈ 488 detection windows per second. Both frameworks have near real-time performance. The ACD-GMM is faster to run than the FCD-GMM by ≈ 6.025 times.

Compared to the original implementation of the FCD and ACD frameworks, adding the GMM classifier stage helps reduce the processing time by $\approx 6\% - 15\%$. The effect of this is more obvious on the ACD-GMM framework than the FCD-GMM framework as the window-evaluation stage in the former does not depend on the distribution of colours, unlike that of the later, which takes into account the distribution of colours.

6.5 Summary

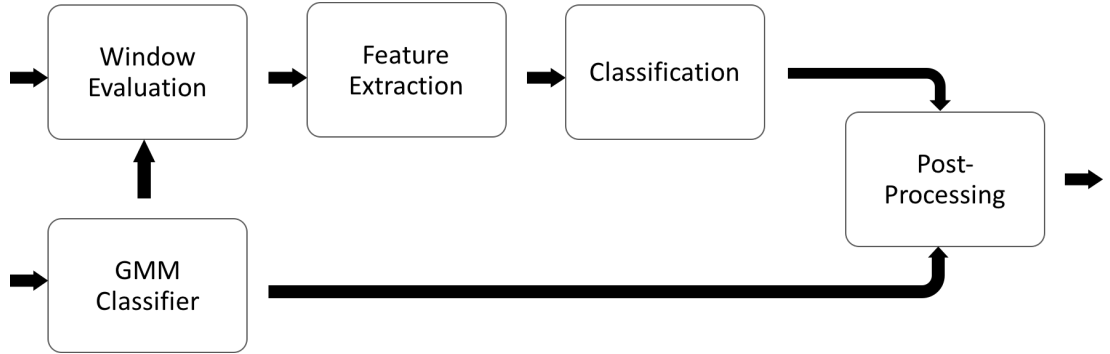


Figure 6.6: Block diagram of the proposed FCD-GMM and ACD-GMM frameworks for the detection of cars in airborne imagery using automatic extraction of roads.

This chapter introduced a novel method to extract regions of interest that are placed around road areas using a Gaussain-Mixture-Model (GMM) classifier. The presented colour-based road-extraction method using a GMM classifier is used in conjunction with either the Fixed-Cell-Distribution (FCD) or the Adaptive-Cell-Distribution (ACD) frameworks for the detection of cars in aerial images of urban scenes. It is integrated with the window-evaluation and post-processing stages in order to reduce the rate of false positives. Implementation details were given and discussed and a block diagram of the proposed framework is given in Figure 6.6. The combined FCD-GMM and ACD-GMM frameworks outperform the performance of both of the FCD and ACD frameworks as they achieve higher average precision rates of 68.47% and 69.70% on the Vaihingen dataset, respectively.

Chapter 7

Conclusions and Future work

THIS thesis has been concerned with the detection and localisation of cars in airborne imagery using robust ensembles of image descriptors. The primary focus is the development of a reliable system that detects cars in high-resolution urban scenes. The main research issue that has been addressed is the detection of small targets in cluttered environments. It has been claimed that a well-designed ensemble of image descriptors can divide the hyperspace into two almost linearly-separable subspaces. The first corresponds to the target-class objects and the second corresponds to the non-target-class objects even in cluttered environments.

Three frameworks have been proposed for the robust detection of cars. They follow a general four-stage methodology that consists of a window-evaluation stage, an extraction-and-encoding-of-feature stage, a classification stage and a post-processing stage.

First, the Fixed-Cell-Distribution (FCD) framework has been proposed. It exploits a novel ensemble of modified versions of Histogram-of-Oriented-Gradient, Fourier and truncated-Pyramid-Colour-Self-Similarity descriptors that characterise the local distribution of gradients, the spectral information and the local distribution of colours, respectively. These descriptors are computed over distributions of evenly-

spaced cells.

Second, the Adaptive-Cell-Distribution (ACD) framework has been proposed. It uses a clever ensemble of modified versions of Histogram-of-Oriented-Gradient, truncated-Pyramid-Colour-Self-Similarity and Gabor descriptors as well as their associated second-order statistics. They depict the local distribution of gradients, local distribution of colours and local distribution of texture, respectively. Image descriptors are computed over cell distributions that are function of the dominant orientation of the detection window under examination.

Third, in an attempt to reduce the rate of false positives by narrowing search areas, a new method that is based on the use of a Gaussian Mixture Model (GMM) classifier to automatically extract regions of interest of high probability to contain cars has been developed. This method is used in conjunction with either the FCD or the ACD frameworks for the detection of cars in aerial images of urban scenes. It is integrated with the window-evaluation and post-processing stages of both the FCD and ACD frameworks.

7.1 Conclusions

The performance of the proposed frameworks was assessed on two benchmark datasets, the Vaihingen and the OIRDS datasets, using different performance metrics, namely, the precision-recall, the mean average precision, recall rates at different false-positive-per-image rates and the interpolated average precision. Results corroborate their effectiveness in detecting cars robustly in real urban scenes that were captured under different acquisition conditions. In addition, obtained results endorse their superiority to the state of the art.

In the FCD framework, it was shown that an ensemble of modified versions of HOG, Fourier and tPCSS image descriptors can describe distinctively the most

characteristic features of cars. It has the ability to transform the car detection problem to a hyperspace of $4996D$, wherein the classification problem is almost linearly separable. HOG descriptor was used with its standard implementation to depict the local distribution of image gradients. However, it was shown that HOG can be modified to accommodate for rotation-invariance properties for cars in aerial images in a simple way. Orientation of cars can be estimated using a global histogram of oriented gradients. However, it fails to accurately estimate the orientation of cars of black colour that are surrounded by other objects.

The FCD framework proved its superiority to the work of Shao *et al.* [56] on the Vaihingen. Furthermore, on the OIRDS dataset, the FCD framework achieves higher precision than that of the work of Razakarivony and Jurie [58], despite the fact that no samples of this dataset were included in the training dataset. Nevertheless, in confounding situations, wherein there were objects that possess very similar visual structure to that of cars, the FCD framework produced few false positives. The FCD framework has a slightly lower performance than that of Madhogaria *et al.* [60]. Nevertheless, their method encountered the problem of detecting false positives with high confidence scores, indicating the unreliability of their work.

Several points can be learnt from the implementation and the results of the FCD framework. HOG descriptor can be modified to accommodate for rotation-invariance property for targets that possess a dominant orientation in an easy-to-implement way. In order to estimate the dominant orientation, a global histogram of gradients can be used. Nonetheless, it requires high strength edges. Added to that, local distributions of gradients and colours play an important role in describing a specific object, even in situations, where the object of interest is relatively smaller than other objects in the background.

Despite the fact that the truncated-Pyramid-Colour-Self-Similarity descriptor has the lowest dimensionality among the three exploited image descriptors, it is

very crucial to the discrimination process. The reason for this is that it is the only descriptor that encodes the distribution of colours. However, as black cars can have slightly-different colour distribution as the difference between the colour of windscreens and the colour of the car body is small, the exploited colour descriptor must be implemented in a way to tolerate this small variation in the expected colour distribution, a property that truncated-Pyramid-Colour-Self-Similarity descriptor possess.

In the ACD framework, an ensemble of modified versions of HOG, tPCSS and Gabor image descriptors as well as the their associated second-order statistics have been used. Thanks to incorporating the eigenvalues of the covariance matrices in the extracted feature vectors, the ACD framework achieves a better performance than the FCD framework in depicting the visual properties of cars as it has the ability to transform the car detection problem to a lower-dimensional hyperspace of $3838D$, wherein the classification problem is more linearly separable than the case of the FCD framework. This was validated from the obtained results.

As cars can be found at any orientation in aerial images, either detection windows can be rotated so that cars would have the same orientation or image descriptors can be computed using the novel technique of adaptive distributions of cells, wherein the distribution of cells is a function of the dominant orientation of a given detection window. The dominant orientation can be estimated with a very high accuracy using the eigenvalues of the covariance matrix of the local distribution of gradients. One of the main advantages of the ACD framework is that it circumvents the problem of changing relative distribution of the original pixels caused by non-linear transformation between co-ordinates as a result of the rotation process. In addition, it has lower computational cost than the process of window rotation.

In areas over-imposed by shadows, local normalisation of gradients is essential, if gradients descriptors are used. This normalisation process helps enhance

contrast and consequently, enhances low-strength gradients resulting in a better encoding of local gradients.

Furthermore, it can be deduced that although HOG is a powerful single-feature descriptor, its discriminability on its own is low, when objects of similar visual structures are present in the scenes under examination. To improve its performance, eigenvalues of the covariance matrices of the local cell responses can be used instead. This results in a more powerful descriptor with lower dimensionality. It also can be inferred from the progress achieved by the ACD framework that it is important to take into account features that distinguish the target and features that distinguish the background. This was shown, when Gabor kernels were adopted to distinguish textured surfaces such as vegetation and roofs of buildings, resulting in an improved performance.

The ACD framework outperformed the FCD framework on both the Vaihingen and OIRDS datasets as results have shown that it achieved a higher average precision. Nevertheless, the performance of the ACD framework is limited by the type of cars. It cannot detect cars of small size, e.g., mini cars, unlike the FCD framework. The ACD framework has superior performance to the work of Madhogaria *et al.* [60] despite using a smaller training dataset.

In heavily-cluttered environments, e.g., urban environments, the existence of many objects that are visually similar to cars can produce a high rate of false-positives. In order to overcome this problem, the combined FCD-GMM and ACD-GMM frameworks have been developed. They use an automatic method to eliminate detection windows that are not placed around roads, which results in a much lower false-positive rate and accordingly a higher precision. This indicates that the possible use of road maps to narrow down search areas to roads is useful and effective in urban areas. However, the main disadvantage of this approach is the dependence on the assumption that cars can only be found in road areas in urban scenes.

In general, the success of the three developed frameworks promotes that ensembles of image descriptors can work effectively to describe the most characteristic features of a given target. These ensembles can achieve high precisions with low false-positive rates. They can be linearly separable from image descriptors of non-target objects in low-dimensional spaces if they were chosen and designed carefully. It is also important to take into account distinctive visual properties of the background alongside the foreground in the choice of the exploited image descriptors.

7.2 Avenues of Future Research

Directions of future research include further enhancement of each stage of the proposed pipeline. The performance of the window-evaluation can be improved by the use of probabilistic models, such as the model used in [75] to measure the likeliness of windows to contain cars in addition to the current evaluation criteria. Another possible extension to the current implementation is the modification of the image descriptors that have been used, the combination of more image descriptors and the development of novel descriptors. Moreover, dimensionality reduction techniques can be used in order to shorten the processing time.

Among the limitations of the current implementation is that it is incapable of detecting cars from different altitudes. Hence, another possible area of improvement is the extension of the current implementation to accommodate for multi-scale analysis, which is intuitively straight forward. For example, the multi-scale procedure adopted by [77] can be implemented. Nevertheless, this requires the acquisition of a new dataset that is captured from different altitudes.

Speeding up of the current implementation that has near real-time performance for each of the three proposed framework by porting the code to the C++ environment and running on multiple cores. This would significantly shorten the

time required to process one detection window and consequently achieve a real-time performance.

Although the devised image descriptors were designed to the purpose of the detection of cars in airborne imagery, they have the potential advantage to be generalised and their application to detect other non-car targets can be investigated.

Finally, the use of deep neural networks to solve the problem of the detection of cars can be addressed. However, large-scale datasets of aerial images must be captured.

Bibliography

- [1] “The earth observation handbook,” 2012.
- [2] M. ElMikaty and T. Stathaki, “Car detection in high-resolution urban scenes using multiple image descriptors,” in *IAPR International Conference on Pattern Recognition*, 2014, pp. 4299–4304.
- [3] ———, “Detection and localisation of cars in airborne imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, under review.
- [4] ———, “Detection of cars in high-resolution aerial images of complex urban environments,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, under review.
- [5] J. Matas and S. Obdrzalek, “Object recognition methods based on transformation covariant features,” in *European Conference on Signal Processing*, 2004, pp. 1721–1728.
- [6] S. Shantaiya, K. Verma, and K. Mehta, “A survey on approaches of object detection,” *International Journal of Computer Applications*, vol. 65, no. 18, pp. 14–20, 2013.
- [7] C. Galleguillos and S. Belongie, “Context based object categorization: A critical survey,” *Computer Vision and Image Understanding*, vol. 114, no. 6, pp. 712–722, 2010.

- [8] C. Choi and H. I. Christensen, "Cognitive vision for efficient scene processing and object categorization in highly cluttered environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 4267–4274.
- [9] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [10] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [11] A. Thomas and K. Sreekumar, "A survey on image feature descriptors-color, shape and texture," *International Journal of Computer Science and Information Technology*, vol. 5, no. 5, pp. 7847–7850, 2014.
- [12] R. M. Kumar and K. Sreekumar, "A survey on image feature descriptors," *International Journal of Computer Science and Information Technology*, vol. 5, no. 5, pp. 7668–7673, 2014.
- [13] L. Wang, J. Shi, G. Song, and I. Shen, "Object detection combining recognition and segmentation," in *Asian Conference on Computer Vision*, 2007, pp. 189–199.
- [14] L. G. Roberts, "Machine perception of three-dimensional solids," Ph.D. dissertation, Massachusetts Institute of Technology, 1963.
- [15] D. G. Lowe, "The viewpoint consistency constraint," *International Journal of Computer Vision*, vol. 1, no. 1, pp. 57–72, 1987.
- [16] D. P. Huttenlocher and S. Ullman, "Recognizing solid objects by alignment with an image," *International Journal of Computer Vision*, vol. 5, no. 2, pp. 195–212, 1990.

- [17] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.
- [18] S. Giannarou and T. Stathaki, *A Novel Framework for Object Recognition under Sever Occlusion*. Springer, 2013, vol. 410, pp. 235–258.
- [19] —, “Shape signature matching for object identification invariant to image transformations and occlusion,” in *Computer Analysis of Images and Patterns*. Springer Berlin Heidelberg, 2007, vol. 4673, pp. 710–717.
- [20] H. Bay, A. Ess, T. Tuytelaars, and L. Gool, “Speeded-up robust features SURF,” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008.
- [21] D. G. Lowe, “Object recognition from local scale-invariant features,” in *IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.
- [22] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893.
- [23] J. Sivic and A. Zisserman, “Video google: a text retrieval approach to object matching in videos,” in *IEEE International Conference on Computer Vision*, 2003, pp. 1470–1477.
- [24] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes (VOC) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [25] Y. Bengio, “Learning deep architectures for AI,” *Foundations and Trends Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

- [26] L. Deng and D. Yu, *Deep Learning: Methods and Applications*. NOW Publishers, 2014.
- [27] I. Goodfellow, Y. Bengio, and A. Courville, “Deep learning,” 2016, book in preparation for MIT Press.
- [28] S. Behnke, *Hierarchical Neural Networks for Image Interpretation*. Springer-Verlag New York, Inc., 2003.
- [29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012, pp. 1097–1105.
- [30] A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 427–436.
- [31] R. Ruskone, L. Guigues, S. Airault, and O. Jamet, “Vehicle detection on aerial images: a structural approach,” in *IAPR International Conference on Pattern Recognition*, vol. 3, 1996, pp. 900–904 vol.3.
- [32] V. Ramakrishnan, A. K. Prabhavathy, and J. Devishree, “A survey on vehicle detection techniques in aerial surveillance,” *International Journal of Computer Applications*, vol. 55, no. 18, pp. 43–47, 2012.
- [33] T. Luo-Wei, H. Jun-Wei, and F. Kao-Chin, “Vehicle detection using normalized color and edge map,” in *IEEE International Conference on Image Processing*, vol. 2, 2005, pp. 598–601.
- [34] X. Shi, H. Ling, E. Blasch, and H. Weiming, “Context-driven moving vehicle detection in wide area motion imagery,” in *IAPR International Conference on Pattern Recognition*, 2012, pp. 2512–2515.

- [35] S. Sahli, Y. Ouyang, Y. Sheng, and D. A. Lavigne, “Robust vehicle detection in low-resolution aerial imagery,” in *Proceedings of SPIE on Airborne Intelligence, Surveillance, Reconnaissance Systems and Applications*, vol. 7668, 2010, p. 76680.
- [36] X. Cao, C. Wu, P. Yan, and X. Li, “Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos,” in *IEEE Conference on Image Processing*, 2011, pp. 2421–2424.
- [37] X. Chen, S. Xiang, C. L. Liu, and C. H. Pan, “Vehicle detection in satellite images by hybrid deep convolutional neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, pp. 1797–1801, 2014.
- [38] T. Zhao and R. Nevatia, “Car detection in low resolution aerial image,” in *IEEE International Conference on Computer Vision*, vol. 1, 2001, pp. 710–717 vol.1.
- [39] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [40] S. Hinz, “Detection and counting of cars in aerial images,” in *IEEE International Conference on Image Processing*, vol. 3, 2003, pp. 997–1000.
- [41] S. Hinz, C. Schlosser, and J. Reitberger, “Automatic car detection in high resolution urban scenes based on an adaptive 3D-model,” in *GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas*, 2003, pp. 167–171.
- [42] J. Choi and Y. Yang, “Vehicle detection from aerial images using local shape information,” in *Advances in Image and Video Technology*, ser. Lecture Notes in Computer Science. Springer, 2009, vol. 5414, pp. 227–236.
- [43] D. Comaniciu and P. Meer, “Mean shift analysis and applications,” in *IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1197–1203 vol.2.

- [44] T. Nguyen, H. Grabner, H. Bischof, and B. Gruber, “On-line boosting for car detection from aerial images,” in *IEEE International Conference on Research, Innovation and Vision for the Future*, 2007, pp. 87–95.
- [45] P. Viola and M. Jones, “Robust real-time face detection,” in *IEEE International Conference on Computer Vision*, vol. 2, 2001, pp. 747–747.
- [46] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [47] R. Schapire, “The boosting approach to machine learning: An overview,” in *Nonlinear Estimation and Classification*. Springer New York, 2003, vol. 171, pp. 149–171.
- [48] A. Blum, “On-line algorithms in machine learning,” in *Online Algorithms: The State of the Art*. Springer Berlin Heidelberg, 1998, pp. 306–325.
- [49] H. Garbner, C. Beleznai, and H. Bischof, “Improving adaboost detection rate by wobble and mean shift,” in *Computer Vision Winter Workshop*, 2005, pp. 23–32.
- [50] B. J. Frey and D. Dueck, “Clustering by passing messages between data points,” *Science*, vol. 315, no. 5814, pp. 972–976, 2007.
- [51] Y. Freund and R. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997.
- [52] A. Kembhavi, D. Harwood, and L. Davis, “Vehicle detection using partial least squares,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1250–1265, 2011.

- [53] S. Sahli, P. Duval, Y. Sheng, and D. Lavigne, “Robust vehicle detection in aerial images based on salient region selection and superpixel classification,” in *Proceedings of SPIE on Airborne Intelligence, Surveillance, Reconnaissance Systems and Applications*, vol. 8020, 2011, p. 80200.
- [54] P. L. Rosin, “A simple method for detecting salient regions,” *Pattern Recognition*, vol. 42, no. 11, pp. 2363 – 2371, 2009.
- [55] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, “Slic superpixels,” École Polytechnique Fédérale de Lausanne (EPFL), Tech. Rep., 2010.
- [56] W. Shao, W. Yang, G. Liu, and J. Liu, “Car detection from high-resolution aerial imagery using multiple features,” in *IEEE International Geoscience and Remote Sensing Symposium*, 2012, pp. 4379–4382.
- [57] S. Maji, A. Berg, and J. Malik, “Classification using intersection kernel support vector machines is efficient,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [58] S. S. Razakarivony and F. Jurie, “Small target detection combining foreground and background manifolds,” in *IAPR International Conference on Machine Vision Applications*, 2013, p. 4.
- [59] M. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AIChE Journal*, vol. 37, no. 2, pp. 233–243, 1991.
- [60] S. Madhogaria, P. M. Baggenstoss, M. Schikora, W. Koch, and D. Cremers, “Car detection by fusion of hog and causal mrf,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 575–590, 2015.

- [61] H. Moon, R. Chellappa, and A. Rosenfeld, “Optimal edge-based shape detection,” *IEEE Transactions on Image Processing*, vol. 11, no. 11, pp. 1209–1227, 2002.
- [62] —, “Performance analysis of a simple vehicle detection algorithm,” *Image and Vision Computing*, vol. 20, no. 1, pp. 1–13, 2002.
- [63] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
- [64] D. Ballard, “Generalizing the hough transform to detect arbitrary shapes,” *Pattern Recognition*, vol. 13, no. 2, pp. 111 – 122, 1981.
- [65] H. Zheng, L. Pan, and L. Li, “A morphological neural network approach for vehicle detection from high resolution satellite imagery,” in *Neural Information Processing*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, vol. 4233, pp. 99–106.
- [66] J. Serra, *Image Analysis and Mathematical Morphology*. Academic Press, Inc., 1983.
- [67] D. Svozil, V. Kvasnicka, and J. Pospichal, “Introduction to multi-layer feed-forward neural networks,” *Chemometrics and Intelligent Laboratory Systems*, vol. 39, no. 1, pp. 43 – 62, 1997.
- [68] L. Eikvil, L. Aurdal, and H. Koren, “Classification-based vehicle detection in high-resolution satellite images,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 1, pp. 65 – 72, 2009.
- [69] K. Fu and J. Mui, “A survey on image segmentation,” *Pattern Recognition*, vol. 13, no. 1, pp. 3 – 16, 1981.

- [70] S. Tuermer, F. Kurz, P. Reinartz, and U. Stilla, “Airborne vehicle detection in dense urban areas using hog features and disparity maps,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, no. 99, pp. 1–11, 2013.
- [71] M. Cramer, “The DGPF-test on digital airborne camera evaluation-overview and test design,” *Photogrammetrie-Fernerkundung-Geoinformation*, vol. 2, no. 2, pp. 73–82, 2010.
- [72] F. Tanner, B. Colder, C. Pullen, D. Heagy, M. Eppolito, V. Carlan, C. Oertel, and P. Sallee, “Overhead imagery research data set an annotated data library tools to aid in the development of computer vision algorithms,” in *IEEE Workshop on Applied Imagery Pattern Recognition*, 2009, pp. 1–8.
- [73] *Digital Mapping Camera System*, Intergraph, 2008.
- [74] V. Y. Mariano, M. Junghye, P. Jin-Hyeong, R. Kasturi, D. Mihalcik, H. Li, D. Doermann, and T. Drayer, “Performance evaluation of object detection algorithms,” in *IAPR International Conference on Pattern Recognition*, vol. 3, 2002, pp. 965–969 vol.3.
- [75] B. Alexe, T. Deselaers, and V. Ferrari, “Measuring the objectness of image windows,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2189–2202, 2012.
- [76] P. Dollar, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: An evaluation of the state of the art,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.
- [77] N. Dalal, “Finding people in images and videos,” Ph.D. dissertation, Institut Polytechnique de Grenoble, 2006.

- [78] J. Gorman, O. Mitchell, and F. Kuhl, “Partial shape recognition using dynamic programming,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 2, 1988.
- [79] M. Sarfraz, “Object recognition using fourier descriptors: Some experiments and observations,” in *Computer Graphics, Imaging and Visualisation. International Conference on*, 2006, pp. 281–286.
- [80] E. Karaman, U. Cinar, E. Gedik, Y. Yardimci, and U. Halici, “Fourier based feature descriptors for railroad extraction from aerial images,” in *IEEE International Geoscience and Remote Sensing Symposium*, 2012, pp. 6013–6015.
- [81] E. Mavridou, H. Manh-Dung, J. Crowley, and A. Lux, “Scale normalized radial fourier transform as a robust image descriptor,” in *IAPR International Conference on Pattern Recognition*, 2014, pp. 3993–3998.
- [82] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 3rd ed. Thomson Learning, 2007.
- [83] S. Walk, N. Majer, K. Schindler, and B. Schiele, “New features and insights for pedestrian detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1030–1037.
- [84] N. Wang, X. Gong, and J. Liu, “A new depth descriptor for pedestrian detection in rgb-d images,” in *IAPR International Conference on Pattern Recognition*, 2012, pp. 3688–3691.
- [85] D. Tse and P. Viswanath, *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.
- [86] L. SHEN and Z. JI, “Gabor wavelet selection and SVM classification for object recognition,” *Acta Automatica Sinica*, vol. 35, no. 4, pp. 350 – 355, 2009.

-
- [87] J. Daugman, “Complete discrete 2-d gabor transforms by neural networks for image analysis and compression,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 7, pp. 1169–1179, Jul 1988.
- [88] C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [89] G. McLachlan and D. Peel, *Finite Mixture Models*. Wiley, 2000.

Appendix A

Support Vector Machines

Support Vector Machines (SVM) have been exploited in many machine learning tasks, such as detection, recognition, classification, categorisation and regression [88]. The SVM classifier aims to define a hyperplane that divides the hyperdimensional space into two subspaces. Therefore, it is considered a decision supervised classifier as it does not produce an output posterior probability. It is a convex optimisation problem that maximises the margin between the two data samples classes. Accordingly, it does not take into account all the training samples. Instead, it depends on a set of sparse samples that constitute the support vectors [39] that are close to the separating hyperplane. The mathematical analysis of how the SVM classifier works is presented below and it is based on the analysis of Bishop [39].

Assuming that the total number of negative and positive data samples $s_k \in \{s_1, \dots, s_K\}$ is K and that they were drawn from a hyperspace of dimensionality D and their associated binary target values $t_k \in \{-1, 1\}$, the goal is to solve the classification problem using a linear model in the of form of Equation A.1.

$$z(s) = \mathbf{w}^T \phi(s) + b \tag{A.1}$$

where $\phi(s)$ represents the transformation from the spatial domain to the hyper-dimensional space, wherein each data sample s_k is represented by a hyper-dimensional feature vector $\mathbf{x}_k$ and b is a bias. In other words, the goal is to determine the hyperplane in the feature space $\phi(S) \in \mathbb{R}^D$ that best separates the target and non-target classes. This can be achieved by maximising the margin between the hyperplane and data samples that are close to it as mentioned before. The distance from a data sample in the hyperspace and the separating hyperplane can be calculated using Equation A.2 [39].

$$\frac{t_k z(s_k)}{\|\mathbf{w}\|} = \frac{t_k (\mathbf{w}^T \mathbf{x}_k + b)}{\|\mathbf{w}\|} \quad (\text{A.2})$$

Parameters $\mathbf{w}$ and b are optimised in order to maximise the margin using the training data samples and the solution is given by Equation A.3.

$$\operatorname{argmax}_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_k [t_k (\mathbf{w}^T \mathbf{x}_k + b)] \right\} \quad (\text{A.3})$$

The above optimisation problem can be reduced to minimising $\|\mathbf{w}\|^2$ as in Equation A.4 with the constraint given in Equation A.5 [39].

$$\operatorname{argmin}_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (\text{A.4})$$

$$t_k (\mathbf{w}^T \mathbf{x}_k + b) \geq 1 \quad (\text{A.5})$$

These assumptions are valid only for perfect linearly separable classes which is not the case in many scenarios. For overlapping classes, a slack variable $\xi_k \geq 0$ where $k = 1, \dots, K$ is introduced to allow some training samples to be misclassified, changing the quadratic optimisation problem in Equation A.4 to a constrained

minimisation problem of $\|\mathbf{w}\|^2$ in the form of Equation A.6 and the associated constraints are in the form of Equation A.7.

$$c \sum_{k=1}^K \xi_k + \frac{1}{2} \|\mathbf{w}\|^2 \quad (\text{A.6})$$

subject to the following constraints:

$$\begin{cases} t_k(\mathbf{w}^T \mathbf{x}_k + b) \geq 1 - \xi_k \\ \xi_k \geq 0 \end{cases} \quad (\text{A.7})$$

where c is a non-negative regularisation constant and it controls the trade-off between the slack variable ξ_k and the margin. If the distance from the hyperplane that is computed using Equation A.1 is found to be positive and greater than a threshold that is found empirically, the test sample is classified as a detection. More details about SVM classifiers can be found in [39] and [88].

In the implementation of the FCD framework, $\phi(s_k)$ transforms the training samples to a feature vector $\mathbf{x}_k$ of dimensionality of $4996D$ using a regularisation parameter c of 15, whereas in the implementation of the ACD framework, $\phi(s_k)$ transforms the training samples to a feature vector $\mathbf{x}_k$ of dimensionality of $3838D$ using a regularisation parameter c of 15.

In order to classify a new data sample s , the distance from the separating hyperplane is calculated using the optimal weight $\mathbf{w}_{opt}$ and bias b using Equation A.8. The sign of $z(s)$ determines on which side of the hyperplane the data sample s is and the distance from the hyperplane represents its confidence score.

$$z(s) = \mathbf{w}_{opt}^T \phi(s) + b \quad (\text{A.8})$$

Appendix B

Gaussian Mixture Models

Gaussian distributions are known for their important analytical features [89]. The Gaussian Mixture Model (GMM) tries to model the probability density function of a given set of data using a linear combination of several Gaussian distributions in an unsupervised way and the GMM classifier tries to assign a new data sample to one of these components based on the output posterior probability. The mathematical analysis of how the GMM classifier works is presented below and it is based on the analysis of Bishop in [39].

Assuming that the total number of negative and positive data samples $s_k \in \{s_1, \dots, s_K\}$ is K and that they were drawn from a hyperspace of dimensionality D and that each is represented by a hyper-dimensional feature vector $\mathbf{x}_k$, the goal is to model the probability density function of $\mathbf{x}$ using an N multivariate Gaussian components in the form of Equation B.1

$$pdf(\mathbf{x}) = \sum_n \alpha_{\mathbf{x},n} \mathcal{N}(\mathbf{x} | \mu_{\mathbf{x},n}, \mathbb{R}_{\mathbf{xx},n}), n \in \{1, 2, \dots, N\} \quad (\text{B.1})$$

where $\mathcal{N}(\mathbf{x} | \mu_{\mathbf{x},n}, \mathbb{R}_{\mathbf{xx},n})$ is one Gaussian component with mean vector $\mu_{\mathbf{x},n}$ and covariance $\mathbb{R}_{\mathbf{xx},n}$, N represents the total number of Gaussians used to represent the

distribution of data samples in the hyper space, $\alpha_{\mathbf{x},n}$ is the mixing coefficient with constraints defined in Equation B.2 and the Gaussian component $\mathcal{N}(\mathbf{x}|\mu_{\mathbf{x},n}, \mathbb{R}_{\mathbf{xx},n})$ is a multivariate Gaussian function defined in Equation B.3.

$$\begin{cases} \sum_n \alpha_{\mathbf{x},n} = 1 \\ 0 \leq \alpha_{\mathbf{x},n} \leq 1 \end{cases} \quad (\text{B.2})$$

$$\mathcal{N}(\mathbf{x}|\mu_{\mathbf{x},n}, \mathbb{R}_{\mathbf{xx},n}) = \frac{1}{(2\pi)^{D/2} |\mathbb{R}_{\mathbf{xx},n}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu_{\mathbf{x},n})^T \mathbb{R}_{\mathbf{xx},n}^{-1} (\mathbf{x} - \mu_{\mathbf{x},n}) \right\} \quad (\text{B.3})$$

The mean $\mu_{\mathbf{x},n}$ and covariance $\mathbb{R}_{\mathbf{xx},n}$ of each Gaussian component are estimated from the training dataset using the known iterative expectation-maximisation algorithm (EM) [89] that aims to find the maximum likelihood solutions [39]. This iterative algorithm has two main steps the E step and the M step.

In the E step, current values for $\mu_{\mathbf{x},n}$, $\mathbb{R}_{\mathbf{xx},n}$ and $\alpha_{\mathbf{x},n}$ are used to compute an estimation of the posterior probabilities $\hat{\gamma}_{k,n}$ that using Equation B.4.

$$\hat{\gamma}_{k,n} = \frac{\alpha_{x,n} \mathcal{N}(\mathbf{x}_k | \mu_{\mathbf{x},n}, \mathbb{R}_{\mathbf{xx},n})}{\sum_j \alpha_{\mathbf{x},j} \mathcal{N}(\mathbf{x}_k | \mu_{\mathbf{x},j}, \mathbb{R}_{\mathbf{xx},j})}, j \in \{1, 2, \dots, N\} \quad (\text{B.4})$$

In the M step, parameters $\mu_{\mathbf{x},n}$, $\mathbb{R}_{\mathbf{xx},n}$ and $\alpha_{\mathbf{x},n}$ are recomputed based on the estimated $\hat{\gamma}_{in}$ using Equation B.5, Equation B.6, Equation B.7 and Equation B.8.

$$\Gamma_n = \sum_k \hat{\gamma}_{k,n}, k \in \{1, 2, \dots, K\} \quad (\text{B.5})$$

$$\hat{\mu}_{\mathbf{x},n} = \frac{1}{\Gamma_n} \sum_k \hat{\gamma}_{k,n} \mathbf{x}_k, k \in \{1, 2, \dots, K\} \quad (\text{B.6})$$

$$\hat{\mathbb{R}}_{\mathbf{xx},n} = \frac{1}{\Gamma_n} \sum_k \hat{\gamma}_{k,n} (\mathbf{x}_k - \hat{\mu}_{\mathbf{x},n})(\mathbf{x}_k - \hat{\mu}_{\mathbf{x},n})^T, k \in \{1, 2, \dots, K\} \quad (\text{B.7})$$

$$\hat{\alpha}_{\mathbf{x},n} = \frac{\Gamma_n}{K} \quad (\text{B.8})$$

The log likelihood is then computed using Equation B.9 and the above procedure is iterated till convergence is achieved.

$$\ln pdf(\mathbf{X}|\mu_{\mathbf{X}}, \mathbb{R}_{\mathbf{XX}}, \alpha) = \sum_k \ln \left\{ \sum_n \alpha_n \mathcal{N}(\mathbf{x}_k|\mu_{\mathbf{x},n}, \mathbb{R}_{\mathbf{xx},n}) \right\} \quad (\text{B.9})$$

$$k \in \{1, 2, \dots, K\}, n \in \{1, 2, \dots, N\}$$

A new data sample is assigned to one of the Gaussian components according to the maximum posterior probability. More details about the GMM and the EM algorithm can be found in [39] and [89].