DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PEDESTRIAN DETECTION FOR RAILWAY DRIVER SUPPORT SYSTEMS

by

Tuğçe TOPRAK

July, 2018

İZMİR

PEDESTRIAN DETECTION FOR RAILWAY DRIVER SUPPORT SYSTEMS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences Dokuz Eylül University
In Partial Fullfilment of the Requirements for the Degree of Master of Science
in Electrical and Electronics Engineering

by
Tuğçe TOPRAK

July, 2018
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read this thesis entitled "**PEDESTRIAN DETECTION FOR RAILWAY DRIVER SUPPORT SYSTEMS**" completed by **TUĞÇE TOPRAK** under supervision of **ASSOC. PROF. DR. MUSTAFA ALPER SELVER** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

……………………………………………………

Assoc. Prof. Dr. Mustafa Alper SELVER

Supervisor

……………………………………………… ……………………………………………

Dr. Metehan Makınacı

(Jury Member)

Assist. Prof. M. Zübeyir Ünlü

(Jury Member)

L. Salum

Prof. Dr. Latif SALUM
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my thesis advisor Assoc. Prof. Dr. M. Alper Selver for his guidance, valuable suggestions, and patiance. I also would like to present my appreciation to Prof. Dr. E. Yeşim Zoral for her guidance during my master studies.

In addition, I would like to thank my brother Göktuğ Toprak for his support and bringing happiness to me, though he also has a rough ride, so glad i have you. I would like to express my deepest gratitude to my parents for their support, patience and understanding throughout my life.

At last, I would like to thank Ersin Tepegöz for his trust, patience and encouragement. None of this would have been possible without you.

Tuğçe TOPRAK

PEDESTRIAN DETECTION FOR RAILWAY DRIVER SUPPORT SYSTEMS

ABSTRACT

Pedestrian detection is one of the most studied issues of advanced driver assistance systems. Although a tremendous effort is already given to create adequate datasets and to develop advanced classifiers for cars, studies about railway systems remain very limited.

The research done within the scope of the thesis shows that direct application of neither existing advanced object detection systems, nor specifically created ones for pedestrian detection (such as classifiers which is pre-trained well-known pedestrian datasets Caltech, INRIA etc.), can provide enough performance to overcome railway specific challenges. Fortunately, it is also shown that without waiting the collection of a mature dataset for railways as comprehensively diverse and annotated as the existing ones for cars, a transfer learning approach to fine-tune various successful deep models (pre-trained using both extensive image and pedestrian datasets) to railway pedestrian detection tasks provides an effective solution.

In the light of this information, to achieve transfer learning, a new Railway Pedestrian Dataset (RAWPED) is collected, annotated and divided into challenge based subgroups. Moreover, the localization and adaptation limitations of deep models are resolved with a feature-classifier ensemble. The application of resulting two stage system to various railway scenes demonstrate that employed transfer learning strategies enable reliable adaptation of pre-trained models to railway pedestrian detection scenes. Furthermore, complementary properties of the transferred models, classifiers and diversity of their results are analyzed. Based on the findings, a novel machine learning strategy is structured to create an ensemble, which defragments outputs of individual models and performs consistently better than its components.

Keywords: Pedestrian detection, transfer learning, railways, classifier ensembles

DEMİRYOLU SÜRÜCÜ DESTEK SİSTEMLERİ İÇİN YAYA SAPTAMA

ÖZ

Yaya saptama, gelişmiş sürücü destek sistemlerinin üzerinde en çok çalışılan konularından biridir. Karayolu araçları için, yeterli veri kümeleri oluşturmak ve üstün sınıflayıcılar geliştirmek üzere çok büyük çaba sarfedilmiş olmasına rağmen demiryolu sistemleriyle ilgili çalışmalar çok sınırlı kalmaktadır.

Tez kapsamında yapılan çalışmalar, var olan gelişmiş nesne saptama sistemlerinin (örneğin, AlexNet, VGG) ve özel olarak yaya saptama üzerine geliştirilmiş sistemlerin (örneğin, Caltech ya da INRIA gibi denektaşı veri kümeleri ile eğitilmiş sınıflayıcılar) doğrudan demiryolu çalışmalarına yapılan uygulamalarında, bu sistemde var olan zorlukların üstesinden gelebilecek yeterli performansı sağlamadığını göstermektedir. Neyse ki, demiryolu sistemi için karayolu araçlarında mevcut olduğu gibi kapsamlı ve açıklayıcı bir veri kümesinin olgunlaşmasını beklemeksizin, önceden bu kapsamlı veri kümeleri kullanılarak yaya saptamak üzere eğitilmiş çeşitli başarılı derin ağ modellerine aktarmalı öğrenme yaklaşımının uygulanması, bu sistemlerde yaya saptama amacına etkili bir çözüm sağlamaktadır.

Bu bilgiler ışığında, aktarmalı öğrenme yaklaşımının uygulanabilmesi için, Demiryolu Yaya Veri Kümesi (Railway Pedestrian Dataset-RAWPED) toplanmış, etiketlenmiş ve zorluk seviyelerine göre alt gruplara ayrılmıştır. Buna ek olarak, derin ağ modellerinin, yer belirleme ve adaptasyon sınırlamaları bir sınıflayıcı topluluğu ile çözülmüştür. Elde edilen iki aşamalı sistemin çeşitli demiryolu sahnelerine uygulanması, kullanılan aktarmalı öğrenme stratejilerinin, önceden eğitilmiş modellerin yaya saptama amacı ile demiryolu sahnelerine güvenilir şekilde adaptasyonunu mümkün kıldığını göstermektedir. Ayrıca, aktarılan modellerin tamamlayıcı özellikleri ve sonuçlarının çeşitliliği de analiz edilmiştir. Tüm bu bulgulara dayanarak, ayrı ayrı modellerin sonuçlarını birleştiren ve bileşenlerinden sürekli olarak daha iyi performans gösteren yeni bir makine öğrenmesi stratejisi yapılandırılmıştır.

Anahtar Kelimeler: Yaya saptama, aktarmalı öğrenme, demiryolu, sınıflayıcı toplulukları

CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

Railroad accident prevention needs various novel technological developments for all types of transporters. Currently, pedestrian accidents are the leading cause of death on railways (Lavalle, 2015) and therefore, on-board driver assistance systems are required in order to protect lives with instant reactions and to collect information about near–miss events for long term analyzes and planning (Aminmansour, Maire, Laure & Wullems, 2015). Despite multi-sensor pedestrian detection possibility in self-driving cars (Bila, Sivrikaya, Khan, & Albayrak, 2017), railway on-board driver assistance systems mainly rely on camera vision due to the application limitations (economic infeasibility, short range, vibration sensitivity etc.) of alternative sensors (LIDAR, RADAR, ultrasonic etc.) (Selver, Atac, Belenlioglu, Dogan & Zoral, 2017; Li, Wang X., Xu & Wang J., 2016).

Camera based pedestrian detection is becoming a mature field of pedestrian protection systems (Gandhi & Trivedi, 2007) with outstanding achievements for self-driving cars in the last decade (Enzweiler & Gavrila, 2009; Geronimo, Lopez, Sappa & Graf, 2010; Zhang, Benenson, Omran, Hosang & Schiele, 2017). Although various categorizations are possible, the existing systems can be divided into two in the context of this study. The first one consists of systems that use hand-crafted feature extraction (e.g. histogram of gradients (HOG) (Dalal & Triggs, 2005), aggregated or locally decorrelated channel features (ACF (Dollar, Appel, Belongie & Perona, 2014) or LDCF (Nam, Dollar & Han, 2014)) followed by a classifier (e.g. support vector machine (SVM) (Dalal & Triggs, 2005), AdaBoost cascades (Dollar et. al., 2014)). The second category includes deep learning strategies, especially convolutional neural networks (CNN), with internal feature representation by low-level (i.e. generalized) properties produced by the initial layers and high-level (i.e. application specific) ones associated with the last layers of the model (Benenson, Omsan, Hosang & Schiele, 2014; Hosang, Omran, Beneson & Schiele, 2015).

The complexity of pedestrian detection problem enables challenge-based systems to improve performance. Some examples of these challenges and associated solutions can be related to appearance such as high variations at shape (Zhang, Bauckhage & Cremers, 2015), pose (Li, Chen & Wang, 2015), spatial scales (Li et. al., 2016), illumination (Liu et. al., 2015) and background clutter (Simonnet, Velastin, Turkbeyler & Orwell, 2012). A second set of challenges is caused by uncertainty of evaluation such as partial occlusion (Li et. al., 2014), deformation or truncation (Ouyang et. al., 2017), and people-person decision (Ouyang, Zeng & Wang, 2013). These endless challenges motivate creation of datasets with diverse properties (i.e. Caltech (Dollar, Wojek, Schiele & Perona, 2012), INRIA (Dala & Triggs, 2005), Citypersons (Zhang, Benenson & Schiele, 2017), KITTI (Geiger, Lenz & Urtasun, 2013), TUD (Wojek, Walk & Schiele, 2013), ETH (Ess, Leibe, Schindler & Van Gool, 2008), Daimler (Enzweiler & Gavrila, 2009)) to represent real life situations in a better way. Moreover, existing metrics are rectified and new measures are introduced in parallel to these developments (Dollar et. al., 2012). Studies that compare the performance of different algorithms applied to the same dataset and compare the results from different perspectives are occasionally being published (Benenson et. al., 2014; Dollar et. al., 2012).

1.1 Aim of the Study

Up to our knowledge, none of these techniques are applied to railway on-board driver assistance systems yet, except studies using fixed cameras at pre-defined locations such as stations and level crossings (Kirbas & Quek, 2004; Freeman & Rakotonirainy, 2015). As vision based applications such as railroad extraction (Aminmansour et. al., 2015; Selver et. al., 2017; Nassu & Ukai, 2012; Selver, Belenlioglu & Soyaslan, 2016) or anti-collision systems for maintenance vehicles (Maire, 2007) are studied in detail, it is clear that camera is a key element of on-board driver assistance systems and the next step is to use it for pedestrian detection.

Accordingly, this thesis presents a new dataset, a novel system, comprehensive tests and their results for pedestrian detection in on-board driver assistance systems of intelligent railway transportation systems. The four main contributions of the study can be listed as follows:

1. A Railway Pedestrian Dataset (RAWPED), which is the first of its kind, is constructed. The videos, which are acquired during ADORAS project and collected from public domain, are trimmed to include pedestrians that vary widely in appearance. Furthermore, different scenarios are created (e.g. approaching a level crossing or a station) to provide better representation of real world situations and allow in depth analyzes. All frames of interest are manually annotated.

2. Several pre-trained benchmark models including feature-classifier strategies and deep networks are tested on RAWPED. Both multi-purpose object detectors and pedestrian detection systems are included. Important situations of practical interest are highlighted under which existing models fail.

3. The outcome of the two analyzes mentioned above show that the performances of pre-trained models on RAWPED are not accurate enough. Since RAWPED is not as comprehensive as other benchmark datasets for pedestrian detection, an ensemble version of detectors are examined for reducing false negatives and increase the performance with compared to individual detectors.

4. To further decrease the total false positives of the ensemble non-deep models, the complementarity of fine-tuned and trained CNNs and SVM classifiers is also investigated and a new ensemble system is proposed (Figure 1.1). The novelty of the proposed system has two folds. In the first level, non-deep pre-trained detectors are combined in such a way that maximum number of targets (pedestrians) is collected without a penalty for false positives. Then, at the second level, transferred network ensemble is constructed by employing a transfer learning strategy through the objective of false positive elimination.

1.2 Thesis Outline

This thesis consists of six chapters. Chapter One presents the introduction section in order to provide information, including pedestrian detection issue, substantial techniques about detection and prior knowledge for necessity of railway on-board driver assistance systems. Related works about techniques and ensemble systems are given in Chapter Two. Chapter Three informs about RAWPED and its technical properties comparison with other benchmark pedestrian datasets. Chapter Four presents the information about each model which can be used in ensembled system. The evaluation process and analyzes of complementarity of non-deep detectors are given in Chapter Five. Chapter Six includes the applications, their results and also conclusion.

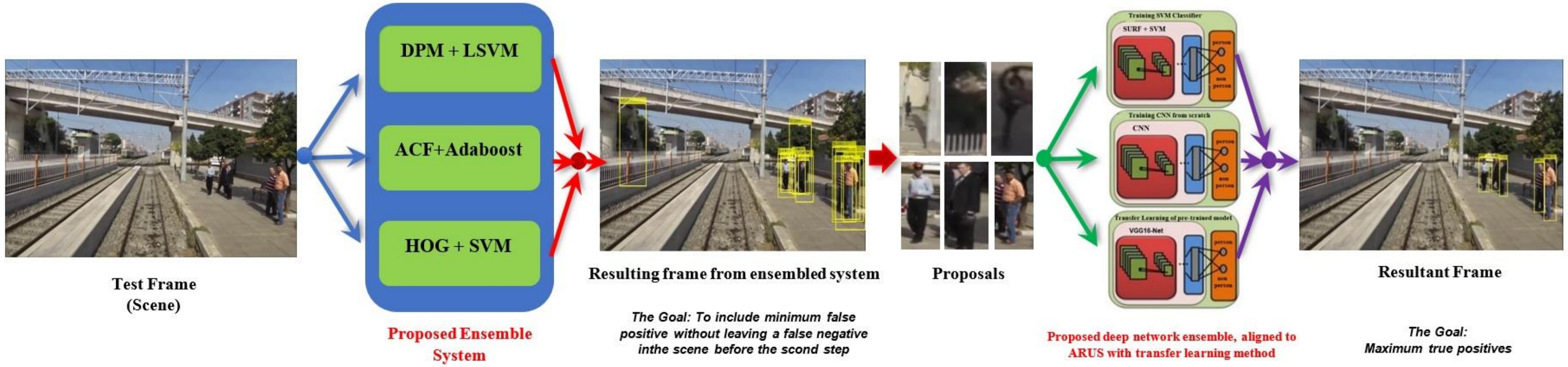

Test Frame
(Scene)
DPM + LSVM
ACF+Adaboost
HOG + SVM
Proposed Ensemble System
Resulting frame from ensembled system
The Goal: To include minimum false positive without leaving a false negative inthe scene before the scond step
Proposals
Training SVM Classifier
SURF + SVM
person
non person
Training CNN from scratch
CNN
person
non person
Transfer Learning of pre-trained model
VGG16-Net
person
non person
Proposed deep network ensemble, aligned to ARUS with transfer learning method
Resultant Frame
The Goal:
Maximum true positives

Figure 1.1 Proposed strategy of pedestrian detection for on-board railway driver assistance systems

CHAPTER TWO
RELATED WORK

There exist two main models for pedestrian detection:

1. Systems creating a bounding box including confidence scores (Dalal & Triggs, 2005; Dollar et. al., 2014; Nam et. al., 2014).
2. Deep networks making binary decisions (i.e. person or not) for a given window (i.e. sub-image).

The former models are specifically designed for pedestrian detection and computationally efficient when complete image is considered (Dollar et. al., 2012). The latter models consist of general purpose detectors having capacity to classify thousands of different objects. They are more accurate and faster for a small region of interest rather than complete scene. These complementary properties of two models motivate several studies which focus on their combined usage as discussed in the next subsection.

2.1 Related Work on Classifier Ensembles and Feature Fusion

Recent studies show that cascaded models or employing their ensembles can boost the results in pedestrian detection (Zhang et. al., 2017; Benenson et. al., 2014; Wang, Choi & Lin, 2016; Benenson, Mathias, Timofte & Van Gool, 2012; Angelova, Krizhevsky, Vanhoucke, Ogale & Ferguson, 2015). Among those, two major strategies seem to come forward with heterogeneous combinations of non-deep and deep detectors.

The first one is replacing or integrating hand-crafted features with high-level features, which are implicitly obtained from the last layers of a deep network, and fed them to a (pre-trained) deep (Ribeiro, Carneiro, Nascimento & Bernardino, 2017; Hu, Wang, Shen, Van Den Hangel & Porikli, 2017; Cai, Saberian & Vasconcelos, 2015) or non-deep classifier (Cao, Pang & Li, 2017) or ensembles created by their combinations.

The second strategy relies on selecting one of many non-deep state-of-the-art pedestrian detectors through a switching mechanism in order to create bounding boxes, called “proposals”, which are fed for classification to a deep network (such as CNN) pre-trained on benchmark datasets such as ImageNet (Ribero, Nascimento, Bernardino & Carneiro, 2017). This approach is further improved by incorporating bounding box scores to the learning process (Li et. al, 2017).

It is also shown that different types of features yield improved results for their combinations (Benenson et. al., 2014). Here, the critical point is how to combine them in order to take advantage of diversity and integrate their complementary properties to training. Applications show performance improvements such as spatial pooling for maximizing the detection rate at a user defined range (Paisitkriangkrai, Shen & Van Den Hengel, 2016), joint learning for different levels of occlusion (Zhu & Peng, 2015) or occlusion-deformation (Ouyang et. al., 2017), and feature co-occurrence selection (Li Q., Wang, Yan, Li B. & Chen, 2017). Despite their success, no optimal method has been proposed for combining features or constructing ensembles yet. Thus, the problem of integrating features and classifiers still remains application dependent.

2.2 Related Work on Transfer Learning

The above mentioned advanced methods for pedestrian detection show improved performance only when the training and testing datasets have very similar distributions in the feature space. In other words, when the same dataset is divided into training and test groups for performance analyzes, inter data usability of the detectors, which is very important in practice because of the scene complexity and variations, is ignored (Yosinski, Clune, Bengio & Lipson, 2014). This causes a dramatic decrease in pedestrian detection performance when training and test groups are selected from different datasets (Benenson et. al., 2014). This lack of adaptation can be compensated with transfer learning, which takes advantage of the correlation between the datasets for re-adjusting the last layers of the networks using a relatively smaller dataset (Pan & Yang, 2010).

Transfer learning provides a principled way to solve domain adaptation problems and therefore, it has been successfully applied to various pedestrian detection related issues including classification of objects (Kulis, Saenko & Darrell, 2011), scenes (Qi et. al., 2011), actions (Liu, Shah, Kuipers & Savarese, 2011), and visual concepts (Duan, Tsang, Xu & Maybank, 2009; Wang X., Wang M. & Li, 2014).

The studies on transfer learning for pedestrian detection mainly focus on improving diminished performance of a generic detector for specific scenes due to the mismatch between the training and the target sets (Roth, Sternig, Grabner & Bischof, 2009; Stalder & Grabner, 2010; Ali, Hasler & Fleuret, 2011). However, when the scene becomes steady, detector can take advantage of several assumptions such as the location of negative samples to train separate detectors for different regions (Stalder & Grabner, 2010) or scene geometry to assist labeling (All et. al., 2011). Moreover, a sparse labeling can be enough for training when supported by video tracking (All et. al., 2011).

Having very dynamic sceneries, pedestrian detection for railway on-board driver assistance systems needs to deal with more challenging and application specific problems such as adaption of the distribution of pedestrian detection benchmark datasets to RAWPED and incorporating similar visual structures between them into transfer learning. The overall aim is to integrate training scene-specific detectors to predict RAWPED targets in a structured way, such that transfer learning is robust to wrongly predicted labels while preserving learning efficiency.

CHAPTER THREE
RAILWAY PEDESTRIAN DATASET (RAWPED)

Simultaneous to the advancements in pedestrian detection methods, more challenging datasets are collected for better representation of real life conditions. This is achieved by including diverse and rich set of parameter ranges such as scale, occlusion and pose variation. Each dataset not only brings harder challenges, but also enables discussions on how to evaluate the performance.

Despite extensive amount of ongoing research in pedestrian detection for self-driving cars and surveillance, no dataset or method is introduced for railway systems yet. Thus, the generation of RAWPED aims to determine how well the current detectors work on railway scenes and to observe the failure modes and developing a full system based on the results (Figure 3.1).

The data are collected from the videos acquired during ADORAS project and from public domain sources given in (Selver et. al., 2016). Since most of the video frames do not include any pedestrian, the durations of particular interest such as approaching to level crossings (Figure 3.1.a-d) and stations (Figure 3.1.e-i), rail workers on tracks (Figure 3.1.j-l) or pedestrians walking around railroads (Figure 3.1.m-n) are determined and trimmed.

The video resolutions are between 640x356 and 1920x1080 to allow tests for varying conditions (i.e. low, medium and high resolutions). The medium and low resolution effects are especially critic for pedestrians that are far away from the camera (Figure 3.1.b, 3.1.i, 3.1.l). Since pedestrian detection at low resolution images is reported to have serious problems (Dollar et. al., 2012), testing such cases with improved approaches (Yan, Zhang, Lei, Liao, & Li, 2013; Crete, Dolmiere, Ladret, & Nicolas, 2017) plays an important role in the evaluation phase. Moreover, no special stabilization is used to remove effects of vibration and the overall image quality of the frames is lower than that of still images of comparable resolution. The camera position and viewing angle change slightly depending on the differences of mounting.

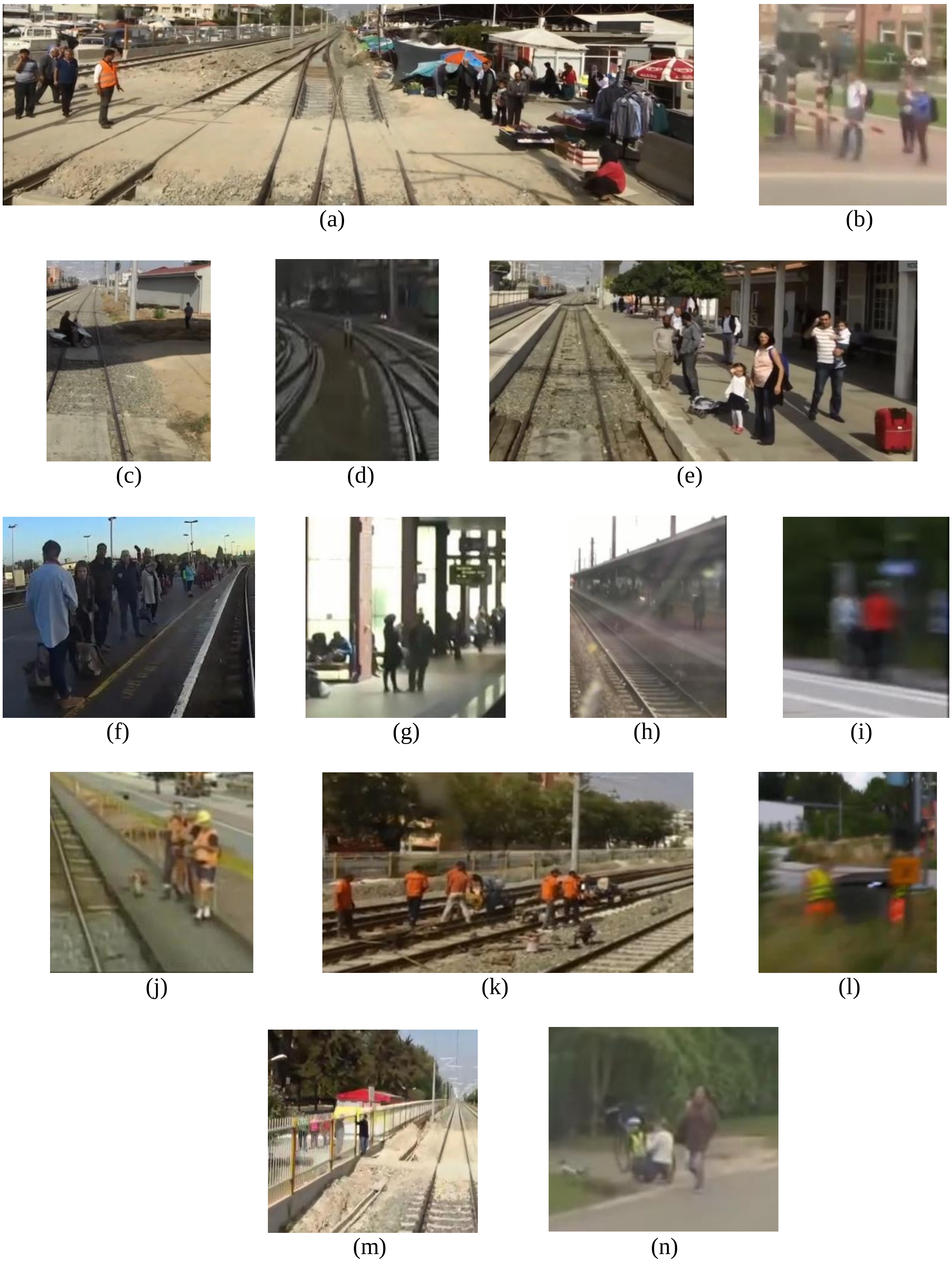


Figure 3.1 Sample frames from RAWPED (Level crossings: (a) a complete scene at near range located besides a bazaar and surrounded by several people waiting to cross (various kinds of occlusions), (b) medium range appearance of pedestrians waiting behind the control bar (low resolution), (c) a pedestrian and a motorcycle at far range (under shadow), (d) a pedestrian between two rail tracks at very far range under low light conditions. Station: (e) a complete scene (near range) (f) effect of camera position on pose and weather on appearance (near range), (g) medium range appearance of pedestrians under artificial lighting, (h) far range appearance affected by inverse illumination and (i) high speed very far range appearance with motion blur. Rail workers: (j) near range low speed, (k) medium range medium speed, (l) long range high speed (motion blur and truncation). (m) Pedestrians at sideways (high speed very far range). (n) Pedestrians at level crossing truncated by security fence (high speed very far range) (PS. The sizes of pedestrians vary because of cropping differences for illustrative purposes))

The frame rates of the cameras are 25, 29, 30 and 50 frames per second (fps), which create varying degrees of quality effects at different speeds. Cameras that operate between 25 and 30 fps are enough for straight railroads and low speed turns (lower than 20 km/h), while even smaller fps is better at low light conditions in order to increase exposure time (Figure 3.1.d). On the other hand, 50 fps or higher is better for high speed (around 100 km/h). Since a different camera with a fixed fps is used for recording of each video, the image quality is degraded due to non-optimal acquisition conditions.

In total, ~26000 frames are annotated in 136 video segments trimmed from 700 minutes long 14 different videos. The labeling is performed using “groundTruthLabeler” application in Matlab, which provides an interactive procedure by manual delineation of a sparse set of frames and automatic prediction at intermediate ones. Figure 3.2 gives information about labeling process.

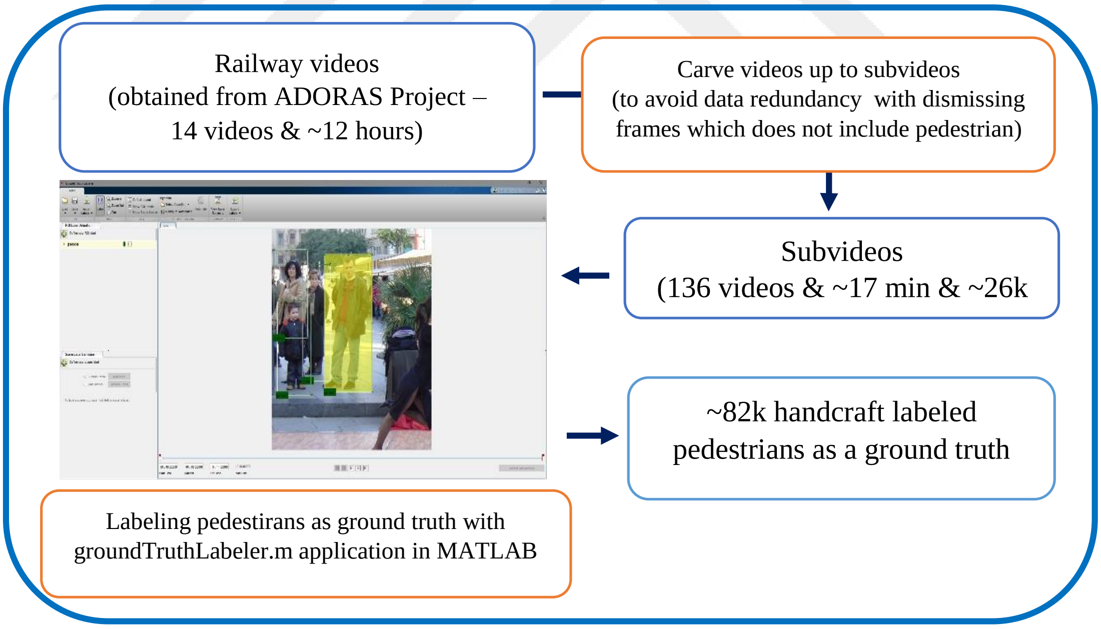

Railway videos
(obtained from ADORAS Project –
14 videos & ~12 hours)
Carve videos up to subvideos
(to avoid data redundancy with dismissing
frames which does not include pedestrian)
Subvideos
(136 videos & ~17 min & ~26k
~82k handcraft labeled
pedestrians as a ground truth
Labeling pedestrians as ground truth with
groundTruthLabeler.m application in MATLAB

Figure 3.2 Labeling process and numerical information about videos, frames and ground truths

The pedestrians are grouped by their image size (height in pixels) for a similar classification with other studies (Dollar et. al., 2012). The range of predefined four scales are selected as near (more than 200 pixels), medium (between 200-100 pixels), far (between 100-50 pixels) and very far (less than 50 pixels), smaller than which reliable annotating is not possible. In Figure 3.3, the histogram of all bounding boxes is given. Cutoffs for the near and far scales result with 62% of the pedestrians to lie in the medium. Figure 3.4 shows the differences between pedestrian appearances in different scales.

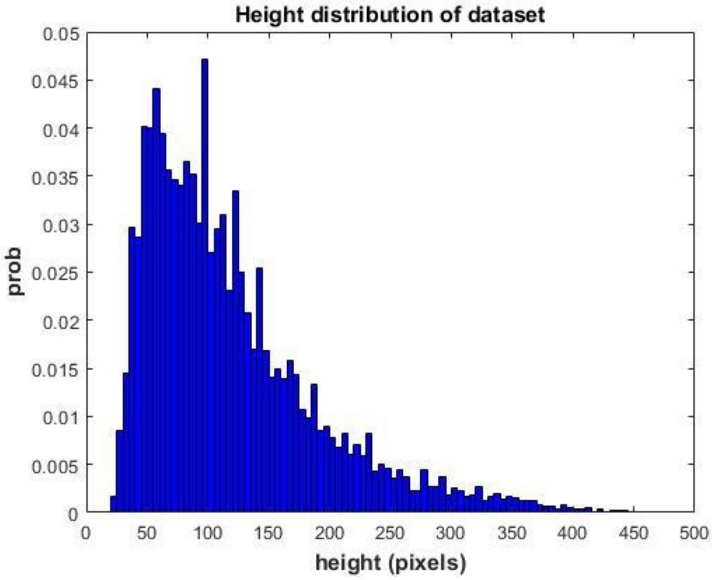
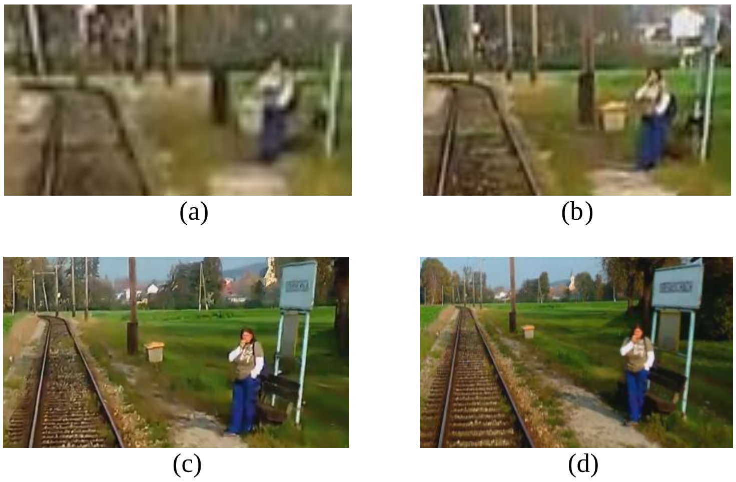


Figure 3.3 Distribution of pedestrian pixel heights (Near scale includes pedestrians over 200 pixels while medium and far scales include 200-80 pixels and under 80 pixels, respectively, and also most observed pedestrians (62%) are at the medium scale)


(a)

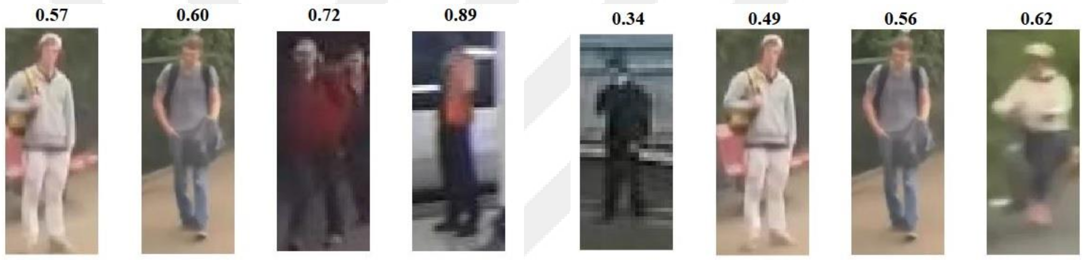
(b)

(c)

(d)

Figure 3.4 Pedestrian appearance (a) very far, (b) far, (c) medium, (d) near ranges for high to low speed (High:70 km/h, low:0 km/h and all images are expanded same size for better comparison)

According to Figures 3.1 and 3.4, blur and contrast properties are important for analyzing, since images has not same resolution and brightness. In order to comparison between benchmark datasets, scale/contrast/blur and score correlations are examined. Examples from RAWPED about contrast, blur values and their effects on appearance of pedestrians are shown in Figure 3.5. Since comparison with a benchmark dataset is necessary to understand results better, Caltech pedestrian dataset is considered in addition to RAWPED. In attempt to get scores for both positive and negative detection bounding boxes, a detector, which is trained as AdaBoost cascades with aggregate channel features extracted from Caltech pedestrian dataset (detailed information is given in Section 4.1), is applied. Figure 3.6 and 3.7 shows the results for Caltech dataset and RAWPED respectively.



a) Contrast

b) Blur

Figure 3.5 Examples for different blur and contrast levels, and the number of on top each image point to value of blur or contrast

As a result of this height/blur/contrast and score analyzes, datasets are observed to be similar characteristics. Even though the detector is trained with Caltech dataset, the distribution of negative and positive detections are similar on both sets. Based on the qualitative observations on the results, the main reasons of false negatives can be categorized into three pedestrian conditions as small size due to range, occlusion and posing from side. Moreover, the effect of range is not only translated to size, but also blur and contrast such that the pedestrians at far usually have low contrast or blurred appearance.

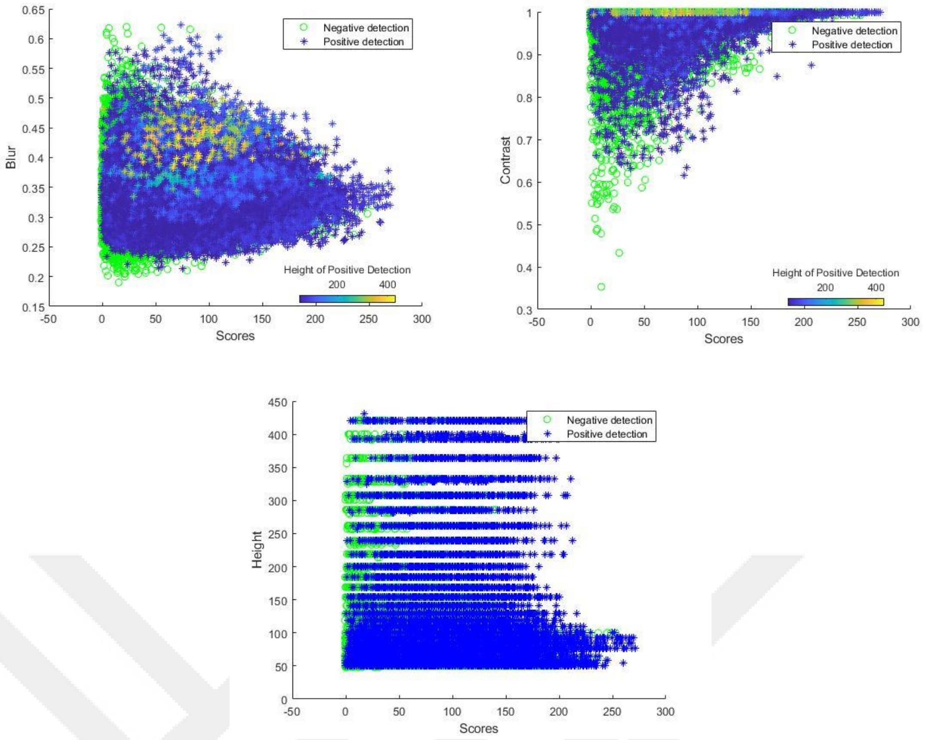


Figure 3.6 Height/blur/contrast and score analyzes for Caltech pedestrian dataset

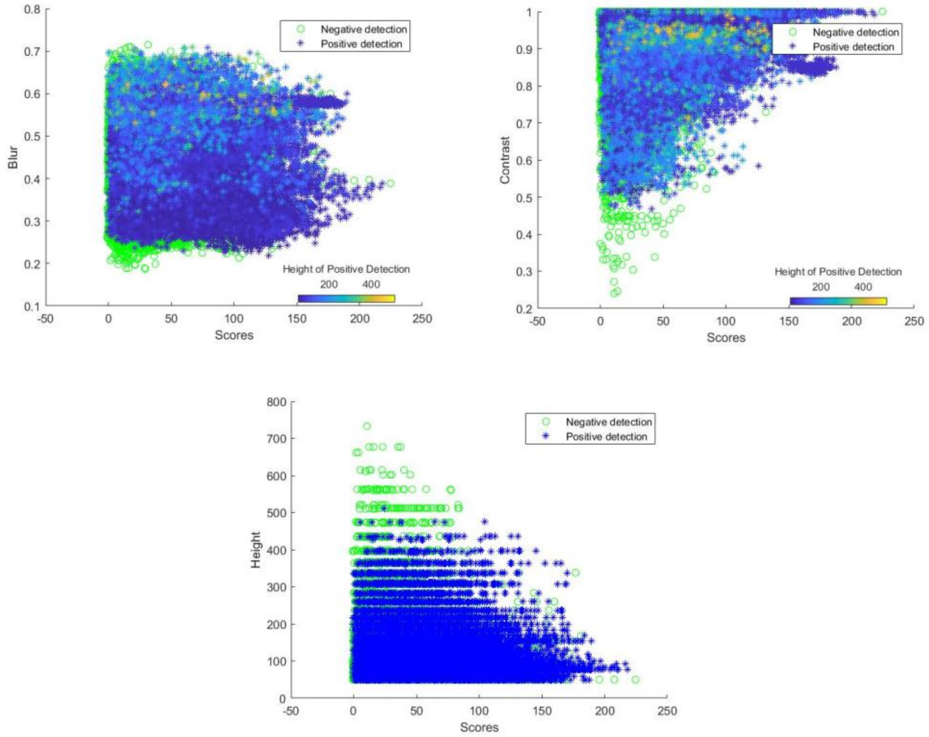


Figure 3.7 Height/blur/contrast and score analyzes for RAWPED

Another important factor that shows the characteristics of the pedestrians in the dataset is log-average aspect ratio of bounding boxes. Since the ratio can significantly change with the pose, its distribution (Figure 3.8) provides information about the variations of pedestrians' appearance and can be used to evaluate the challenging cases of individual detection. The ratio is found to be 0.4 for RAWPED, while it is 0.41 for Caltech, and 0.33, 0.5 for INRIA and KITTI, respectively.

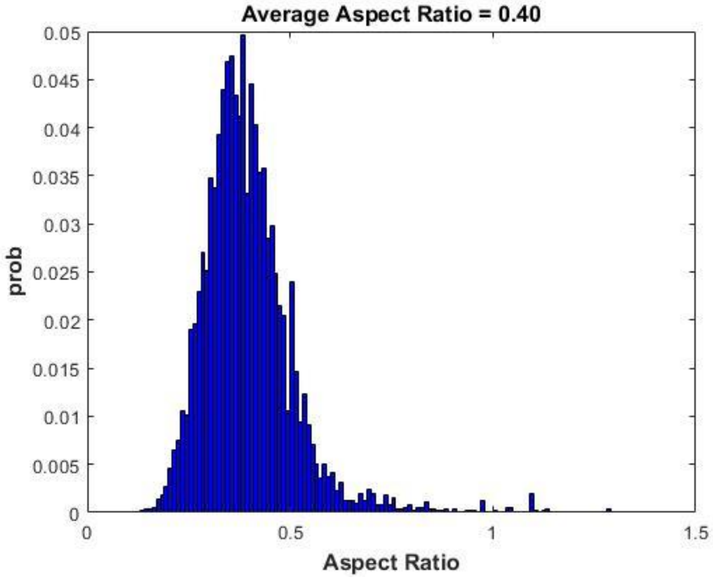


Figure 3.8 Distribution of bounding box aspect ratio

Position statistics of bounding boxes show that the pedestrians are concentrated only in certain regions, which are located around the rail tracks as expected. As can be seen in the heat map plotted based on expected position (Figure 3.9) pedestrians are typically located in two narrow band running vertically across the image corresponding to surroundings of the tracks.

Although train type and mounting positions change, the cameras are always positioned to see the complete front view and therefore, the acquisition constraints are different than the datasets with fixed vehicle-position configuration (such as Caltech (Dollar et. al., 2012)) or the ones containing images obtained from arbitrary viewpoints (such as INRIA (Dalal & Triggs, 2005)).

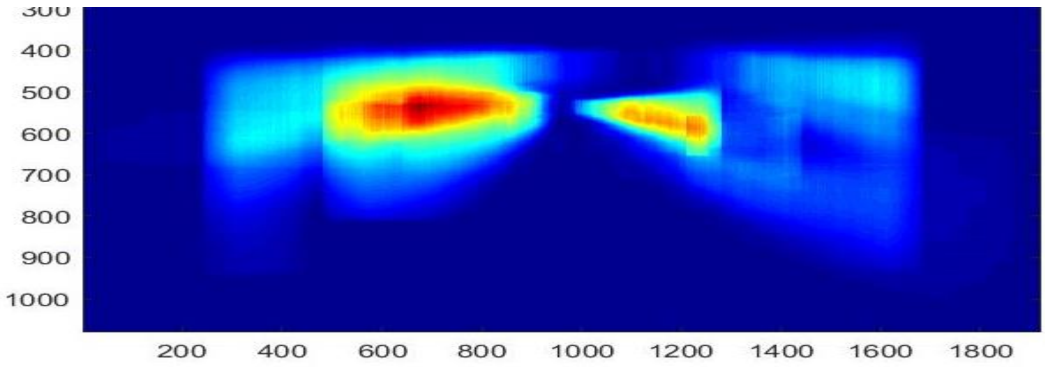


Figure 3.9 Center location of pedestrian bounding boxes for ground truth

Here, it is worth to point that the resolution, frame rate and distance considerations mentioned above should not be considered as the requirements of real systems. In this study, the main focus is given to determine the performance of the current detectors and the proposed system for varying ranges of these parameters. Integration of a detector as an element of a railway on-board driver assistance system needs to take several other properties of the train into account in order to prevent the mismatch between research results and actual system. For instance, detecting near scale pedestrians may leave sufficient time to alert the driver if the train is already slowing down for a station, while even a far scale pedestrian detection may not be enough for a very fast train. Nevertheless, the near scale definition is analyzed throughout this work, because it is important at different sections of a journey. For instance, safety system detection must perform accurately at near/medium scale when approaching a station. Using cameras with higher resolution or zoom can increase the range of detection, but they do not change the actual detector performance for a given scale.

CHAPTER FOUR
PROPOSED PEDESTRIAN DETECTION SYSTEM

Besides continuous efforts for performance improvement in pedestrian detection for self-driving cars and scene specific applications, some of the most recent studies have focused on identifying failure cases, diagnosing reasons behind and providing new insights regarding how to overcome them (Zhang et. al., 2017; Benenson et. al., 2014). In order to translate those experiences to railway on-board driver assistance systems, a series of detailed analyzes are performed on RAWPED and the outcomes construct the foundations of the proposed design. Briefly, the developed system has two main stages (Figure 1.1).

Stage 1: A group of non-deep detectors, which are accurate in localization with computational efficiency, but perform relatively less accurate, are applied to generate candidate bounding boxes (i.e. proposals). Then, their results are combined in order to include as many true positives as possible alongside all candidates (i.e. true positives + false positives). Here, the main drawback of such an approach, increased number of false positives, is handled as a secondary concern and the primary objective is set to minimize false negatives by using complementary properties of diverse features.

Stage 2: Once the disadvantages of well localization and computational burden of deep networks are resolved at the first stage through generation of proposed bounding boxes, CNNs and SVM classifiers highly accurate detection rates can be used to eliminate false positives. Thus, the second stage consists of employing pre-trained CNNs which are fine-tuned to RAWPED scenes via transfer learning, CNNs which are trained from scratch and SVMs that are trained as classifiers for pedestrians.

Thus, from an optimization point of view, the proposed two stage strategy aims to minimize false negatives (low-scores or missing proposals) at the first stage and maximize false positive elimination (background and other objects) within bounding box candidates produced by the first stage at the second stages. The following subsections introduce the detailed analyzes of each stage and present associated analyzes.

4.1 Stage 1: Fusion of Proposals with Non-Deep Detectors

It is experimentally shown by earlier studies that majority of the latest advancements in pedestrian detection can be attributed to the improvement in feature representations rather than the classifier models (Zhang et. al., 2017). Moreover, among the two sources of features, which are implicit generation by learning and explicit extraction via hand-crafting, the latter is observed to dominate the field based on retrospective analysis (Benenson et. al., 2014).

Despite different origins and non-identical implementations of various hand-crafted features, their state-of-the-art results are reported to be significantly close to each other for existing benchmark datasets. On the other hand, in-depth analysis of the results points out complementary outcomes, which motivate the development of fusion strategies. Considering the RAWPED, the need for such a fusion is investigated both quantitatively via true positive – false positive counts and qualitatively by observing bounding boxes at numerous challenging scenes. After extensive experimentation with RAWPED, three models are selected by considering diversity and complementarity.

4.1.1 Aggregate Channel Features (ACF) + AdaBoost Cascades

The structure of the ACF system is based mainly on the extension of channels and has been applied since digital images began to be used. ACF combines gradient histograms and gradient magnitudes with color features and corresponds to an important benchmark in pedestrian detection. The most common color channels are red-green-blue (RGB), hue-saturation-value (HSV) and luminance-chroma-hue (LUV). In addition, different channels can be created using linear or non-linear transformations of the view. The gradient magnitude and gradient histogram extraction from these channels are also generalized as HOG features. Although, many variants are proposed mainly by utilizing different filter types (i.e. square averaging, checkerboards, eigen-vectors from linear discriminant analysis and rotated filters), ACF constitutes a baseline to all. Its computational efficiency is increased by approximations or multiple model use across different scales and neighboring windows.

After extension of channels by appropriate methods, they are transferred to the sub-sample space depending on a predetermined coefficient. All the pixels of the ACF channels that are transmitted to the sub-sample space are transformed into a vector on a look-up table. In this way, the feature vector to be trained with the AdaBoost cascades is obtained.

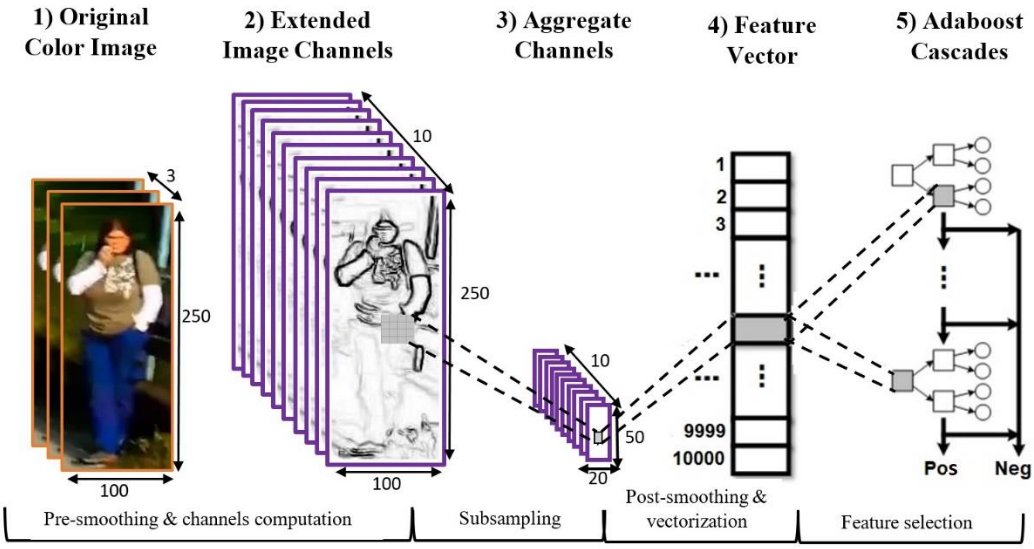


Figure 4.1 Work-flow of ACF + AdaBoost cascades

AdaBoost is one of the popular boosting technique that helps to combine multiple weak classifiers into a strong classifier. Weak classifier performs poorly, but also performs better than random guessing. Usage of cascade version of these classifiers is very practical method for classification problems, but weight of each classifier is important to determine for getting efficient combination. In AdaBoost cascades, AdaBoost determines how much weight should be given to each classifier’s proposed answer when combining their result. The basic scheme of the ACF + AdaBoost system is presented in Figure 4.1.

In this thesis, ACF + AdaBoost cascades model is implemented with use of “detectPeopleACF” function in Matlab. There is two different kind of trained detector as ACF model. One of them is trained with INRIA pedestrian dataset and the classification model of the function have to be set as “inria-100x41”. The other one is trained with Caltech pedestrian dataset and classification model have to be set as “caltech-50x21”. The function have many options which are changable according to applications, but the most important option is “SelectStrongest”. SelectStrongest option can be selected as false for cancelling selection of strongest box from a group, so more detected bounding boxes can be located in image. The results of detections with ACF, the difference between “SelectStrongest” is true or false and also comparison with other detectors will be discussed in next chapter.

4.1.2 Histogram of Oriented Gradients (HOG) + Support Vector Machines (SVM)

Feature extraction using the HOG method is one of the well-known methods that is often used to express the characteristics of an image and training features with SVM in pedestrian detection systems achieve high performance (Li et. al., 2016).

According to this method, a mask, such as (-1, 0, 1), is applied to image in order to find gradients. Subsequently, the image is divided into regions called cell which consist of independent pixels. One dimensional gradient histograms are calculated for each cell region. For each pixel in the cell, the gradient is assigned to the appropriate one of 9 different directions (0-40, ... , 321-360) and each pixel votes in a direction that is proportional to the magnitude of the gradient value. The groups of cells formed in this way are combined to generate blocks. Thus, the histogram of each cell region is normalized by looking at the gradient energies of the other cell regions in the block. The gradient histogram vectors from each block in the image are accumulated to obtain the final feature vector to be used in the classification. The extraction of HOG features belonging to a pedestrian running alongside the rails is presented in Figure 4.2.

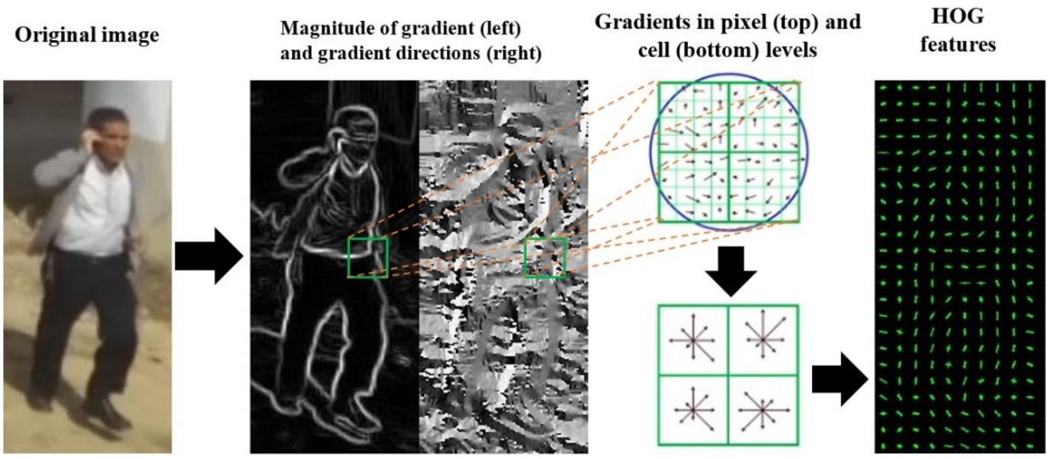

Original image
Magnitude of gradient (left) and gradient directions (right)
Gradients in pixel (top) and cell (bottom) levels
HOG features

Figure 4.2 Feature extraction with HOG method

Support vector machines perform classification by determination of hyperplane that maximizes the margin between two classes. In our case, support vector machine algorithm learns person/non-person classification according to HOG feature vector. Linear SVM is used for the classification procedure, because of there is only two different class and it provides computational efficiency.

In this thesis, HOG + SVM model is implemented with use of “vision.PeopleDetector” function in Matlab. The classification model of the function which is more suitable for problem, is “UprightPeople_96x48”. The function have many options which are changable according to applications, but the important options are “MergeDetections” and “ClassificationThreshold”. MergeDetections option is used for cancelling the merged similar detections, so more detected bounding boxes can be located in image. In case, ClassificationThreshold is the threshold value for decision of person or not. If the threshold is smaller, the more box is labeled as person. The results of detections with HOG, the difference between “MergeDetections” is true or false and also comparison with other detectors will be discussed in next chapter.

4.1.3 Deformable Part Models + Latent SVM

Deformable Part Models (DPM) have recently emerged as a useful and popular tool for conflict with the diversity problem on object detection systems. When objects detect with whole feature map of training model, some kind of position changing situations might be a problem for algorithm. For our problem, standing person and running person has different aspects and according to test model, one of them might be missed.

Basically, deformable part models include histogram of gradient features and the SVM algorithm is utilized for generation of the model. However, differently from HOG + SVM model detection system, DPM has model parts for object to be detected. The parts of model are determined by voting the magnitude and orientation of gradients.

The test part of detection algorithm has two feature maps. Feature map with low resolution is to convolve with whole body model and the other one for parts of model. After convolving the whole body and parts, responses are obtained. The combined score of root locations are determined with adding all responses each other. The work-flow of testing DPM + LSVM algorithm for person is presented in Figure 4.3.

In this thesis, DPM + LSVM model is implemented with use of a function that can be run in Matlab. The DPM function is released by (Felzenszwalb, Girshick, McAllester, & Ramanan, 2010) in gitHub and that includes open source code. The function is trained with four different data sets which are INRIA, Pascal VOC 2006, Pascal VOC 2007 and Pascal VOC 2008. The classification model of the function is chosen as “Pascal VOC 2006” in this thesis. The results of detections with DPM and comparison with other detectors will be discussed in next chapter.

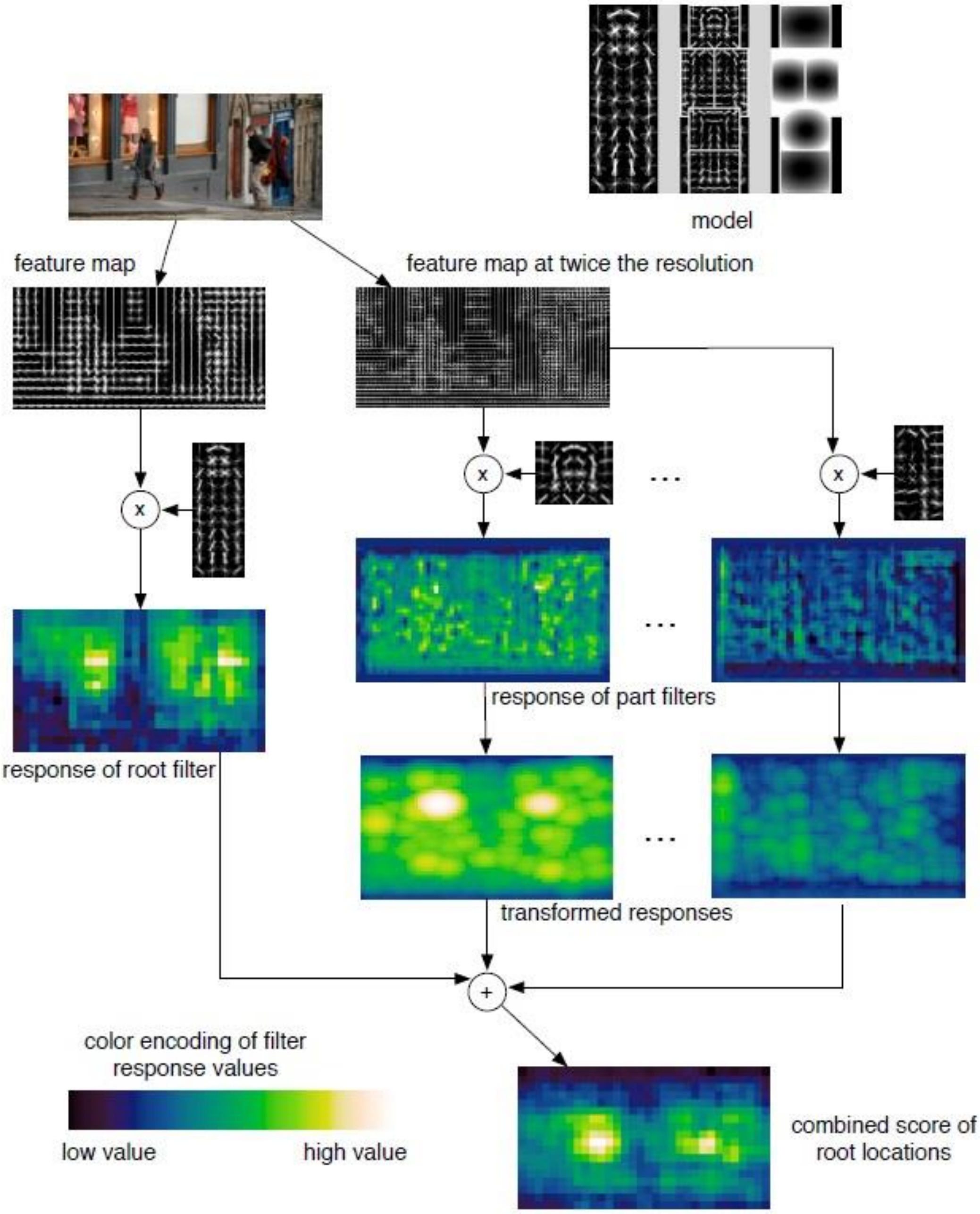


Figure 4.3 Detection steps with DPM + LSVM algorithm (Felzenszwalb et. al., 2010)

4.2 Stage 2: Ensembled Classifiers

The main shortcoming of the hand-crafted features used in Stage 1 is their shape dependency, which makes the classification process vulnerable to similar-shaped non-pedestrian objects. However, the features in deep networks are extracted by network itself and can be learned more specifically depending on deepness of network. Unfortunately, it is still a very challenging and tedious task to train a deep pedestrian detection model which works reliably on all kinds of scenes. The dataset should contain a large diversity of viewpoints, resolutions, scales, and challenges such as illumination conditions, motion blur, occlusions, weather effects and varying backgrounds. Moreover, CNN architecture and parameters should be tweaked many times in order to reach the best performance of the required complex model. It is also not possible to use a state-of-the-art pedestrian detection model trained on another benchmark dataset for railway on-board driver assistance systems since the performance drops significantly.

Therefore, it is much more practical to detect as many as possible pedestrian with ensembled model of non-deep detectors and reduce the false positives with pedestrian classification model which is fine-tuned using the data from RAWPED via transfer learning or which is trained as CNN or SVM classifier from scratch. Even though CNNs are not accurate in pedestrian detection for inadequate datasets, they have reasonable results for classifications.

In the second stage of proposed model, the chosen three classifier will be ensembled for reducing the number of false positives. These subsections give information about theory of each model.

4.2.1 Convolutional Neural Networks (CNNs)

CNNs are very similar to the artificial neural networks in their working principal. However, the main difference between CNNs and artificial neural networks is that CNNs are mainly utilized in the field of pattern recognition. That primary usage provides an opportunity to train CNNs with images.

Another important point in CNNs is that the features are extracted for input images in private. That means, CNNs learn the specific features about that detection/classification problem. The privatization is obtained with convolutional layers, because convolution operation contains calculations about local regions and their weights. The basic architecture of CNN is given in Figure 4.4.

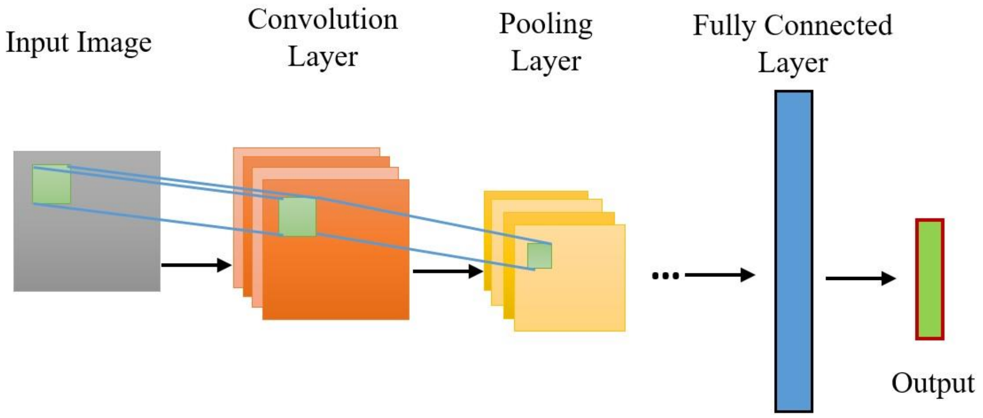


Figure 4.4 Basic architecture of CNNs

Almost all CNNs have same type of layers, but their format, number of layers and sizes of convolutional filters and pooling can differ from each other. This differences are obtained as resultant of experiments which depend on applications.

In this thesis, four different types of pre-trained CNNs are fine-tuned with transfer learning and one CNN is trained from scratch. AlexNet is the first one of the pre-trained CNN models. Actually, it is also first work about popularization about CNN researches. AlexNet was trained with ImageNet dataset which have nearly 14 million images and can classify 1000 different types. The network have only 25 layers and five of them are convolution layers. The difference between typical CNNs and AlexNet is cross channel normalization layers. Cross channel normalization layer replaces each element with their normalized value which is obtained from certain number of neighboring channels. This procedure provides computational simplicity for next layers and also, since the normalization window is determined by network itself, there is no need to change parameters according to applications.

The second pre-trained model, which is fine tuned, is GoogleNet. GoogleNet was also trained with ImageNet. It has 144 layers in total, but the main novelty is not the deepness of the network. The difference between AlexNet and GoogleNet is inception modules of GoogleNet. In typical procedure of CNNs, all convolutional layers are applied respectively, but the model can choose the size of convolutional layers with inception modules. When GoogleNet at inception module, all size of convolutional layers are determined and network picks the best. That module reduces the number of parameters which is 60M in AlexNet but 4M in GoogleNet. Additionally, the network have average pooling before fully connected layer, that helps eliminating a large amount of parameters that do not seem metter much. Hence, GoogleNet is the improved version of AlexNet with its shrunk parameters.

The third fine tuned model is VGG-16 for elimination of false positives. It is also trained with ImageNet and can classify 1000 different types. Although GoogleNet has 144 layers, this total number is determined with inception layers. However, GoogleNet has 12 convolutional layers which are picked from inception layers and are included training. The developers of VGG-16 aim to show the importance of depth of network. Hence, VGG-16 has 16 convolutional layers and extremely homogeneous architecture, because the sizes of convolutional layers are only 2x2 and 3x3. Although VGG-16 performs well, the downside of the network is the large number of parameters (140M) and uses lots of memory.

The last one of the pre-trained model is ResNet which is also trained with ImageNet and can classify 1000 different types like others. Residual network features special skip connections and lots of use of batch normalization layers. Although ResNet has 152 layers in total, because of the normalization layers, has lower complexity if it is compared with VGG-16.

The other CNN which is used in this thesis, is designed with 22 layers and trained from scratch for problem. These layers include 4 convolutional layers which are followed by ReLu and normalization layers. The network can classify only 2 different types which are person and backround. The final result of this CNN classifier might not be enough, but it provides contribution to reduce total number of false positives. The detailed information about results is given in Chapter 6.

4.2.2 Non-Deep SVM Classifier

The support vector machines are utilized for classification problems. Similar to Section 4.1, the classification of bounding boxes as person or not can be applicable. The difference between SVMs that is mentioned in Section 4.1 and this classifier is there is no localization or confidence score. The problem is about decision.

In this thesis, the SVM is trained with bag of visual features. Since the ACF and HOG features are extracted in detection part, SURF features are chosen for classification. SURF features use wavelet responses in horizontal and vertical directions. The mastery orientation is estimated by determination the sum of all responses within a sliding orientation window. Figure 4.5 shows visualization of SURF features for an example pedestrian image from RAWPED. When the SURF features are extracted in Matlab, there is an option to choose orientation as upright, it improves speed and robustness. The “bagOfFeatures” function is used for extraction of SURF features and “trainImageCategoryClassifier” function is used for training SVM with that features.

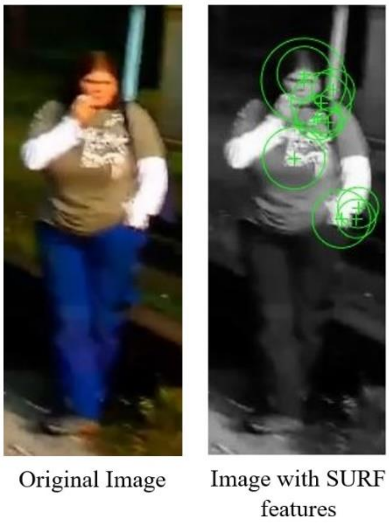
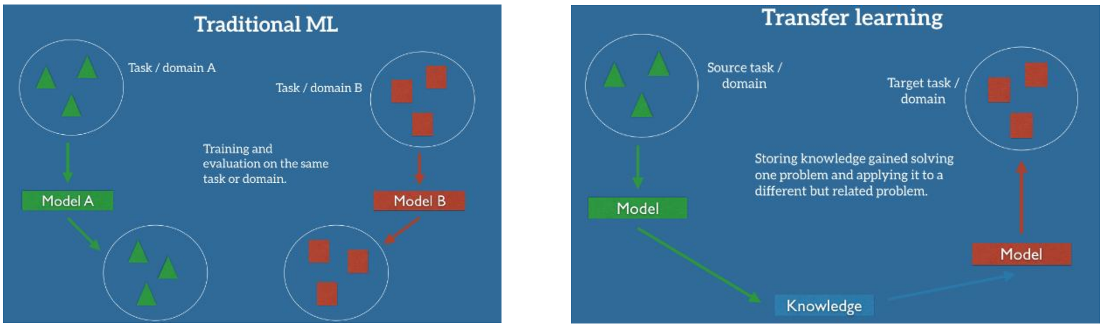


Figure 4.5 Visualization of SURF features

4.2.3 Transfer Learning for Fine-Tuning


Traditional ML
Task / domain A
Task / domain B
Training and evaluation on the same task or domain.
Model A
Model B
Transfer learning
Source task / domain
Target task / domain
Storing knowledge gained solving one problem and applying it to a different but related problem.
Model
Model
Knowledge

(a) (b)

Figure 4.6 Differences between procedures of (a) traditional machine learning and (b) transfer learning

Machine learning algorithms are used for detection or classification of tasks. However, it requires large datasets for better learning which are divided to training and testing parts for algorithms. If the dataset, which needs to be learned for detection or classification, is not adequate for machine learning, transfer learning technique can be used.

Pre-trained models include unspesific features like edges, colors etc. at first layers and application specific features at last layers. The main idea of transfer learning is to retrain the last layers of model with new dataset and to use the knowledge in first layers. Figure 4.6 gives basic information about transfer learning and differences between machine and transfer learnings.

The usage of transfer learning in this thesis, is to fine-tune the pre-trained models which are explained in Section 4.2.2, with RAWPED. For that purpose, the fully connected layers and output layers of these models are changed and retrained. Chapter 6 gives detailed information about their applications and results.

CHAPTER FIVE
EVALUATION METHODOLOGY AND COMPLEMENTARITY OF NON-DEEP DETECTORS

5.1 Evaluation Methodology

Detectors, that have mentioned in Section 4.1, generate bounding boxes and confidence scores as an output. Hence, there must be an evaluation part to understand performance of detectors.

The first step of the evaluation is to decide which bounding box is in true location and detects a person successfully. For that purpose, overlap ratio between detected bounding box and hand-crafted ground truth is determined. According to this ratio value, the bounding box is labeled to be included person or not. However, the threshold value for this classification can be changeable for different applications or expectations. For pedestrian detection applications, threshold value is taken as 0.5. In that case, if the ratio between ground truth and bounding box is smaller than or equal to 0.5, that box is labeled as false positive and the ratio is greater than 0.5, the box is labeled as true positive. There is two other conditions except true positive and false positive. If there is a ground truth but any of bounding boxes does not overlap with it, it is called false negative. The final one is true negative that represents actually all not detected and not labeled as ground truth parts of image. Figure 5.1 shows a scheme for overlap conditions between ground truth and bounding box. In the detection process, number of false negatives and number of false positives are desired to be less. Inversely, number of true positives are desired to be more.

The miss rate term in literature is defined for quantitative representation of these conditions. The formula for miss rate is given in equation 5.1.

$$\mathrm{MR} = 1 - \frac{\text{number of total true positives}}{\text{number of total ground truths}} \qquad (5.1)$$

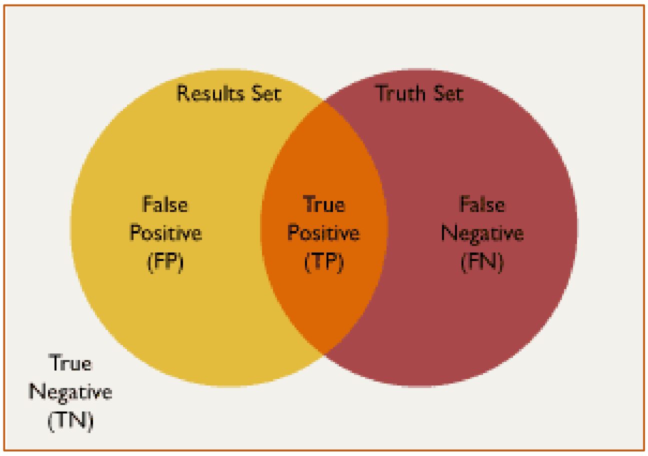


Figure 5.1 Scheme for overlap between ground truth and detected bounding box

The other important metric about evaluation of detection performance is false positives per image (FPPI). Ideally, FPPI is desired to be zero and that means results of detection does not contain false positives. However, false positives are always detected even if they are a little. The formula for FPPI is given in equation 5.2.

$$\mathrm{FPPI} = 1 - \frac{\text{number of total false positives}}{\text{number of total images}} \tag{5.2}$$

Detection algorithms are applied for all images seperately and these numbers (total true positives, total ground truths, total false positives and total images) are increased image by image, because the calculation is done for not only one image but also all images until then. For observing the progress in every iteration, all of numbers are determined and kept in a vector. Following these calculations for all images, the “miss rate – FPPI” graph can be plotted.

In this thesis, the “evaluateDetectionMissRate” function in Matlab is used to obtain the graph. Figure 5.2 shows six chosen example images for better understanding of evaluation methodology. These images are taken from Caltech pedestrian dataset. The blue boxes represent results of detector and the yellow boxes are ground truths.

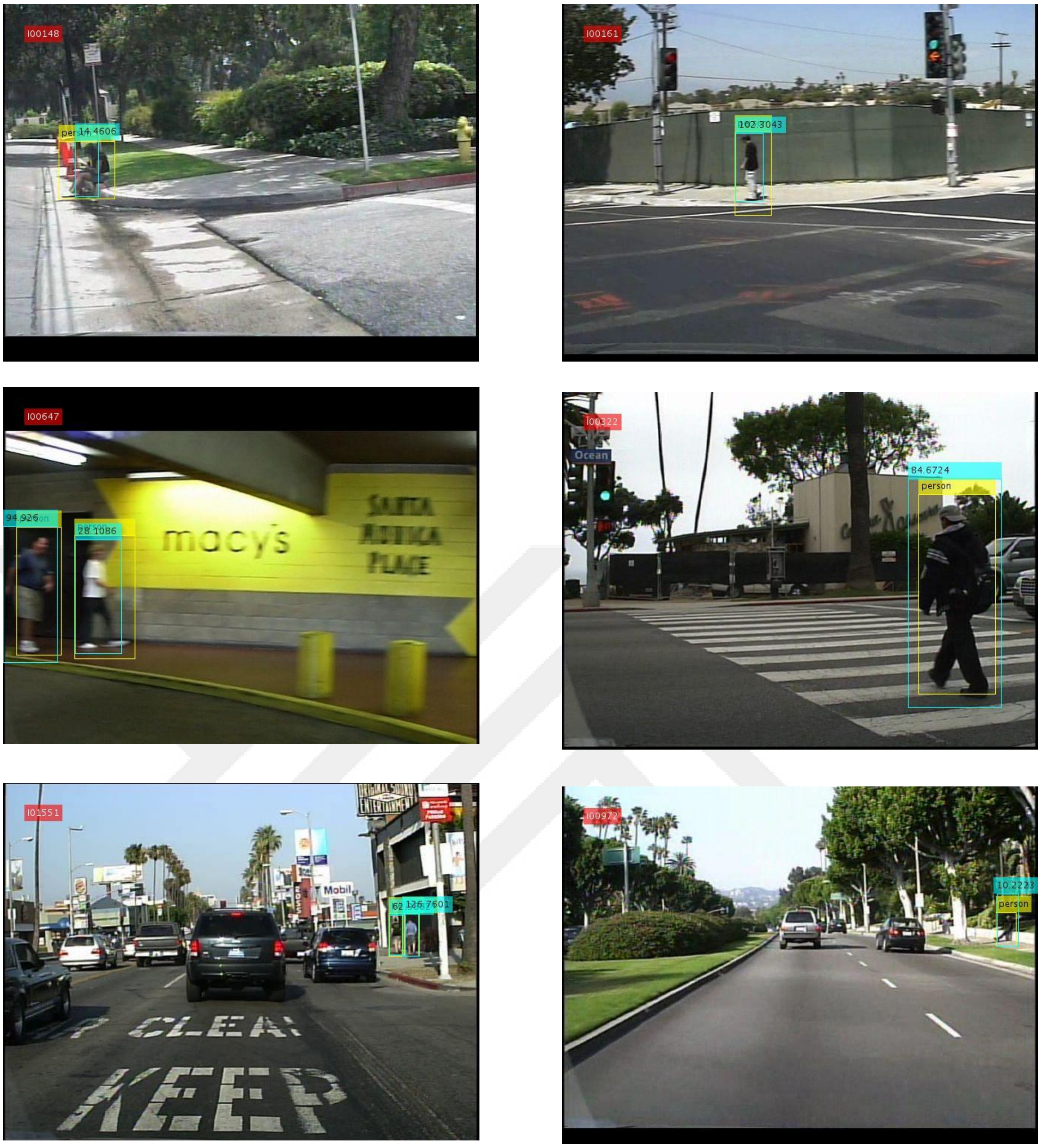


Figure 5.2 Example labeled images from Caltech pedestrian dataset

According to the all labels, that can be mentioned as all pedestrians are detected. Hence, this assumption needs to be proved with quantitative metrics.

“evaluateDetectionMissRate” checks the overlap between ground truths and detected bounding boxes, if the box is true positive, that bounding box is labeled as 1 and if the box is false positive, that bounding box is labeled as 0. Detected bounding boxes and ground truths are given as input for the function and labels are determined as equation 5.3.

$$\text{labels} = [1; 1; 1; 1; 1; 1; 0; 1] \tag{5.3}$$

Even though there are six example images, the length of labels vector is eight because it is equal to the number of detected bounding boxes. The next step, after getting labels, is generation of true and false positive vectors.

$$\text{tp} = \text{labels} > 0 \tag{5.4}$$
$$\text{tp} = [1; 1; 1; 1; 1; 1; 0; 1]; \tag{5.5}$$
$$\text{fp} = \text{labels} \leq 0 \tag{5.6}$$
$$\text{fp} = [0; 0; 0; 0; 0; 0; 1; 0]; \tag{5.7}$$

Equations 5.4 and 5.6 represents the code versions of calculations. The results of these calculations are given in equations 5.5 and 5.7. These vectors contain information about each box seperately. However, the formulas of FPPI and miss rate include the total numbers for every iteration. The cumulative sum operaion is sufficent for that purpose. Equations 5.8 to 5.11 show the codes and their results.

$$\text{tp} = \text{cumsum(tp)}; \tag{5.8}$$
$$\text{tp} = [1; 2; 3; 4; 5; 6; 6; 7]; \tag{5.9}$$
$$\text{fp} = \text{cumsum(fp)}; \tag{5.10}$$
$$\text{fp} = [0; 0; 0; 0; 0; 0; 1; 1]; \tag{5.11}$$

Except total number of false and true positives, the total number of images and total number of ground truths are necessary. These numbers do not change in different iterations. Hence, they are taken as constants. Total number of images is six as mentioned before and total number of ground truths is eight. Miss rate and FPPI vectors can be determined according to equations 5.1 and 5.2.

$$\text{mr} = [0.875; 0.75; 0.625; 0.5; 0.375; 0.25; 0.25; 0.125]; \tag{5.12}$$
$$\text{fppi} = [0; 0; 0; 0; 0; 0; 0.1667; 0.1667]; \tag{5.13}$$

The miss rate – FPPI graph is plotted in logarithmic axises theorically. Since the results of these determination are real as shown in equation 5.12 and 5.13, they have to be referred in logarithmic scale. For that aim, there is reference points in literature as written in equation 5.14.

ref = [0.01; 0.01778; 0.0316; 0.0562; 0.1; 0.17782; 0.316; 0.5623; 1];   (5.14)

The reference points are compared with FPPI values in every iteration. The index of the last FPPI value, which is smaller than or equal to the reference point in that iteration, is taken and the value of the miss rate at that index is written over value of reference. When all iterations are determined, the reference vector is changed to values of miss rate which depend on variation of the FPPI values and it is shown in equation 5.15.

ref = [0.25; 0.25; 0.25; 0.25; 0.25; 0.125; 0.125; 0.125; 0.125];   (5.15)

The last term for evaluation is log-average miss rate value which can be determined with equation 5.16 and its result in equation 5.17. The log-average miss rate (LAMR) is very important for comparison results of methods. The smallest LAMR value is better and that means the method detects more true positives and less false positives.

amr = exp(mean(log(ref)));   (5.16)
amr = 0.1837   (5.17)

In this thesis, the comparisons are discussed with plotting miss rate – FPPI graphs as Figure 5.3 and with determination of LAMR values.

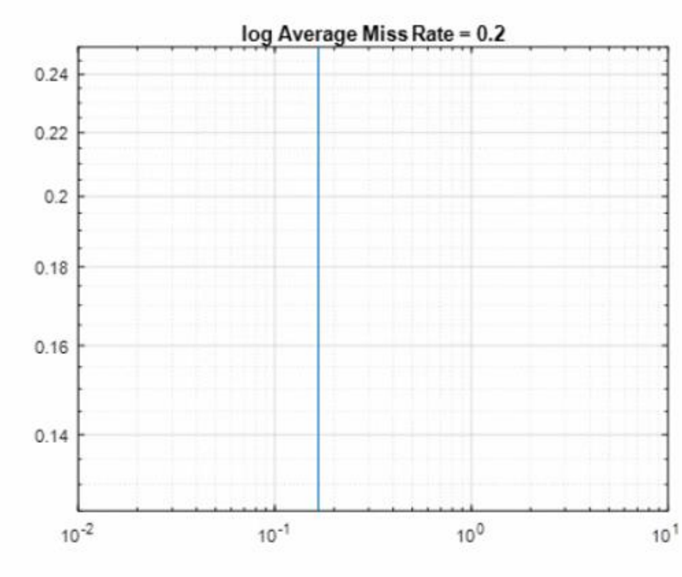


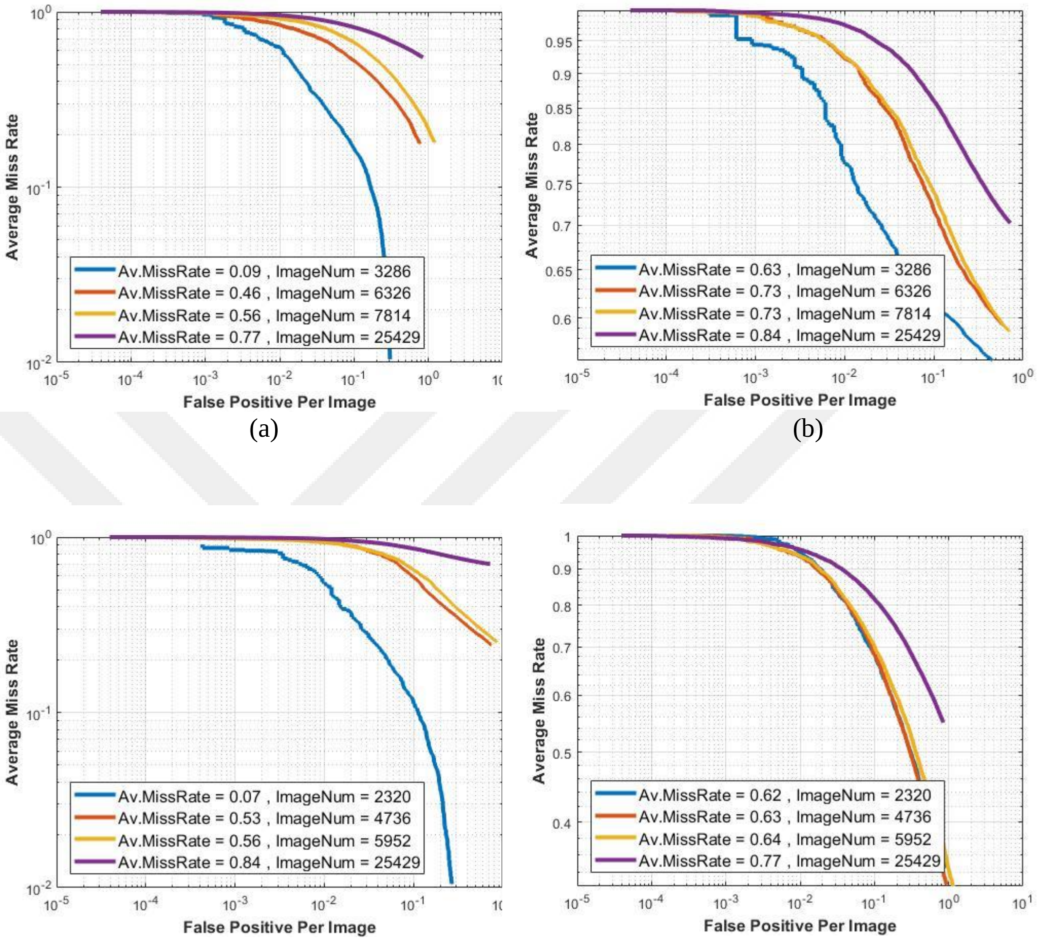
Figure 5.3 Miss rate – FPPI graph for example images

5.2 Complementarity of Non-Deep Detectors

Both complementarity and inter-usability of the mentioned models in Section 4.1 are important factors for covering the whole solution space without false negatives. Moreover, it provides an insight about the generalization capability of the systems. The previous studies indicate that the detectors, which complete their training with a dataset that is different than the test set, can not perform enough accuracy. This lack of generalization is observed to be related with diversity of the pedestrians in a dataset rather than number of samples.

The first part of the proposed model assumes that detection algorithms complement each other and number of false negatives are decreased with respect to their individual versions. For proving this thesis, first of all, detection algorithms have to be examined seperately. Table 5.1 shows detection results for each model. LAMR values of all models are very high and that means all of them are inadequate for railway on-board driver assistance systems.

The direct use of non-deep detectors in railway images, which are trained with the effectiveness datasets in pedestrian detection, produces unsuccessful results. At this point, an important experiment was performed for ACF+AdaBoost systems to see the effect of the dataset on which the model is trained. The RAWPED dataset is divided by the performance of the algorithm instead of the increasing difficulty level. For example, the score values, in the application result of the Caltech-trained model, are sorted by values and first quarter with high score is formed as first set. Four subsets of RAWPED were reorganized in this manner. Then, both the Caltech-trained and INRIA-trained model were applied to these groups (Figure 5.4.a-b). Similarly, they were applied to the sets that were reorganized according to the performance of the INRIA-trained model (Figure 5.4.c-d). Pursuant to obtained results, a model has inability to achieve same performance on the set where the performance of another model is high, and even this performance remains at a very low level. Different methods, different models and even similar models trained with different datasets provides good performance for different parts of the same dataset. This ensures high diversity results and shows that the results, that will occur when different systems are used as an ensemble, are potentially complementary.



(a)

(b)

(c)

(d)

Figure 5.4 (a) Caltech-trained model and (b) INRIA-trained model test results for RAWPED groups which reorganized with Caltech-trained model, (c) INRIA-trained model and (scrd) Caltech-trained model test results for RAWPED groups which reorganized with INRIA-trained model

Table 5.1 The detection results for each detector seperately (“Miss” is the number of not detected ground truth, “Hit” is number of detected ground truth and the summation of them is constant for all and equal to 81481 which is also total number of ground truth)

<table>
<tr><th colspan="5">DPM</th></tr>
<tr><td>Miss</td><td>Hit</td><td>True Positives</td><td>False Positives</td><td>LAMR</td></tr>
<tr><td>61510</td><td>19971</td><td>20273</td><td>15632</td><td>0.86</td></tr>
<tr><th colspan="5">ACF (w/ Caltech) – SelectStrongest True</th></tr>
<tr><td>Miss</td><td>Hit</td><td>True Positives</td><td>False Positives</td><td>LAMR</td></tr>
<tr><td>53355</td><td>28126</td><td>28485</td><td>29328</td><td>0.85</td></tr>
<tr><th colspan="5">ACF (w/ Caltech) – SelectStrongest False</th></tr>
<tr><td>Miss</td><td>Hit</td><td>True Positives</td><td>False Positives</td><td>LAMR</td></tr>
<tr><td>49483</td><td>31998</td><td>171774</td><td>214214</td><td>0.96</td></tr>
<tr><th colspan="5">ACF (w/ Inria) – SelectStrongest True</th></tr>
<tr><td>Miss</td><td>Hit</td><td>True Positives</td><td>False Positives</td><td>LAMR</td></tr>
<tr><td>56787</td><td>24694</td><td>24856</td><td>18493</td><td>0.84</td></tr>
<tr><th colspan="5">ACF (w/ Inria) – SelectStrongest False</th></tr>
<tr><td>Miss</td><td>Hit</td><td>True Positives</td><td>False Positives</td><td>LAMR</td></tr>
<tr><td>53791</td><td>27690</td><td>520705</td><td>278772</td><td>0.98</td></tr>
<tr><th colspan="5">HOG (Upright 96x48) – MergeDetections True – Class. Thresh. = 1</th></tr>
<tr><td>Miss</td><td>Hit</td><td>True Positives</td><td>False Positives</td><td>LAMR</td></tr>
<tr><td>57942</td><td>23539</td><td>23697</td><td>256614</td><td>0.97</td></tr>
<tr><th colspan="5">HOG (Upright 96x48) – MergeDetections False – Class. Thresh. = 1</th></tr>
<tr><td>Miss</td><td>Hit</td><td>True Positives</td><td>False Positives</td><td>LAMR</td></tr>
<tr><td>45551</td><td>35930</td><td>256126</td><td>1229947</td><td>0.96</td></tr>
<tr><th colspan="5">HOG (Upright 96x48) – MergeDetections False – Class. Thresh. = 0</th></tr>
<tr><td>Miss</td><td>Hit</td><td>True Positives</td><td>False Positives</td><td>LAMR</td></tr>
<tr><td>33966</td><td>47515</td><td>744928</td><td>10964269</td><td>0.96</td></tr>
</table>

In addition to inadequacy for problem, there is an obscurity whether hitted bounding boxes of detection models are same or not. Measurement of the diversity of the individual methods can be analyzed in two subsection. The first one is the pairwise diversity measurement. That measurement technique works on only two different method for comparison and have some different types in itself. However, the Q statistics is handled for 2-by-2 comparison of methods in this thesis (Yule, 1900). Table 5.2 shows the technique and equation 5.18 gives detailed information about determination of Q statistic.

Table 5.2 2-by-2 table for relationship between two detectors

	Detector$_1$ Hit	Detector$_1$ Miss
Detector$_2$ Hit	N^{11}	N^{10}
Detector$_2$ Miss	N^{01}	N^{00}

$$Q = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \tag{5.18}$$

According to equation 5.18 and Table 5.2, the all combinations of the detectors, which are mentioned in Section 4.1, are observed and their Q values are determined. The equation 5.18 shows that if Q is equal to 1, the detectors are same with each other. The results can be interpreted in light of this information. The Tables 5.3 to 5.25 gives the details.

Table 5.3 The diversity between DPM and ACF trained with Caltech dataset when “SelectStrongest” is true

DPM & ACF (w/ Caltech) – SelectStrongest True		
	DPM Hit	DPM Miss
ACF Hit	11516	16610
ACF Miss	8455	44900
Q = 0.573		

Table 5.4 The diversity between DPM and ACF trained with Caltech dataset when “SelectStrongest” is false

DPM & ACF (w/ Caltech) – SelectStrongest False		
	DPM Hit	DPM Miss
ACF Hit	12457	19541
ACF Miss	7514	41969
Q = 0.561		

Table 5.5 The diversity between DPM and ACF trained with INRIA dataset when “SelectStrongest” is true

DPM & ACF (w/ Inria) – SelectStrongest True		
	DPM Hit	DPM Miss
ACF Hit	14246	10448
ACF Miss	5725	51062
Q = 0.848		

Table 5.6 The diversity between DPM and ACF trained with INRIA dataset when “SelectStrongest” is false

DPM & ACF (w/ Inria) – SelectStrongest False		
	DPM Hit	DPM Miss
ACF Hit	14923	12767
ACF Miss	5048	48743
Q = 0.837		

Table 5.7 The diversity between DPM and HOG “UprightPeople_96x48” when “MergeDetections” is true and “ClassificationThreshold” is 1

DPM & HOG (Upright 96x48) – MergeDetections True – Class. Thresh. = 1		
	DPM Hit	DPM Miss
HOG Hit	11033	12506
HOG Miss	8938	49004
Q = 0.657		

Table 5.8 The diversity between DPM and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 0

DPM & HOG (Upright 96x48) – MergeDetections False – Class. Thresh. = 0		
	DPM Hit	DPM Miss
HOG Hit	18547	28968
HOG Miss	1424	32542
Q = 0.872		

Table 5.9 The diversity between DPM and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 1

DPM & HOG (Upright 96x48) – MergeDetections False – Class. Thresh. = 1		
	DPM Hit	DPM Miss
HOG Hit	16528	19402
HOG Miss	3443	42108
Q = 0.810		

Table 5.10 The diversity between ACF trained with Caltech dataset when its “SelectStrongest” option is true and ACF trained with INRIA dataset when its “SelectStrongest” option is true

ACF (w/ Caltech) – SelectStrongest True & ACF (w/ Inria) – SelectStrongest True		
	ACF(w/ Caltech) Hit	ACF(w/ Caltech) Miss
ACF(w/ Inria) Hit	14263	10431
ACF(w/ Inria) Miss	13863	42924
Q = 0.618		

Table 5.11 The diversity between ACF trained with Caltech dataset when its “SelectStrongest” option is true and ACF trained with INRIA dataset when its “SelectStrongest” option is false

ACF (w/ Caltech) – SelectStrongest True & ACF (w/ Inria) – SelectStrongest False		
	ACF(w/ Caltech) Hit	ACF(w/ Caltech) Miss
ACF(w/ Inria) Hit	15095	12595
ACF(w/ Inria) Miss	13031	40760
Q = 0.579		

Table 5.12 The diversity between ACF trained with Caltech dataset when its “SelectStrongest” option is false and ACF trained with INRIA dataset when its “SelectStrongest” option is true

<table>
<tr><th colspan="3">ACF (w/ Caltech) – SelectStrongest False
& ACF (w/ Inria) – SelectStrongest True</th></tr>
<tr><td></td><td>ACF(w/ Caltech) Hit</td><td>ACF(w/ Caltech) Miss</td></tr>
<tr><td>ACF(w/ Inria) Hit</td><td>15933</td><td>8761</td></tr>
<tr><td>ACF(w/ Inria) Miss</td><td>16065</td><td>40722</td></tr>
<tr><td colspan="3"><b>Q = 0.643</b></td></tr>
</table>

Table 5.13 The diversity between ACF trained with Caltech dataset when its “SelectStrongest” option is false and ACF trained with INRIA dataset when its “SelectStrongest” option is false

<table>
<tr><th colspan="3">ACF (w/ Caltech) – SelectStrongest False
& ACF (w/ Inria) – SelectStrongest False</th></tr>
<tr><td></td><td>ACF(w/ Caltech) Hit</td><td>ACF(w/ Caltech) Miss</td></tr>
<tr><td>ACF(w/ Inria) Hit</td><td>17280</td><td>10410</td></tr>
<tr><td>ACF(w/ Inria) Miss</td><td>14718</td><td>39073</td></tr>
<tr><td colspan="3"><b>Q = 0.630</b></td></tr>
</table>

Table 5.14 The diversity between ACF trained with Caltech dataset when its “SelectStrongest” option is true and HOG “UprightPeople_96x48” when “MergeDetections” is true and “ClassificationThreshold” is 1

<table>
<tr><th colspan="3">ACF (w/ Caltech) – SelectStrongest True &
HOG (Upright 96x48) – MergeDetections True – Class. Thresh. = 1</th></tr>
<tr><td></td><td>ACF(w/ Caltech) Hit</td><td>ACF(w/ Caltech) Miss</td></tr>
<tr><td>HOG Hit</td><td>10931</td><td>12608</td></tr>
<tr><td>HOG Miss</td><td>17195</td><td>40747</td></tr>
<tr><td colspan="3"><b>Q = 0.345</b></td></tr>
</table>

Table 5.15 The diversity between ACF trained with Caltech dataset when its “SelectStrongest” option is true and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 0

<table>
<tr><th colspan="3">ACF (w/ Caltech) – SelectStrongest True &
HOG (Upright 96x48) – MergeDetections False – Class. Thresh. = 0</th></tr>
<tr><td></td><td>ACF(w/ Caltech) Hit</td><td>ACF(w/ Caltech) Miss</td></tr>
<tr><td>HOG Hit</td><td>18632</td><td>28883</td></tr>
<tr><td>HOG Miss</td><td>9494</td><td>24472</td></tr>
<tr><td colspan="3"><b>Q = 0.249</b></td></tr>
</table>

Table 5.16 The diversity between ACF trained with Caltech dataset when its “SelectStrongest” option is true and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 1

<table>
<tr><th colspan="3">ACF (w/ Caltech) – SelectStrongest True &
HOG (Upright 96x48) – MergeDetections False – Class. Thresh. = 1</th></tr>
<tr><td></td><td>ACF(w/ Caltech) Hit</td><td>ACF(w/ Caltech) Miss</td></tr>
<tr><td>HOG Hit</td><td>16439</td><td>19491</td></tr>
<tr><td>HOG Miss</td><td>11687</td><td>33864</td></tr>
<tr><td colspan="3"><b>Q = 0.419</b></td></tr>
</table>

Table 5.17 The diversity between ACF trained with Caltech dataset when its “SelectStrongest” option is false and HOG “UprightPeople_96x48” when “MergeDetections” is true and “ClassificationThreshold” is 1

<table>
<tr><th colspan="3">ACF (w/ Caltech) – SelectStrongest False &
HOG (Upright 96x48) – MergeDetections True – Class. Thresh. = 1</th></tr>
<tr><td></td><td>ACF(w/ Caltech) Hit</td><td>ACF(w/ Caltech) Miss</td></tr>
<tr><td>HOG Hit</td><td>11957</td><td>11582</td></tr>
<tr><td>HOG Miss</td><td>20041</td><td>37901</td></tr>
<tr><td colspan="3"><b>Q = 0.322</b></td></tr>
</table>

Table 5.18 The diversity between ACF trained with Caltech dataset when its “SelectStrongest” option is false and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 0

<table>
<tr><th colspan="3">ACF (w/ Caltech) – SelectStrongest False &
HOG (Upright 96x48) – MergeDetections False – Class. Thresh. = 0</th></tr>
<tr><td></td><td>ACF(w/ Caltech) Hit</td><td>ACF(w/ Caltech) Miss</td></tr>
<tr><td>HOG Hit</td><td>21144</td><td>26371</td></tr>
<tr><td>HOG Miss</td><td>10854</td><td>23112</td></tr>
<tr><td colspan="3"><b>Q = 0.261</b></td></tr>
</table>

Table 5.19 The diversity between ACF trained with Caltech dataset when its “SelectStrongest” option is false and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 1

<table>
<tr><th colspan="3">ACF (w/ Caltech) – SelectStrongest False &
HOG (Upright 96x48) – MergeDetections False – Class. Thresh. = 1</th></tr>
<tr><td></td><td>ACF(w/ Caltech) Hit</td><td>ACF(w/ Caltech) Miss</td></tr>
<tr><td>HOG Hit</td><td>18301</td><td>17629</td></tr>
<tr><td>HOG Miss</td><td>13697</td><td>31854</td></tr>
<tr><td colspan="3"><b>Q = 0.414</b></td></tr>
</table>

Table 5.20 The diversity between ACF trained with INRIA dataset when its “SelectStrongest” option is true and HOG “UprightPeople_96x48” when “MergeDetections” is true and “ClassificationThreshold” is 1

<table>
<tr><th colspan="3">ACF (w/ Inria) – SelectStrongest True &
HOG (Upright 96x48) – MergeDetections True – Class. Thresh. = 1</th></tr>
<tr><td></td><td>ACF(w/ Inria) Hit</td><td>ACF(w/ Inria) Miss</td></tr>
<tr><td>HOG Hit</td><td>12513</td><td>11026</td></tr>
<tr><td>HOG Miss</td><td>12181</td><td>45761</td></tr>
<tr><td colspan="3"><b>Q = 0.620</b></td></tr>
</table>

Table 5.21 The diversity between ACF trained with INRIA dataset when its “SelectStrongest” option is true and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 0

<table>
<tr><th colspan="3">ACF (w/ Inria) – SelectStrongest True &
HOG (Upright 96x48) – MergeDetections False – Class. Thresh. = 0</th></tr>
<tr><td></td><td>ACF(w/ Inria) Hit</td><td>ACF(w/ Inria) Miss</td></tr>
<tr><td>HOG Hit</td><td>23010</td><td>24505</td></tr>
<tr><td>HOG Miss</td><td>1684</td><td>32282</td></tr>
<tr><td colspan="3"><b>Q = 0.895</b></td></tr>
</table>

Table 5.22 The diversity between ACF trained with INRIA dataset when its “SelectStrongest” option is true and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 1

ACF (w/ Inria) – SelectStrongest True & HOG (Upright 96x48) – MergeDetections False – Class. Thresh. = 1		
	ACF(w/ Inria) Hit	ACF(w/ Inria) Miss
HOG Hit	19564	16366
HOG Miss	5130	40421
Q = 0.808		

Table 5.23 The diversity between ACF trained with INRIA dataset when its “SelectStrongest” option is false and HOG “UprightPeople_96x48” when “MergeDetections” is true and “ClassificationThreshold” is 1

ACF (w/ Inria) – SelectStrongest False & HOG (Upright 96x48) – MergeDetections True – Class. Thresh. = 1		
	ACF(w/ Inria) Hit	ACF(w/ Inria) Miss
HOG Hit	13307	10232
HOG Miss	14383	43559
Q = 0.595		

Table 5.24 The diversity between ACF trained with INRIA dataset when its “SelectStrongest” option is false and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 0

ACF (w/ Inria) – SelectStrongest False & HOG (Upright 96x48) – MergeDetections False – Class. Thresh. = 0		
	ACF(w/ Inria) Hit	ACF(w/ Inria) Miss
HOG Hit	25275	22240
HOG Miss	2415	31551
Q = 0.874		

Table 5.25 The diversity between ACF trained with INRIA dataset when its “SelectStrongest” option is false and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 1

ACF (w/ Inria) – SelectStrongest False & HOG (Upright 96x48) – MergeDetections False – Class. Thresh. = 1		
	ACF(w/ Inria) Hit	ACF(w/ Inria) Miss
HOG Hit	21173	14757
HOG Miss	6517	39034
Q = 0.791		

As a result of all these tables, the assumption of diversity of detectors is proved, they can complement each other. The number of total missed ground truth is always smaller than both of their individual versions. The Q statistic values are support this assumption. However, some versions of this comparison are more diverse than others according to value of Q. For example in Table 5.18, value of Q and number of total missed ground truth is small. Ensemble of these two detector types might seem to be accurate. Except that, the number of total false positives is very important for efficacious detection. Number of total false positives is nearly 11.5 million for that example. That number is very-high and this highness reduces accuracy of detection. Hence, the point to consider about the ensemble of detectors is not only number of missed ground truth but also highness of number of false positives.

The other important lesson which is taken from tables, is that changing the options of detectors in Matlab effects results. The less number of total missed ground truth is observed when “SelectStrongest” and “MergeDetections” are false. Hence, the ensembled version of detectors might be set to false. However, there is four different detectors and pairwise analyzes only provide information about ensemble of two detectors.

The non-pairwise analyzes are also useful for ensembled system. The κ can be used when raters asses subjects to measure the level of agreement and detectors are different raters for this problem (Fleiss, 1981). The formula for κ is given in equation 5.19.

$$\kappa = 1 - \frac{\frac{1}{\mathrm{L}} \sum_{j=1}^{N} l(z_j)\,(L - l(z_j))}{\mathrm{N(L-1)}\bar{\rho}(1-\bar{\rho})} \tag{5.19}$$

$$l(z_j) = \sum_{i=1}^{L} y_{j,i} \tag{5.20}$$

$$\bar{\rho} = \frac{1}{\mathrm{NL}} \sum_{j=1}^{N} l(z_j) \tag{5.21}$$

The $y_{j,i}$ term means that ith detector detects jth ground truth or not. If detection result is labeled as true positive $y_{j,i}$ becomes 1 otherwise $y_{j,i}$ becomes 0. This y values are summed for all detectors and $l(z_j)$ term is observed as equation 5.20. The $\bar{\rho}$ term means the average of indivudial classification accuracy for all detectors and ground truths as shown in equation 5.21. N is equal to number of total ground truths and L is equal to number of detectors. In the light of this informations, the κ is determined for different versions of ensembled systems. The details about observations are given in from Tables 5.26 to 5.32.

Table 5.26 The diversity between DPM, ACF trained with Caltech when "SelectStrongest" is true, ACF trained with INRIA when "SelectStrongest" is true and HOG "UprightPeople_96x48" when "MergeDetections" is true and "ClassificationThreshold" is 1

		ACF (Caltech) Miss	**ACF (Inria) Miss**	**ACF (Caltech) Miss**	**ACF (Inria) Hit**	**ACF (Caltech) Hit**	**ACF (Inria) Miss**	**ACF (Caltech) Hit**	**ACF (Inria) Hit**
DPM Miss	**HOG Miss**	33060		3509		10042		2393	
DPM Miss	**HOG Hit**	6383		1928		1577		2598	
DPM Hit	**HOG Miss**	1655		2523		1004		3756	
DPM Hit	**HOG Hit**	1826		2451		1240		5516	
κ = 0.312									
Number of total true positives = 97311									
Number of total false positives = 134258									

Table 5.27 The diversity between DPM, ACF trained with Caltech when “SelectStrongest” is true, ACF trained with INRIA when “SelectStrongest” is true and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 1

		ACF (Caltech) Miss	ACF (Inria) Miss	ACF (Caltech) Miss	ACF (Inria) Hit	ACF (Caltech) Hit	ACF (Inria) Miss	ACF (Caltech) Hit	ACF (Inria) Hit
DPM Miss	**HOG Miss**	29743		2167		9203		995	
DPM Miss	**HOG Hit**	9700		3290		2416		3996	
DPM Hit	**HOG Miss**	910		1044		565		924	
DPM Hit	**HOG Hit**	2571		3930		1679		8348	
κ = 0.351									
Number of total true positives = 329740									
Number of total false positives = 926090									

Table 5.28 The diversity between DPM, ACF trained with Caltech when “SelectStrongest” is false, ACF trained with INRIA when “SelectStrongest” is true and HOG “UprightPeople_96x48” when “MergeDetections” is true and “ClassificationThreshold” is 1

		ACF (Caltech) Miss	ACF (Inria) Miss	ACF (Caltech) Miss	ACF (Inria) Hit	ACF (Caltech) Hit	ACF (Inria) Miss	ACF (Caltech) Hit	ACF (Inria) Hit
DPM Miss	**HOG Miss**	31421		2891		11681		3011	
DPM Miss	**HOG Hit**	6022		1635		1938		2911	
DPM Hit	**HOG Miss**	1534		2055		1125		4224	
DPM Hit	**HOG Hit**	1745		2180		1321		5787	
κ = 0.307									
Number of total true positives = 283040									
Number of total false positives = 276704									

Table 5.29 The diversity between DPM, ACF trained with Caltech when “SelectStrongest” is true, ACF trained with INRIA when “SelectStrongest” is false and HOG “UprightPeople_96x48” when “MergeDetections” is true and “ClassificationThreshold” is 1

		ACF (Caltech) Miss	**ACF (Inria) Miss**	**ACF (Caltech) Miss**	**ACF (Inria) Hit**	**ACF (Caltech) Hit**	**ACF (Inria) Miss**	**ACF (Caltech) Hit**	**ACF (Inria) Hit**
DPM Miss	**HOG Miss**	31654		4915		9687		2748	
DPM Miss	**HOG Hit**	6025		2306		1377		2798	
DPM Hit	**HOG Miss**	1386		2792		832		3928	
DPM Hit	**HOG Hit**	1695		2582		1135		5621	
κ = 0.303									
Number of total true positives = 593160									
Number of total false positives = 394537									

Table 5.30 The diversity between DPM, ACF trained with Caltech when “SelectStrongest” is false, ACF trained with INRIA when “SelectStrongest” is false and HOG “UprightPeople_96x48” when “MergeDetections” is true and “ClassificationThreshold” is 1

		ACF (Caltech) Miss	**ACF (Inria) Miss**	**ACF (Caltech) Miss**	**ACF (Inria) Hit**	**ACF (Caltech) Hit**	**ACF (Inria) Miss**	**ACF (Caltech) Hit**	**ACF (Inria) Hit**
DPM Miss	**HOG Miss**	30294		4018		11047		3645	
DPM Miss	**HOG Hit**	5802		1855		1600		3249	
DPM Hit	**HOG Miss**	1324		2265		894		4455	
DPM Hit	**HOG Hit**	1653		2272		1177		5931	
κ = 0.302									
Number of total true positives = 778889									
Number of total false positives = 536983									

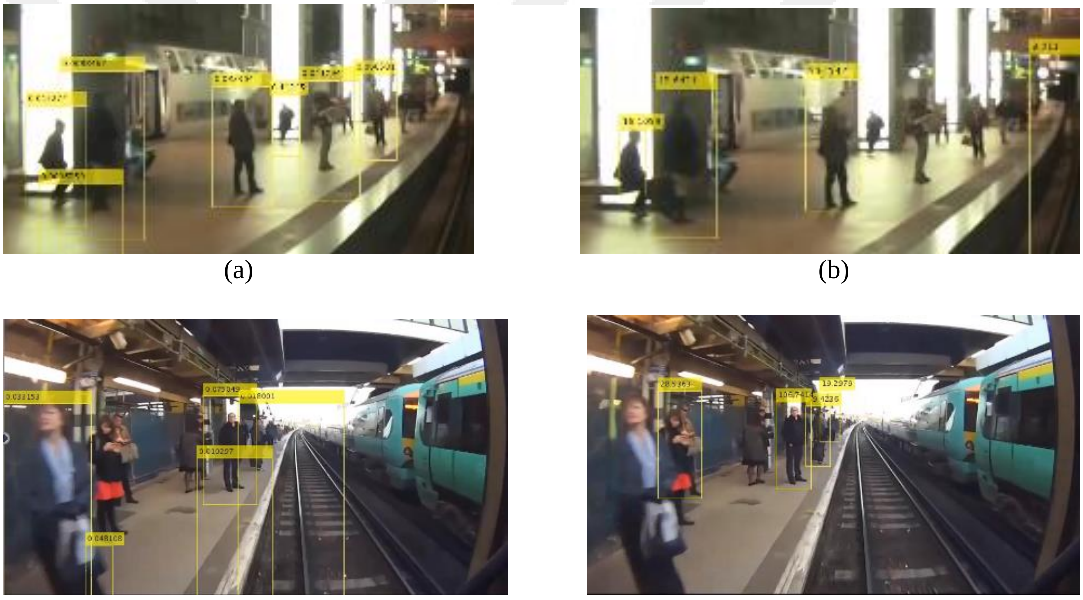
Table 5.31 The diversity between DPM, ACF trained with Caltech when “SelectStrongest” is false, ACF trained with INRIA when “SelectStrongest” is false and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 1

		ACF (Caltech) Miss	ACF (Inria) Miss	ACF (Caltech) Miss	ACF (Inria) Hit	ACF (Caltech) Hit	ACF (Inria) Miss	ACF (Caltech) Hit	ACF (Inria) Hit
DPM Miss	HOG Miss	27542		2642		10244		1680	
DPM Miss	HOG Hit	8554		3231		2403		5214	
DPM Hit	HOG Miss	714		956		534		1239	
DPM Hit	HOG Hit	2263		3581		1537		9147	
κ = 0.350									
Number of total true positives = 1011318									
Number of total false positives = 1328815									

Table 5.32 The diversity between DPM, ACF trained with Caltech when “SelectStrongest” is false, ACF trained with INRIA when “SelectStrongest” is false and HOG “UprightPeople_96x48” when “MergeDetections” is false and “ClassificationThreshold” is 0

		ACF (Caltech) Miss	ACF (Inria) Miss	ACF (Caltech) Miss	ACF (Inria) Hit	ACF (Caltech) Hit	ACF (Inria) Miss	ACF (Caltech) Hit	ACF (Inria) Hit
DPM Miss	HOG Miss	21389		1118		9415		620	
DPM Miss	HOG Hit	14707		4755		3232		6274	
DPM Hit	HOG Miss	307		298		440		379	
DPM Hit	HOG Hit	2670		4239		1631		10007	
κ = 0.302									
Number of total true positives = 1500120									
Number of total false positives = 8917708									

According to all these tables above, the ensembled method is proper for reducing missed ground truths. This inference can be corroborated with examples from RAWPED. Examples of the potential of the two models to complement each other are shown in Figure 5.5. In Figure 5.5.a, when the bounding boxes, detected by the HOG+SVM model, are considered in the image containing the station departure scene in the artificial lighting environment, it seems that HOG+SVM model is much more successful than the ACF + AdaBoost model given in Figure 5.5.b. Although the overall performance of HOG+SVM in railway applications is very low compared to the ACF+AdaBoost model, it can contribute to the true positive performance improvement in special situations such as these scenes. In Figure 5.5.c, HOG + SVM is much more successful than ACF + AdaBoost (Figure 5.5.d), in order to find the large-scale pedestrian closer to the camera, although it produces too many false positives throughout the scene. Although the combination of the results of the two models contain almost all the pedestrians in the scene, neither of the two models were able to detect the pedestrian whose back is turned.



(a) (b)

(c) (d)

Figure 5.5 The detected bounding boxes(yellow), in the station departure scene with artificial lighting environment, by (a) HOG+SVM based model, (b) ACF+AdaBoost model and in another station scene with different angle of camera, by (c) HOG+SVM based model, (d) ACF+AdaBoost model

As a result, the diversity measurements and examples from images support the idea of the complementarity of the models. Hence, the first stage of proposed model will be implemented in the light of this information in next chapter.

CHAPTER SIX
APPLICATIONS AND RESULTS

The results of first stage of the proposed system was already given in Chapter 5, when the complementarity of detectors was investigated. Figure 6.1 gives information about Stage 1 and its quantative results.

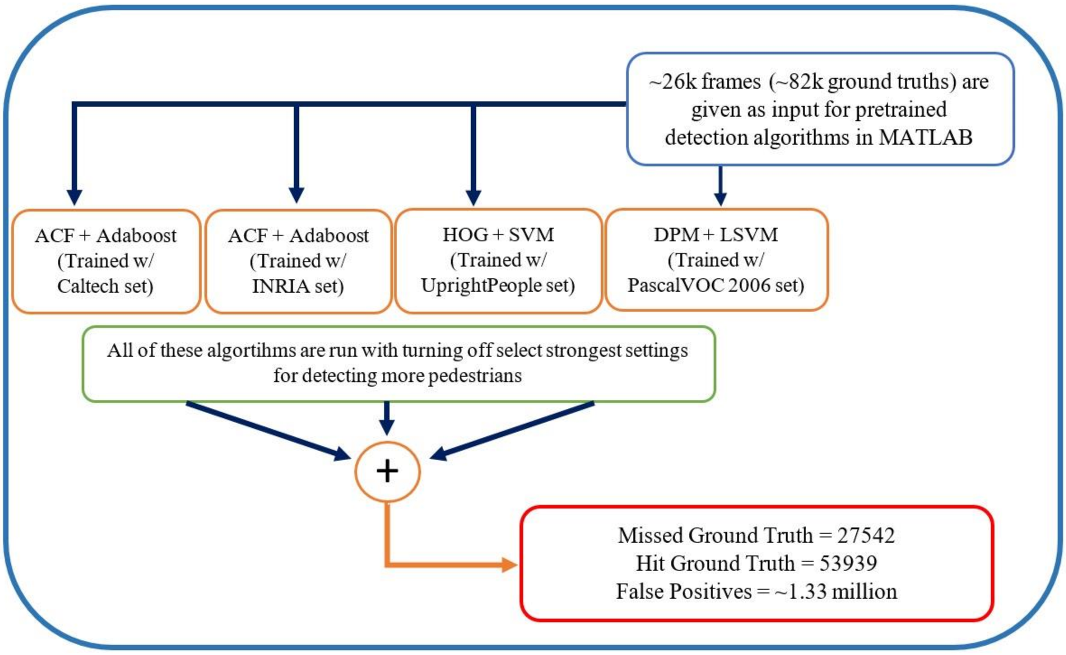

~26k frames (~82k ground truths) are given as input for pretrained detection algorithms in MATLAB
ACF + Adaboost (Trained w/ Caltech set)
ACF + Adaboost (Trained w/ INRIA set)
HOG + SVM (Trained w/ UprightPeople set)
DPM + LSVM (Trained w/ PascalVOC 2006 set)
All of these algortihms are run with turning off select strongest settings for detecting more pedestrians
+
Missed Ground Truth = 27542
Hit Ground Truth = 53939
False Positives = ~1.33 million

Figure 6.1 Scheme for first stage of the proposed system

The results of the used ensembled method for Stage 1 is also shown in Table 5.31. The ensembled version shown in Table 5.32 is not prefered because of its false positives are so much more but the false negatives are not less with compared to redundancy of false positives.

Afterwards the ensembled detection part is applied, two main problems come to exist. The first one is the number of missed ground truths and the second one is the redundancy of false positives. RAWPED is seperated four groups according to height of the pedestrians to overcome this problems.

For better understanding of implementation, pedestrians in near scale are chosen to apply the ensembled classification method in Stage 2. Figure 6.2 shows the work-flow for second stage.

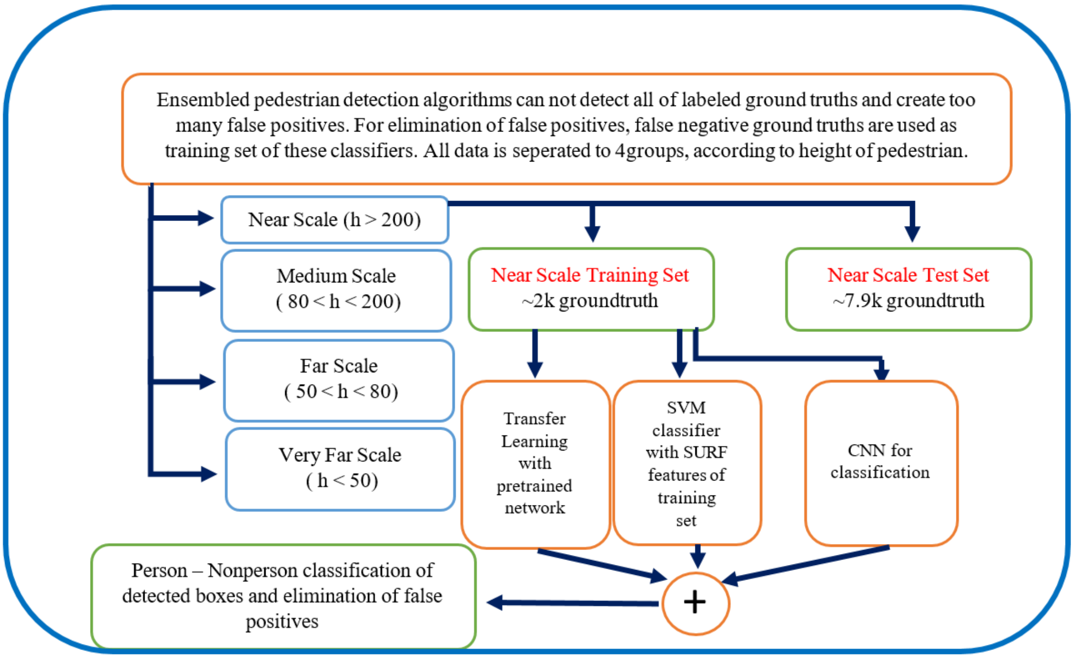

Ensembled pedestrian detection algorithms can not detect all of labeled ground truths and create too many false positives. For elimination of false positives, false negative ground truths are used as training set of these classifiers. All data is seperated to 4groups, according to height of pedestrian.
Near Scale (h > 200)
Medium Scale (80 < h < 200)
Far Scale (50 < h < 80)
Very Far Scale (h < 50)
Near Scale Training Set ~2k groundtruth
Near Scale Test Set ~7.9k groundtruth
Transfer Learning with pretrained network
SVM classifier with SURF features of training set
CNN for classification
+
Person – Nonperson classification of detected boxes and elimination of false positives

Figure 6.2 Work-flow of the ensembled classification in Stage 2

The detailed results for near scale from Stage 1 are given in Table 6.1. The missed near scale ground truths are used for training and fine tuning parts of the second stage. Therefore, Table 6.1 shows only test part (hit ground truths and total detected bounding boxes) of the pedestrians in near scale.

Table 6.1 Test part of first stage results for pedestrians in near scale

Ground Truths	Total Detected Bounding Boxes	LAMR
7844	420375	0.58

The main aim is reducing the number of total detected bounding boxes without changing the number of ground truths. However, the classification models might classify the boxes wrong and this situation causes decrease on number of ground truths. Hence, all methods given in Section 4.2 are applied to near scale to get better results.

First of all, the transfer learning method is applied to pre-trained models. All models are available in Matlab. Afterwards changing the fully connected layer which controls the number of types for classification, all models are fine-tuned with only ~2k training set. Table 6.2 gives results for fine-tuned pre-trained networks.

Table 6.2 Results for individual fine-tuned pre-trained networks

	Hit	**Miss**	**Total Box**	**LAMR**
AlexNet	5121	2723	9971	0.53
GoogleNet	3838	4006	5972	0.72
VGG-16	3920	3924	20069	0.68
ResNet	2094	5750	2520	0.78

The results for CNN which is trained with the near scale training set from scratch are given in Table 6.3.

Table 6.3 Results for CNN which is trained from scratch

Hit	**Miss**	**Total Box**	**LAMR**
4270	3574	13295	0.67

The final part of the proposed ensembled classifier is non-deep SVM classifier and it is trained with bag of visual features from near scale training set. The results are given in Table 6.4.

Table 6.4 Results from non-deep SVM classifier

Hit	**Miss**	**Total Box**	**LAMR**
6243	1601	43781	0.57

According to these results SVM classifier gives the best results alone. However, there is a setting as initial learning rate for training process and the default value of the inital learning rate in Matlab is 0.001. If this value is changed to 0.0005 for VGG-16 and retrained VGG-16 as version two. VGG-16 is chosen for this procedure because of its total box value can be reduced more than others. The results for VGG-16v2 are shown in Table 6.5.

Table 6.5 Results for VGG-16v2

Hit	Miss	Total Box	LAMR
6488	1356	9853	0.48

For ensembled method, VGG-16v2 + SVM + CNN are applied to test set and the results are given in Table 6.6. According to these results, hit ground truths are more than all of individual results of classifiers which is aim to be. Although the total box can not be reduced enough, the LAMR criteria for better detection can be reduced. Hence, both LAMR and number of hit ground truth results are acceptable.

Table 6.6 Results for VGG-16v2 + SVM + CNN ensembled classifier model

Hit	Miss	Total Box	LAMR
7696	148	43811	0.46

CHAPTER SEVEN
CONCLUSIONS

In this thesis, pedestrian detection for railway driver support systems is implemented in two stage and the dataset is prepared for applications. Due to the inadequacy in dataset, the ensemble of detectors is preferred instead of training a network from scratch.

At the first stage of the purposed method, main aim is to detect as many as possible true positives. When all models are examined individually, the necessity of ensemble system is observed. Afterwards detailed analyzes about complementarity of all chosen non-deep detectors, the most convenient version is applied to RAWPED. The results of ensemble version show that LAMR value of these results is lower than all individual models. That means non-deep detectors are complement each others and detect more pedestrians.

However, the downside of the first stage is the total number of false positives. For reducing that number, the ensemble of fine-tuned CNNs, CNN trained from scratch and SVM classifier is proposed. The second stage is analyzed only for near scale pedestrians. After fine-tuning of CNNs with false negative near scale pedestrians, the best result is observed from VGG-16v2. Actually, that can be predictable, because of VGG-16 the deepest and homogeneous network from between pre-trained CNNs.

The SVM classifier and CNN are trained from scratch for that problem. The ensemble version of these VGG-16v2 + SVM + CNN gives the best result for detection of near scale pedestrian, its LAMR value is the lowest value determined in thesis.

As a conclusion of these results, non-deep detectors are dependent on their training dataset, but they can complement each other for same problem with different dataset. This complementarity is an advantage for inadequate datasets. Although ensemble of detectors produces many false positives, another ensemble method with classifiers can be feasible for reducing the number of false positives.

REFERENCES

All, K., Hasler, D., & Fleuret, F. (2011). FlowBoost—appearance learning from sparsely annotated video. In *2011 IEEE Conference on Computer Vision and Pattern Recognition*, 1433-1440.

Angelova, A., Krizhevsky, A., Vanhoucke, V., Ogale, A. S., & Ferguson, D. (2015). Real-time pedestrian detection with deep network cascades. In *British Machine Vision Conference, 2*, 4.

Aminmansour, S., Maire, F., Larue, G. S., & Wullems, C. (2015). Improving near-miss event detection rate at railway level crossings. In *2015 IEEE International Conference on Digital Image Computing: Techniques and Applications*, 1-8.

Benenson, R., Mathias, M., Timofte, R., & Van Gool, L. (2012). Pedestrian detection at 100 frames per second. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2903-2910.

Benenson, R., Omran, M., Hosang, J., & Schiele, B. (2014). Ten years of pedestrian detection, what have we learned?. In *European Conference on Computer Vision*, 613-627.

Bila, C., Sivrikaya, F., Khan, M. A., & Albayrak, S. (2017). Vehicles of the future: A survey of research on safety issues. In *IEEE Transactions on Intelligent Transportation Systems*, *18*(5), 1046-1065.

Cai, Z., Saberian, M., & Vasconcelos, N. (2015). Learning complexity-aware cascades for deep pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 3361-3369.

Cao, J., Pang, Y., & Li, X. (2017). Learning multilayer channel features for pedestrian detection. *IEEE Transactions on Image Processing*, *26*(7), 3210-3220.

Crete, F., Dolmiere, T., Ladret, P., & Nicolas, M. (2007). The blur effect: perception and estimation with a new no-reference perceptual blur metric. In *Human Vision and Electronic Imaging XII*, *6492*, 64920I.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *1*, 886-893.

Dollár, P., Appel, R., Belongie, S., & Perona, P. (2014). Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(8), 1532-1545.

Dollar, P., Wojek, C., Schiele, B., & Perona, P. (2012). Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(4), 743-761.

Duan, L., Tsang, I. W., Xu, D., & Maybank, S. J. (2009). Domain transfer svm for video concept detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 1375-1381.

Enzweiler, M., & Gavrila, D. M. (2009). Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(12), 2179-2195.

Ess, A., Leibe, B., Schindler, K., & Van Gool, L. (2008). A mobile vision system for robust multi-person tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 1-8.

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

Fleiss, J. L., Levin, B., & Paik, M. C. (2013). *Statistical methods for rates and proportions* (3rd ed.). New Jersey: John Wiley & Sons.

Freeman, J., & Rakotonirainy, A. (2015). Mistakes or deliberate violations? A study into the origins of rule breaking at pedestrian train crossings. *Accident Analysis & Prevention*, *77*, 45-50.

Gandhi, T., & Trivedi, M. M. (2007). Pedestrian protection systems: Issues, survey, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, *8*(3), 413-430.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition,* 3354-3361.

Geronimo, D., Lopez, A. M., Sappa, A. D., & Graf, T. (2010). Survey of pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(7), 1239-1258.

Hosang, J., Omran, M., Benenson, R., & Schiele, B. (2015). Taking a deeper look at pedestrians. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 4073-4082.

Hu, Q., Wang, P., Shen, C., van den Hengel, A., & Porikli, F. (2017). Pushing the limits of deep cnns for pedestrian detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(6), 1358-1368.

Kirbas, C., & Quek, F. (2004). A review of vessel extraction techniques and algorithms. *ACM Computing Surveys*, *36*(2), 81-121.

Kulis, B., Saenko, K., & Darrell, T. (2011). What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *2011 IEEE Conference on Computer Vision and Pattern Recognition*, 1785-1792.

Lavelle M., (2015). Train deaths rise amid energy-driven rail transformation. *Special Report on Train Tragedies and Transformations, Scientific American.*

Li, B., Chen, Y., & Wang, F. Y. (2015). Pedestrian detection based on clustered poselet models and hierarchical and–or grammar. *IEEE Transactions on Vehicular Technology*, *64*(4), 1435-1444.

Li, J., Liang, X., Li, J., Wei, Y., Xu, T., Feng, J., & Yan, S. (2017). Multi-stage object detection with group recursive learning. *IEEE Transactions on Multimedia*, *20*(7), 1645-1655.

Li, J., Liang, X., Shen, S., Xu, T., Feng, J., & Yan, S. (2018). Scale-aware fast R-CNN for pedestrian detection. *IEEE Transactions on Multimedia*, *20*(4), 985-996.

Li, K., Wang, X., Xu, Y., & Wang, J. (2016). Density enhancement-based long-range pedestrian detection using 3D range data. *IEEE Transactions on Intelligent Transportation Systems*, *17*(5), 1368-1380.

Li, Q., Wang, H., Yan, Y., Li, B., & Chen, C. W. (2017). Local co-occurrence selection via partial least squares for pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, *18*(6), 1549-1558.

Li, W., Ni, H., Wang, Y., Fu, B., Liu, P., & Wang, S. (2014). Detection of partially occluded pedestrians by an enhanced cascade detector. *IET Intelligent Transport Systems*, *8*(7), 621-630.

Liu, J., Shah, M., Kuipers, B., & Savarese, S. (2011). Cross-view action recognition via view knowledge transfer. In *2011 IEEE Conference on Computer Vision and Pattern Recognition*, 3209-3216.

Liu, W., Yu, B., Duan, C., Chai, L., Yuan, H., & Zhao, H. (2015). A pedestrian-detection method based on heterogeneous features and ensemble of multi-view–pose parts. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 813-824.

Maire, F. (2007). Vision based anti-collision system for rail track maintenance vehicles. In *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, 170-175.

Nam, W., Dollár, P., & Han, J. H. (2014). Local decorrelation for improved pedestrian detection. In *Advances in Neural Information Processing Systems*, 424-432.

Nassu, B. T., & Ukai, M. (2012). A vision-based approach for rail extraction and its application in a camera pan–tilt control system. *IEEE Transactions on Intelligent Transportation Systems*, *13*(4), 1763-1771.

Ouyang, W., Zeng, X., & Wang, X. (2013). Modeling mutual visibility relationship in pedestrian detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 3222-3229.

Ouyang, W., Zhou, H., Li, H., Li, Q., Yan, J., & Wang, X. (2017). Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (1), 1.

Paisitkriangkrai, S., Shen, C., & van den Hengel, A. (2016). Pedestrian detection with spatially pooled features and structured ensemble learning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *38*(6), 1243-1257.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345-1359.

Park, D., Ramanan, D., & Fowlkes, C. (2010). Multiresolution models for object detection. In Daniilidis K., Maragos P., Paragios N. (Eds.). *Computer Vision – ECCV 2010, Lecture Notes in Computer Science, 6314,* (241-254). Berlin: Springer.

Qi, G. J., Aggarwal, C., Rui, Y., Tian, Q., Chang, S., & Huang, T. (2011). Towards cross-category knowledge propagation for learning visual concepts. In *2011 IEEE Conference on Computer Vision and Pattern Recognition*, 897-904.

Ribeiro, D., Carneiro, G., Nascimento, J. C., & Bernardino, A. (2017). Multi-channel convolutional neural network ensemble for pedestrian detection. In Alexandre L., Salvador Sanchez J., Rodrigues J. (Eds.). *Pattern Recognition and Image Analysis, Lecture Notes in Computer Sciences, 10255,* (122-130). Cham: Springer.

Ribeiro, D., Nascimento, J. C., Bernardino, A., & Carneiro, G. (2017). Improving the performance of pedestrian detectors using convolutional learning. *Pattern Recognition*, *61*, 641-649.

Roth, P. M., Sternig, S., Grabner, H., & Bischof, H. (2009). Classifier grids for robust adaptive object detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition,* 2727-2734.

Selver, A. M., Ataç, E., Belenlioglu, B., Dogan, S., & Zoral, Y. E. (2018). Visual and LIDAR data processing and fusion as an element of real time big data analysis for rail vehicle driver support systems. In Kohli S., Kumar A., Easton J., & Roberts. (Eds), *Innovative Applications of Big Data in the Railway Industry* (40-66). Hershey, PA: IGI Global.

Selver, M. A., Er, E., Belenlioglu, B., & Soyaslan, Y. (2016). Camera based driver support system for rail extraction using 2D Gabor wavelet decompositions and morphological analysis. In *2016 IEEE International Conference on Intelligent Rail Transportation*, 270-275.

Simonnet, D., Velastin, S. A., Turkbeyler, E., & Orwell, J. (2012). Backgroundless detection of pedestrians in cluttered conditions based on monocular images: a review. *IET Computer Vision*, *6*(6), 540-550.

Stalder, S., Grabner, H., & Van Gool, L. (2010). Cascaded confidence filtering for improved tracking-by-detection. In Daniilidis K., Maragos P., Paragios N. (Eds.), *Computer Vision – ECCV 2010, Lecture Notes in Computer Sciences, 6311*, (369-382).Berlin: Springer.

Wang, X., Wang, M., & Li, W. (2014). Scene-specific pedestrian detection for static video surveillance. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *36*(2), 361-374.

Wojek, C., Walk, S., & Schiele, B. (2009). Multi-cue onboard pedestrian detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 794-801.

Yan, J., Zhang, X., Lei, Z., Liao, S., & Li, S. Z. (2013). Robust multi-resolution pedestrian detection in traffic scenes. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 3033-3040.

Yang, F., Choi, W., & Lin, Y. (2016). Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2129-2137.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. In *Advances in Neural Information Processing Systems,* 3320-3328.

Yule, G. U. (1900). On the association of attributes in statistics. *Phil. Trans., A*, *194*, 257-319.

Zhang, S., Benenson, R., & Schiele, B. (2017). Citypersons: A diverse dataset for pedestrian detection. In *The IEEE Conference on Computer Vision and Pattern Recognition, 1*(2), 3.

Zhang, S., Benenson, R., Omran, M., Hosang, J., & Schiele, B. (2018). Towards reaching human performance in pedestrian detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *40*(4), 973-986.

Zhang, S., Bauckhage, C., & Cremers, A. B. (2015). Efficient pedestrian detection via rectangular features based on a statistical shape model. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 763-775.

Zhu, C., & Peng, Y. (2015). A boosted multi-task model for pedestrian detection with occlusion handling. *IEEE Transactions on Image Processing*, *24*(12), 5619-5629.