A Novel Semantic Feature Fusion-based Pedestrian Detection System to Support Autonomous Vehicles

by

Mingzhi Sha

Thesis submitted to the University of Ottawa
in partial Fulfillment of the requirements for the
Master of Computer Science degree

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Intelligent transportation systems (ITS) have become a popular method to enhance the safety and efficiency of transportation. Pedestrians, as an essential participant of ITS, are very vulnerable in a traffic collision, compared with the passengers inside the vehicle. In order to protect the safety of all traffic participants and enhance transportation efficiency, the novel autonomous vehicles are required to detect pedestrians accurately and timely.

In the area of pedestrian detection, deep learning-based pedestrian detection methods have gained significant development since the appearance of powerful GPUs. A large number of researchers are paying efforts to improve the accuracy of pedestrian detection by utilizing the Convolutional Neural Network (CNN)-based detectors.

In this thesis, we propose a one-stage anchor-free pedestrian detector named Bi-Center Network (BCNet), which is aided by the semantic features of pedestrians' visible parts. The framework of our BCNet has two main modules: the feature extraction module produces the concatenated feature maps that extracted from different layers of ResNet, and the four parallel branches in the detection module produce the full body center keypoint heatmap, visible part center keypoint heatmap, heights, and offsets, respectively. The final bounding boxes are converted from the high response points on the fused center keypoint heatmap and corresponding predicted heights and offsets.

The fused center keypoint heatmap contains the semantic feature fusion of the full body and the visible part of each pedestrian. Thus, we conduct ablation studies and discover the efficiency of feature fusion and how visibility features benefit the detector's performance by proposing two types of approaches: introducing two weighting hyper-parameters and applying three different attention mechanisms.

Our BCNet gains 9.82% MR^{-2} (the lower the better) on the Reasonable setup of the CityPersons dataset, compared to baseline model which gains 12.14% MR^{-2}. The experimental results indicate that the performance of pedestrian detection could be significantly improved because the visibility semantic could prompt stronger responses on the heatmap. We compare our BCNet with state-of-the-art models on the CityPersons dataset and ETH dataset, which shows that our detector is effective and achieves a promising performance.

List of Publications

The following publications are part of this thesis, and my supervisor is Azzedine boukerche.

- **Journals:**
  - Azzedine Boukerche and Mingzhi Sha. *Design Guidelines on Deep Learning-based Pedestrian Detection Methods for Supporting Autonomous Vehicles.* Accepted by the ACM Computing Surveys (CSUR) in April 2021.
  - Mingzhi Sha and Azzedine Boukerche. *Visibility Semantic Feature-Aided Pedestrian Detection for Supporting Autonomous Vehicles.* Submitted to Elsevier, Computer Communications in July 2020.
  - Mingzhi Sha and Azzedine Boukerche. *Performance Evaluation of CNN-based Pedestrian Detectors for Autonomous Vehicles.* Submitted to Elsevier, Ad Hoc Networks in April 2021.
- **Conference:**
  - Mingzhi Sha and Azzedine Boukerche. *Semantic fusion-based pedestrian detection for supporting autonomous vehicles.* In Proceedings of the IEEE Symposium on Computers and Communications, pages 618–623, 2020.
  - Mingzhi Sha and Azzedine Boukerche. *Performance Evaluation of Pedestrian Detectors for Autonomous Vehicles.* Accepted by the 4th International Workshop on Intelligent Transportation and Autonomous Vehicles Technologies (ITAVT 2021).

Acknowledgements

I would like to thank everyone who helped me. Without them, I will not be able to successfully complete my master's degree.

Firstly, I would like to express my sincere gratitude to my parents for their love. They support me both mentally and financially. Thanks for the education and upbringing of my family members, which makes me the person I am today.

Secondly, I would like to thank my supervisor Professor Azzedine Boukerche for his continuous encouragement, guidance, and financial support. He regularly holds laboratory meetings and provides valuable suggestions.

Next, I would like to thank all PARADISE Laboratory members, who have provided me with generous help during my study. I want to thank Dr. Peng Sun for his patience in revising my paper and helpful suggestions for improving my writing skills. I want to thank Mr. Xiren Ma for helping me to configure the computer system. Thanks to Jiahao Wang, Yue Chen, Dunhao Zhong, Heqi Cui, Qiyue Wu, Zhijun Hou, Shichao Guan, Yiheng Zhao, Lining Zheng, Weihong Zhao, Chong Luo, and all PARADISE Laboratory members who made my master thesis possible.

Table of Contents

List of Tables viii

List of Figures ix

Nomenclature xi

1 Introduction 1

1.1 Background and Problem Statement 1

1.2 Motivation and Contribution 2

1.3 Thesis Outline 5

2 Preliminaries 6

2.1 Object Detector Frameworks 6

2.1.1 Basic Concepts of Deep Learning-based Methods 6

2.1.2 Backbone Models 7

2.1.3 Benchmark Detection Frameworks 10

2.2 Pedestrian Detection Datasets, Evaluation Metrics, and Inherent Attributes 14

2.2.1 Pedestrian Detection Datasets 14

2.2.2 Evaluation Metrics 17

2.2.3 Inherent Attributes of Pedestrian Detection 18

3 Related Work 22

3.1 Occlusion Handling 22

3.1.1 Part Information Benefits the Occlusion Handling 22

3.1.2 Addressing the Occlusion Caused by the Crowd . . . 24
3.1.3 Comparisons and Conclusions . . . 26
3.2 Multi-scale Feature Extraction . . . 27
3.2.1 Comparisons and Conclusions . . . 29
3.3 Multi-scope Data Utilization . . . 30
3.3.1 Exploiting Semantic Features . . . 30
3.3.2 Using Keypoint Sets to Replace Bounding Boxes . . . 32
3.3.3 Expanding the Data . . . 34
3.3.4 Comparisons and Conclusions . . . 34
3.4 Hard Negatives Processing . . . 34
3.4.1 Dealing with Hard Negatives by Sampling . . . 36
3.4.2 Adopting Attention Mechanisms . . . 36
3.4.3 Comparisons and Conclusions . . . 39

4 Proposed Methods **41**
4.1 Pre-processing . . . 41
4.1.1 Annotations . . . 41
4.1.2 Images . . . 42
4.2 Proposed Networks . . . 44
4.2.1 Model Overview . . . 44
4.2.2 Pedestrian Detection with the Semantic of the Visible Part . . . 48
4.2.3 Fusing the Full Body Semantic and Visible Part Semantic . . . 49
4.2.4 Loss Function . . . 52
4.3 Post-processing . . . 53
4.3.1 NMS . . . 54

5 Experiments **55**
5.1 Datasets and Evaluation Metrics . . . 55
5.2 Experimental Settings . . . 56

5.2.1 Baseline Model with Attention Mechanisms 56
5.2.2 Existing Methods 57
5.3 Experimental Results 58
5.3.1 The Ablation Study on Two Hyper-parameters α and β 59
5.3.2 The Ablation Study on Attention Modules 60
5.3.3 Evaluation and Analysis 62
5.3.4 Test on the Additional Dataset 63
5.4 Evaluation Demo 64

6 Conclusion and Future Work **73**
6.1 Conclusion 73
6.2 Future Work 74
6.2.1 Accuracy-related open challenges 74
6.2.2 Efficiency-related open challenges 76
6.2.3 Security 78

References **80**

List of Tables

2.1 Overview of the existing popular backbones. 8

2.2 Some comparions of most commonly used pedestrian datasets. 16

2.3 Optimization methods overview. 21

3.1 Representative pedestrian detectors for occlusion handling. 23

3.2 Representative pedestrian detectors for addressing multi-scale feature extraction. 27

3.3 Overview of the representative pedestrian detectors in Section 3.3. 31

3.4 The overview of the representative detectors in Section 3.4. 35

4.1 The structure of ResNet-50 [124]. 45

4.2 Variants and aliases of BCNet that adopt attention methods in different locations. 52

5.1 Evaluation Setups of the CityPersons dataset [249] 56

5.2 Variants and aliases of baseline model CSP [162] that adopt attention methods in different locations. 57

5.3 Experimental settings of detectors' training. 70

5.4 Experimental results on the CityPersons validation set. 71

5.5 Experimental results on the CityPersons validation set. The results in boldface indicate the best performance. 72

List of Figures

1.1 An example of detecting pedestrians on an image. 2

1.2 Box 1 denotes a pedestrian that is occluded by the bush from the perspective of the current vehicular camera. However, this pedestrian can be clearly seen from the perspective of other vehicles (in Box 3) on the other side of the road. Box 2 denotes a group of pedestrians. 3

2.1 Examples for each dataset listed in Table 2.2. 15

2.2 The inherent attributes and major challenges of pedestrian detection. . . . 19

3.1 An illustration of the working pipeline of MGA module in [186]. 39

4.1 Images visualization. 43

4.2 An illustration of CKVP and CKFB. The cropped image is from the CityPersons dataset [249]. 45

4.3 The architecture of BCNet. We take ResNet-50 [124] as the backbone in ConvNet. A1-A4 and B1-B4 are used to denote the locations. 46

4.4 The distribution of distances between the CKVP and CKFB for each pedestrian in the CityPersons training set in the resolution of 256×512. Δx and Δy represent the horizontal and vertical distance, respectively. 48

4.5 The naive attention module. The green CKFB and CKVP are draft heatmaps, the red CKFB and CKVP are the re-weighted heatmaps that will be added up to produce the final heatmap. 50

4.6 The dimensions of corresponding feature maps are denoted in height × width × channel format. The input feature maps' channel is 256 in the SE layer of our network. The output feature map is re-weighted by applying the attention scalar to the input feature map. 51

4.7 The comparison of before applying NMS and after applying NMS. 54

5.1 Experiments of varying α and β on Reasonable setup of CityPersons dataset, evaluated by MR^{-2}. 59

5.2 The heatmaps are cropped into size 40×60 for visualization. Width and height are in pixels. 61

5.3 ETH dataset results 65

5.4 ETH dataset results 66

5.5 Plotting prediction results and comparison with the baseline model. . . . 67

5.6 Plotting prediction results and comparison with the baseline model. 68

5.7 Plotting prediction results and comparison with the other models. 69

Nomenclature

ADM	Active Detection Module
AggLoss	Aggregation Loss
AP	Average Precision
ASDN	Adversarial Spatial Dropout Network
bbox	bounding box
BCN	Binary Classification Network
BCNet	Bi-Center Network
Bi-box	Bi-box Regression
CKFB	Center Keypoint of the Full Body
CKVP	Center Keypoint of the Visible Part
CNN	Convolutional Neural Network
Conv	convolutional
CPB	Convolutional Predictor Block
CSP	Center and Scale Prediction-based Detector
ECP	EuroCity Persons dataset
FC	fully connected
FCOS	Fully Convolutional One-stage
FN	False Negative

FP	False Positive
FPN	Feature Pyramid Network
FPPI	False Positive per Image
FPS	Frames per Second
GA-RPN	Guided Anchoring Region Proposal Network
GAN	Generative Adversarial Network
GDFL	Graininess-aware Deep Feature Learning
IoT	Internet of Things
IoU	Intersection-over-Union
ITS	Intelligent Transportation Systems
Lidar	Light Detection and Ranging
mAP	mean Average Precision
MGA	Mask-guided Attention
MR	Miss Rate
MRF	Markov Random Field
NLP	Natural Language Processing
NMS	Non-Maximum Suppression
NMS	Non-maximum Suppression
OSD	One-stage Detector
PR	Precision
RC	Recall
RepLoss	Repulsion Loss
RPN	Region Proposal Network
SADR	Scale Adaptive Deconvolutional Regression

SAWL	Scale-aware Weighting Layer
SDS	Simultaneous Detection and Segmentation
SIL	Segmentation Infusion Layer
SPP	Spatial Pyramid Pooling
STN	Spatial Transformer Network
TN	True Negative
TP	True Positive
TSD	Two-stage Detector

Chapter 1

Introduction

Transportation systems function as the infrastructure system of our society. To improve service quality and safety of transportation systems, the idea of intelligent transportation systems (ITS) has been proposed. ITS enables all of the transportation system participants to communicate with each other by sending and receiving messages [40, 49, 177, 63, 13], which includes pedestrians, all types of vehicles, control centers, traffic signals, etc [168, 111, 38, 99, 196]. The communication could provide all the participants with the states, changes, and conditions of the surroundings [247, 31, 33], which can be very beneficial in improving transportation efficiency and safety [64, 65, 48, 54, 56, 19]. More researchers and industries are working on improving the transportation system by exploiting the properties of ITS, such as [61, 35, 30, 91, 32, 60, 2, 41, 75, 240, 198, 93].

1.1 Background and Problem Statement

The concept of autonomous vehicles was proposed decades ago. With the development of hardware and algorithms, the autonomous vehicles have become a reality [71, 62, 50, 76, 109, 1, 51, 22, 79, 3]. The autonomous vehicle is an important participant of the ITS, and it is capable of self-driving without human drivers' navigation [215, 16, 20]. Many traffic accidents are caused by human drivers' misoperation due to both physical and mental conditions. Besides, some complex traffic scenarios and bad weather conditions can also lead to traffic tragedies.

The application of autonomous vehicles could decrease the number of traffic accidents by making judgements more reasonably and driving more reliably [90, 34, 166, 70, 94]. Before putting the autonomous vehicles into use, the fundamental objective of the current stage is to ensure the safety of all traffic participants [165, 229, 45, 170]. Pedestrians are

very vulnerable in a traffic collision, compared to the passengers inside the vehicle [17]. Therefore, ensuring the safety of pedestrians is a key step in advancing the application of autonomous vehicles [7, 12, 214, 216].

Figure 1.1: An example of detecting pedestrians on an image.

To better protect pedestrians, the first thing we need to let the vehicle know is the pedestrians' location, so that the vehicle can decide whether or not to brake [52, 59, 212, 46, 6], as shown in Fig 1.1. Consequently, researchers have been working on the area of pedestrian detection, which is to find the locations of each pedestrian from the image and denote the locations by using bounding boxes (bbox) that are close to pedestrians.

The objective of object detection problem is to classify and spatially locate multiple objects that belong to different classes in the image at the same time. The outputs of the detection are the class categories and the set of bboxes of the objects. The bbox works to depict the location of object [187], which is usually given in the form of the left-top point's coordination and the height and width of the object.

1.2 Motivation and Contribution

There are two main ways to solve the detection task: one is the traditional machine learning-based methods, which exploit the pre-defined handmade feature descriptors (e.g., SIFT [163], HOG [95], and Haar [188]) to extract local features from the images; the other is the deep learning-based methods, which are capable of learning the features that extracted from input datasets during the training phases. More than a decade ago, pedestrian detection mainly relied on traditional machine learning-based methods; nowadays, the

Convolutional Neural Network (CNN), as a representative network in deep learning-based methods, is attracting researchers' interests and become a popular choice in the detection area.

LeNet-5 [151] is one of the early-stage CNN models, which was proposed by LeCun et al. in 1998. Because of the limited computational power at that time, CNN models were very small-scale and unable to handle tasks as complex as object detection. With the advance in computing power, researchers started to train CNN models on GPUs. AlexNet [147] was proposed by Krizhevsky et al. in 2012, which is the first CNN model that was trained on two parallel GPUs, making AlexNet much more large-scale and powerful than previous CNN models.

Since the effectiveness of AlexNet [147], many more researchers have been working on designing CNN-based detectors to improve the performance of pedestrian detection, such as [142], [248], and [249]. Many researchers have worked on occlusion handling because the occlusion is very common in real-world scenarios, as mentioned in [103] and [249]. However, we have some reasons to believe that detecting heavily occluded pedestrians (i.e., over 35% of the pedestrian's full body is occluded) is not the most urgent task in the area of pedestrian detection for supporting autonomous vehicles. Even though numerous studies

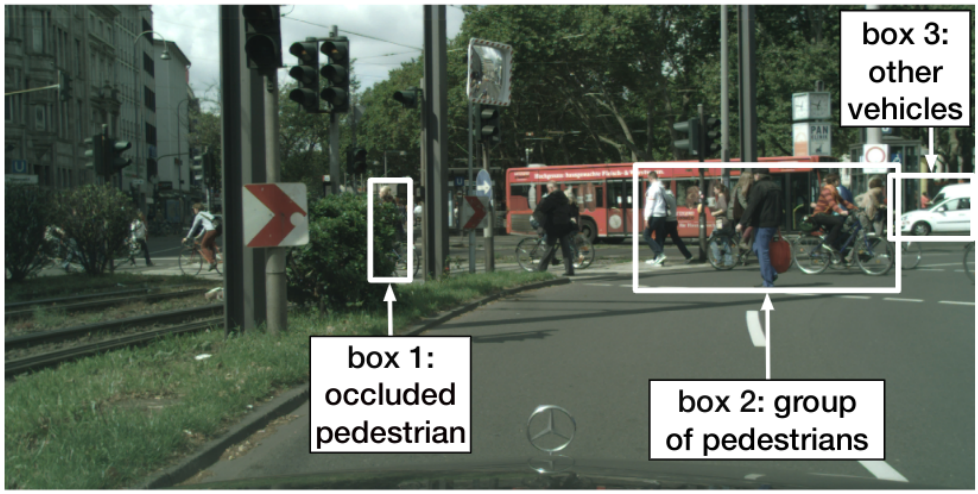


Figure 1.2: Box 1 denotes a pedestrian that is occluded by the bush from the perspective of the current vehicular camera. However, this pedestrian can be clearly seen from the perspective of other vehicles (in Box 3) on the other side of the road. Box 2 denotes a group of pedestrians.

have been conducted on occlusion handling (e.g., [233], [257], and [253]), the detection accuracy of heavily occluded pedestrians is at the bottleneck, far lower than the detection accuracy of other types of pedestrian samples. Furthermore, 48.8% of pedestrians are

occluded by other pedestrians in the CityPersons dataset [249] as mentioned in [233], but the autonomous vehicle is not required to predict how many pedestrians are in the crowd. In Fig 1.2, no matter how many pedestrians are in Box 2 predicted by an autonomous vehicle, the autonomous vehicle should brake safely to avoid harming that group of pedestrians. Lastly, ITS enables autonomous vehicles to send and receive information [179, 255, 98, 27, 10]. Thus, the pedestrian occluded by the bush can be detected by the vehicle in Box 3 by exploiting ITS's ability of sending and receiving information, so that the vehicles in this area will be informed of this pedestrian [57, 73, 77]. Consequently, we consider the importance and urgency of detecting the pedestrians under reasonable occlusion based on the above analysis, and we focus on improving detection accuracy on this type of pedestrians in this paper.

With the advent of GPUs, many CNN models have become more powerful. However, we must acknowledge that one challenge of deep learning-based pedestrian detection that applies to autonomous vehicles is reducing the dependency on computational ability, as vehicular GPUs are not as powerful as the GPUs researchers use in labs [15]. In addition, autonomous vehicles need to perform many other functions simultaneously, and GPUs can consume massive energy, which could prevent other functions from running properly. Consequently, reducing the dependency on computing power (GPUs) is an effective method for improving the practicality of pedestrian detectors in real-world scenarios.

In this paper, we focus on using limited computational ability to improve the performance of pedestrian detection under reasonable occlusion. Our work has the following contributions:

1. We utilize the visible semantic feature of each pedestrian and fuse with full body semantic feature to obtain the enriched feature of each pedestrian. To our best knowledge, we are the first to fuse the center keypoints of full body and the visible part to enhance the pedestrian detection accuracy.

2. We design and adopt different methods, including weighting hyper-parameters and three types of attention mechanisms, to study how the visible semantic feature can improve pedestrian detection accuracy. We visualize the fused heatmaps to study the effect of semantic fusion, and we conduct an ablation study to observe the performance of different attention modules that applied to different layers.

3. We reduce the detector's dependency on GPUs by training our Bi-Center Network (BCNet) on a single GPU with limited computing power, and our BCNet reaches promising results compared to other models trained on multiple powerful GPUs.

1.3 Thesis Outline

This thesis is organized as follows.

Chapter 2 starts with basic concepts that we will use in the thesis. We then overview the development of deep learning-based backbones. We classify classic and state-of-the-art detection frameworks from two different aspects. Next, we list some popular pedestrian detection dataset and introduce the unified evaluation metric. We compare the differences between pedestrian detection and object detection by analyzing the inherent attributes of pedestrian detection application scenarios.

Chapter 3 studies the pedestrian detectors from four main aspects: occlusion handling, multi-scale feature extraction, data utilization from different scopes, and hard negatives handling. We compare and summarize the common methods to address each problem.

Chapter 4 proposes our pedestrian detector. We exploit semantic feature fusion by combining the center keypoint of full body and the center keypoint of the visible part of each pedestrian. We proposes multiple methods to fuse features. We elaborate the pre-processing, network design, and post-processing of the detecting pipeline.

Chapter 5 displays the detailed experimental results and environmental settings. We compare multiple variants of our proposed methods with the baseline detector. We apply our trained detector to another new dataset directly to test the generalization of our model. We also compare our methods with other state-of-the-art detectors.

Chapter 6 concludes our work in both advantage and disadvantage aspects, and some future work directions to improve our methods are also given.

Chapter 2

Preliminaries

Pedestrian detection is an application of object detection so that deep learning-based detection frameworks can be used in both general object detectors and pedestrian detectors. In this chapter, we first explain the basic concepts and outline some classic object detection frameworks. Later, we analyze the inherent attributes of pedestrian detection and introduce datasets and evaluation metrics.

2.1 Object Detector Frameworks

We first introduce the basic concepts in the deep learning-based methods. We will overview the development of classic backbones before we classify and analyze detection frameworks from different aspects, including the detection stages and whether to use anchors.

2.1.1 Basic Concepts of Deep Learning-based Methods

Before diving into the frameworks of deep learning methods, we need to explain the basic terminology used in this thesis. The deep learning-based methods we introduce in this thesis are focused on CNN methods. A basic plain CNN model is stacked of multiple convolutional (Conv) layers that are optionally followed by pooling layers, activation functions, and normalization layers. Generally, there are fully connected (FC) layers at the end of the model. In this paper, we will introduce some high-level building blocks that consist of some specific construction patterns proposed by researchers. The 'building block' is a block unit that is used to build up a model.

In Conv layers, there are kernels, which are also called filters. The feed-forward operation in a Conv layer is to slide the kernel on the input image and make a linear trans-

formation by summing up the dot product at every location [151]. In pooling layers, the forward operation is used for down-sampling the input feature by using the representative feature in a region to replace the original feature [201]. Compared to the original input features before the pooling layer, the pooled features are usually more abstract and have a smaller size. The activation function works to add non-linearity and diversity to the CNN models [171]. The normalization layer [136] would rescale the input data according to the norm, which is of great benefit in helping the models converge during the training phase. In the FC layer, the input data will be dot multiplied with the FC layer's weight to reduce the dimensionality of the data. It is commonly used to generate a score for each class for classification tasks [147]. During the training phase, the gradients of the trainable weights are back-propagated according to the chain rule, and each iteration of trainable weights will be updated based on the gradients [151].

In this paper, when we say the number of layers in CNN models, we are referring to layers with trainable weights. The n×n Conv layer means a Conv layer with a kernel of size n×n. The number of trainable weights in a Conv layer is given by,

$$Num = C \times K \times H \times W, \tag{2.1}$$

where Num is the number of the total trainable weights, and $[C, K, H, W]$ are the kernels' dimensionality. C is the input channels, and kernels have the same number of channels with the input feature. K is the number of kernels. H and W are the height and width of each kernel, respectively. The hyper-parameters, such as the learning rate, are adjustable. They affect the experimental results. The hyper-parameters are adjusted by the researcher, not the model. We will not count the hyper-parameters as the trainable parameters.

2.1.2 Backbone Models

In this subsection, we will introduce some popular and representative backbone models, most of which are still adopted in the feature extraction step. We list the overview of the backbones in Table 2.1.

LeNet-5 [151] is an early-stage CNN model that was proposed in 1998, and it was proposed for handling digits recognition task on MNIST [151] dataset, which only contains 10 classes. Because of limitations in the computational power, neural network models were very small-scale at that time, and unable to handle hard tasks. Specifically, LeNet-5 only uses 6, 16, and 120 kernels for each Conv layer, respectively.

In 2012, AlexNet [147] came out and became the winner of the ImageNet Large Scale Visual Recognition Competition (ILSVRC)[1] 2012 in classification task. It has five Conv

[1] http://www.image-net.org/challenges/LSVRC/

Table 2.1: Overview of the existing popular backbones.

	Year	***# of layers***	***# of parameters***	***Top-5 error(%)***	***Highlight***
LeNet-5 [151]	1998	7	60k	N/A	LeNet-5 utilizes the model to learn the feature of the input data, and it is the first generation of the CNN model.
AlexNet [147]	2012	8	60M	15.32	AlexNet was trained on multiple GPUs.
VGG-16 [208]	2014	16	∼130M	7.33	VGGNet uses multiple 3×3 kernels to replace large kernels.
ResNet [124]	2015	18/34/50/101/152	0.2 to 1.7M	3.57	ResNet uses the identity mapping to help train the deep network.

[1] Top-5 error(%) is the results on ImageNet validation set (LSVRC) for classification task

layers and three FC layers. AlexNet was trained on two GPUs and is much larger-scale than previously due to the powerful computation ability of GPUs. The number of kernels for the five Conv layers are 96, 256, 384, 384, and 256, respectively. The computations are separated equally by two GPUs, and each one is responsible for half of the kernels' computations. The features computed by two GPUs are shared at the third Conv layer and two FC layers. The other layers will only use the output of the previous layer in the same GPU as the input. AlexNet has 60 million parameters, which is much larger than LeNet-5, and AlexNet is able to process large-scale datasets such as the ImageNet dataset [100], either large in sample amount or in data size. AlexNet is able to classify around 1000 classes in the ImageNet dataset, which is a huge improvement compared to LeNet-5, which can only classify 10 classes at that time.

VGGNet [208] came out in 2014, ranking first in the localization sub-task of ILSVRC 2014 classification+localization task, and second in the classification sub-task. One contribution of VGGNet is to use 3×3 kernels instead of using the large kernel size like 5×5 or 11×11. For example, two consecutive 3×3 kernels used in VGGNet can replace the receptive field of one 5×5 kernel, and this helps to reduce the number of parameters. More importantly, using multiple 3×3 kernels to replace one larger-size kernel will increase the neural network's depth, and the non-linearity of these multiple layers enables the network model to learn more complicated features and patterns. Another contribution of VGGNet is to design a deeper network by using more channels, which helps to precisely discriminate more sophisticated features. Taking the 16-layer VGGNet (VGG-16) as an example, it consists of 13 Conv layers and 3 FC layers, and the channels of Conv layers are in the range of [64, 512]. All the kernels are of the same 3×3 size in these 13 Conv layers, and contribute considerably to reducing the parameters of Conv layers. However, the three FC layers, especially the first connected layer, have too many parameters, resulting in a large amount of the parameters being used in VGGNet. Specifically, VGG-16 has more than 130 million parameters in the 16-layer deep neural network.

Proposed in [124], ResNet won the first place on the ILSVRC 2015 classification task. Theoretically, when we stack many layers and construct a very deep plain network, the worst case should be that the deep network has the same performance with a network with fewer layers, when the deeper layers in the deep network are identity mapping and generating the same output as input. He et al. in [124] did an experiment to compare the performance of two plain networks, one with a depth of 20 and the other one with a depth of 56. Unexpectedly, the performance of the deeper network has a higher train error and test error compared with the shallow network. However, it can be inferred that the degradation performance given by the experiment in [124] is caused by the gradient vanishing and exploding, which makes the deep model hard to converge. Based on this,

He et al. proposed a novel shortcut connection by adding the input directly to the output, which makes the network easier to be trained.

$$X_i = \mathcal{H}(X_{i-1}). \tag{2.2}$$

In Eq. (4.1), X_{i-1} is the output feature from $(i-1)^{th}$ layer and the input of i^{th} layer, X_i is the output from i^{th} layer, and layer $\mathcal{H}(x_{i-1})$ is the desired function that we would like to fit by stacking some layers. The residual function is defined as $\mathcal{F}(x_{i-1}) := \mathcal{H}(x_{i-1}) - x_{i-1}$, and thus the original desired function is formed as in Eq. (4.2):

$$X_i = \mathcal{F}(X_{i-1}) + X_{i-1}, \tag{2.3}$$

where $\mathcal{F}(x_{i-1})$ is much easier to approximate. This 'shortcut connection' is designed to deal with the gradient vanish and gradient exploding problems, and helps the deep CNN to converge during the training phase.

2.1.3 Benchmark Detection Frameworks

Detectors can be divided into different categories from different aspects, such as the number of stages, whether to use anchors in the network, etc. The classification of detectors can help readers to better understand the network structure.

Two-stage Detectors vs. One-stage Detectors

Nowadays, the detectors can be roughly divided into two branches, two-stage detectors (TSDs) and one-stage detectors (OSDs). The most representative TSDs are R-CNN [116], Fast R-CNN [115], and Faster R-CNN [195] family. YOLO [193] and SSD [160] are two classic OSDs. The main difference between TSDs and OSDs is whether to generate proposals before predicting the detection results.

Two-stage detectors: In late 2013, Girshick et al. proposed R-CNN [116], which is one of the first generation detectors. R-CNN's workflow is first to perform a selective search [224] on the entire input image and generate around 2000 regional proposals. Girshick et al. fixed all the proposals to the same resolution of 227×227 before feeding them to the neural network one by one and calculating the corresponding features of each candidate. Finally, a class-specific binary SVMs will be appended at the end to classify each proposal. Girshick et al. used the class-specific linear regression to fine-tune the bbox locations from the feature of $pool_5$ layer in R-CNN. With the SVM scores, a greedy-based non-maximum suppression (NMS) is applied to select the highest score among the candidates with the intersection-over-union (IoU) overlap larger than the threshold.

He et al. proposed SPPNet [123], claiming that CNN did not require the fixed-size input image because only FC layers require a fixed input. Based on this scope, He at al. first performed the selective search [224] to generate around 2k proposals and fed the original image to the CNN model to extract features from the full image. They removed the last pooling layer and proposed spatial pyramid pooling (SPP) layers, which took different sizes of the input and generated a fixed size output by combining the outputs from multiple pyramid pooling layers. Each spatial pooling layer divides the input feature with size of $a \times b$ into $m \times n$ bins, and each window is of size $a/m \times b/n$. SPP layers are spatial pooling layers with different scale of $m \times n$ bins, so that a different scale of features can be extracted for one proposal. Because SPPNet shares the parameters and uses the full image to go through CNN, unlike R-CNN which feeds around 2000 proposals to CNN one by one, the speed of SPPNet is more than 30 to 100 times faster than R-CNN.

However, one drawback of SPPNet is that it cannot be trained end-to-end. To cope with that, Girshick proposed Fast R-CNN [115]. The pipeline of Fast R-CNN is to generate object proposals first, and feed the network with the original full image and generated proposals. The feature extraction was done by the ConvNet from VGGNet [208], and Girshick innovated a pooling unit called ROI pooling and replaced the SPP layers of SPPNet [123]. The ROI pooling, which can be regarded a special case of SPP layer with a single scale, takes proposals of different sizes (e.g., $h \times w$) and divides them into a fixed $H \times W$ grid, and each grid of sub-window is in the size of $h/H \times w/W$. A simple max pooling will be applied, and the maximum value in each grid can be kept, then applied to each channel of the feature map. The fixed-size output from the ROI pooling layer can be applied to a sequence of FC layers. At the end, there are two parallel branches: the first one produces the softmax scores for each of the K classes and the background (total of K+1 classes), and this branch is used for classification task; the second one produces a set of 4 real values, indicating the coordination of the bbox, and this one is used for regression. Another benefit of Fast R-CNN is that the parallel branches of classification and regression tasks enable them to share parameters and further prove the regression task can benefit from the feature extracted from the Conv layer, which was only used to classify the objects before Fast R-CNN came out. The integration of multi-task losses unifies the training phase and improves the accuracy by using the softmax classifier instead of multiple binary SVMs.

A commonly used proposal generation method is selective search [224], which is not efficient. Ren et al. [195] carried out a mini-network named Region Proposal Network (RPN) to generate proposals. The proposals of RPN and Faster R-CNN [195] are a landmark contribution in the area of detection. The intention behind RPN is to improve the efficiency of the proposal generation step, since the most commonly used selective search

takes about 2 seconds to process one image. The RPN is a mini-network that takes the output of convolutional features and slides a small window on each location, then appends two sibling FC layers, one for regression task and the other one for object binary classification. The concept of 'anchor' is proposed in RPN, where the anchor is the center of each sliding window, and at each location, nine anchors that combine three scales and three aspect ratios are generated. The binary classification task for RPN only predicts whether a proposal belongs to any category in the dataset, and does not need to classify the category to which the proposal belongs. The output of RPN will be sent to Fast R-CNN to generate the final detection results. The usage of RPN not only improves accuracy but also shortens the inference time, compared to Fast R-CNN [115].

One-stage detectors: Although Faster R-CNN [195] achieves a promising performance in accuracy and takes less inference time than other previous TSDs; its speed is far from real-time. To achieve the real-time detection, one-stage detector YOLO-v1 [193] came out with a speed of 45 to 155 frames per second (FPS). YOLO-v1 divides the input image into $S \times S$ grids, and each grid predicts $\mathcal{B}$ bboxes and scores for $\mathcal{K}$ classes. Each bbox prediction result is a set of 5 values: $center_x$, $center_y$, w, h, and $confidence$. $center_x$ and $center_y$ are the relative distances from the center point of the bbox to the bounds of the grid in width and height, respectively. w and h are the relative width and height of the object, respectively. The $confidence$ score is defined as the production of the possibility score of the object and the IoU of the prediction and ground truth bbox. In other words, the $confidence$ should be zero if there is no object detected in the corresponding grid. In the experiment, the network predicts 2 bboxes and a total of 20 class scores for each grid, and each image is divided into $S \times S$ grid, where $S = 7$. The network has 24 Conv layers to extract features and 2 FC layers for prediction, where the last FC layer is designed to output a tensor of size $7 \times 7 \times 30$. The 7×7 comes from the grids and 30 comes from the length of the prediction of each grid (2×5 for bbox prediction + 20 classes scores). Because its structure differs from that of TSDs, OSD only needs to perform the classification once. However, most of the proposals generated in OSDs are negative samples, which may cause overwhelm during OSDs's training phase. Therefore, compared with TSD, YOLO-v1 is faster but less accurate.

SSD [160] is an OSD that takes the 'anchor' into network design. SSD takes VGG-16 [208] as the backbone. At each location of feature maps obtained from different layers, a set of default boxes with different aspect ratios and scales will be generated. The different-scaled feature maps enable the network to behave better to detect the multi-scale objects. The prediction of each default box consists of the offset and the class scores. When the input image size is 300×300, there are a total of 8732 predictions per class.

Anchor-free vs. Anchor-based Detectors

Anchor boxes are candidates for the region proposals generated by TSDs in the first stage, and they are also candidates for the final bboxes of OSDs. Anchor boxes are the boxes have different pre-defined scales and ratios that are generated at each sliding window location.

Anchor-based detectors: Since the introduction of RPN [195], more detectors tend to use pre-defined anchor boxes, such as SSD [160], YOLO-v2 [192], and RetinaNet [157]. The anchor boxes are a set of boxes with scales and ratios pre-defined by researchers, and they are usually generated by sliding window on the full feature map. The usage of anchor boxes consumes massive computational power because they generate multiple anchor boxes at each anchor point.

Anchor-free detectors: There are roughly two ways to achieve anchor-free detectors: dividing the images into grids like YOLO-v1 [193], and utilizing the keypoints of the object. For example, DeNet [223] used four corner keypoints of an object to generate the detection results instead of using anchor boxes. The proposal of DeNet [223] inspired the appearance of novel anchor-free detectors. Anchor-free detectors can help reduce complexity compared to anchor-based detectors that use more hyper-parameters, such as anchor ratios and anchor scales. CornerNet [150] took HourglassNet [173] as the backbone, which was designed for pose estimation and keypoint detection. The pipeline of CornerNet [150] was to predict two corner keypoints for each object instead of directly generating the anchor boxes by sliding windows. CornerNet produced two sets of heatmaps, embedding vectors, and offsets by using the corner pooling unit. These two sets of predictions are used to predict the top-left corner keypoint and the bottom-right corner keypoint for each object, respectively. The heatmap helps to denote the location of the object's corner keypoint, the embedding vector works to pair two corner keypoints that belong to the same object, and the offset helps to fine-tune the location of two corner keypoints. Taking CornerNet [150] as the backbone, Duan et al. proposed CenterNet [105], which utilized the center keypoint together with the top-left corner keypoint and bottom-right corner keypoint. CenterNet [105] achieved a higher mean Average Precision (mAP) than CornerNet [150] on MS COCO dataset [158]. Obviously, one reasons for this is the use of the center keypoint containing the internal feature of the object. Center and scale prediction-based detector (CSP) [162] underlined the importance of the center keypoint and obtained a promising accuracy on CityPersons dataset [249] by predicting the center of the pedestrians, the scales and offsets of the corresponding pedestrians, without using the corner keypoints.

Anchor-based and anchor-free methods fusion: Anchor mechanisms facilitate multi-scale detection by pre-defining multiple anchor box scales and ratios. Anchor-free detectors focus on semantic features of the target and reduce complexity. In order to

combine the advantages of anchor-based detectors and anchor-free detectors, the authors of [258] added two anchor-free sub-branches on the anchor-based detector RetinaNet [157]. The parameters of the anchor-based branches are frozen and kept unchanged during the training phase. They proposed an online feature selection method by finding the layer with the minimum sum of the classification loss and regression loss of instance i in FPN [156]. The layer with minimum loss was chosen to learn the instance i. The detection results of the anchor-based branches and anchor-free branches were merged together before NMS is applied, and the final detection results are generated. Anchor boxes are determined by locations and shapes. In most anchor-based methods, anchor boxes are generated by sliding windows across the entire image, such as RPN [195] and SSD [160]. Guided Anchoring Region Proposal Network (GA-RPN) [228] trains the model to predict the location of the anchor point by producing a binary result, and the width and height of the anchor box at the corresponding anchor point. Generally speaking, the anchor boxes are generated by anchor-free keypoint prediction methods. Consequently, GA-RPN is a fusion work of anchor-based detection and anchor-free detection methods.

2.2 Pedestrian Detection Datasets, Evaluation Metrics, and Inherent Attributes

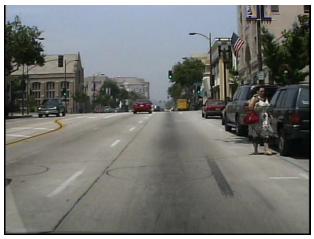
Datasets play an important role in training the deep learning-based models, and they have an impact on the models' generalization ability. To better evaluate the model's performance, the evaluation metric should be unified and reasonable. In this section, we will introduce some representative datasets and the evaluation metric before analyzing the inherent attributes of pedestrian detection.

2.2.1 Pedestrian Detection Datasets

The comparisons of some popular and representative datasets are listed in Table 2.2, including the ETH dataset [110], the Caltech dataset [102], the KITTI dataset [113], the KAIST dataset [134], the CityPersons dataset [249], and the EuroCity Persons (ECP) dataset [80].

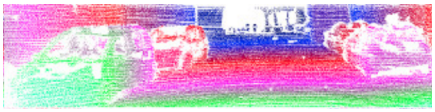
The ETH dataset [110] was proposed in 2007, and its image resolution is very low. It is a small-scale dataset that only contains 2303 video frames. The video was collected by a camera mounted on a stroller. Due to the small size of the dataset, it is now commonly used as a test dataset to test models' generalization abilities without pre-training on the ETH train set.

(a) The ETH dataset.

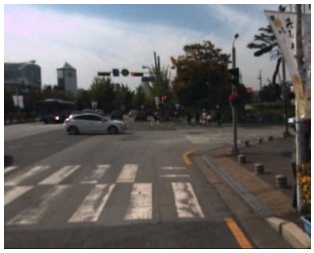


(b) The Caltech dataset

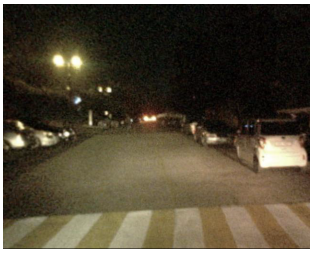
(c) The KITTI dataset.



(d) The KITTI dataset (3D data).



(e) The KAIST dataset (day).



(f) The KAIST dataset (night).

(g) The CityPersons dataset.

(h) The ECP dataset (day).

(i) The ECP dataset (night).

Figure 2.1: Examples for each dataset listed in Table 2.2.

Table 2.2: Some comparions of most commonly used pedestrian datasets.

	ETH	***Caltech***	***KITTI***	***KAIST***	***CityPersons***	***ECP***
# of images	2303	249,884	14,999	95,328	5000	47,335
# of persons	$\sim 14,000$	$\sim 347,000$	$\sim 12,000$	103,128	35,016	$\sim 238,200$
persons per image	6.1	1.4	0.8	1.1	7.0	5.0
resolution	640×480	640×480	1240×376	640×480	2048×1024	1920×1024
train-val-test split (%)	20/0/80	50/0/50	50/0/50	50/0/50	60/10/30	60/10/30
# of seasons	1	1	1	1	3	4
weather	dry	dry	dry	dry	dry	dry&wet
# of cities	1	1	1	1	27	31
# of countries	1	1	1	1	3	12
day / night	day	day	day	day&night	day	day&night
occlusion tags	×	√	√	√	√	√
# of pedestrian labels	1	3	3	3	5	3
ignore region	×	√	√	√	√	√
year	2007	2009	2012	2015	2017	2019

The Caltech dataset [102] [103] was proposed in 2009, and it is the first large-scale pedestrian dataset, with around 10 hours of video captured by a driving vehicle's dash camera. In the Caltech dataset, the researchers annotated the person with the occlusion ratio. They labelled them 'Person' for the individual pedestrian, 'People' for a group of pedestrians, and 'Person?' for an object that cannot be clearly defined as a person. In addition to bbox annotations for the full body, they also provide the bbox annotations of the visible body part for the corresponding pedestrian, which enables researchers to take advantage of this extra information in the network design. The authors proposed the 'ignore' region, and whether or not these ignored regions are detected, the accuracy of the detector will not be different.

The KITTI dataset [113] was published in 2012, and it provides the 3D object detection data generated by using the light detection and ranging (Lidar) method. Lidar utilizes the stereo cameras, 3D laser scanner, Edmund Optics lenses, and GPS navigation system. Object detectors that are designed to detect multi-objects in traffic scenarios can use the KITTI dataset because its images are the real traffic scenarios, and it labels the different types of vehicles together with pedestrians.

The KAIST dataset [134] came out in 2015, and also supports 3D detection. The dataset has the multi-spectral images that use two types of images as a pair: RGB and thermal images. The KAIST dataset offers the data in the night, which enlarges the

diversity of the data. It makes researchers think about how to deal with the night vision detection task.

The CityPersons Dataset [249] came out in 2017, and the images of this dataset are in a large resolution 2048×1024 that can benefit the models with more spatial information. The dataset has the training, validation, and testing splits. The image data was collected in 3 seasons, 27 cities, and 3 countries. This dataset also offers semantic information and a bbox of the visible part for a pedestrian if he/she is occluded. The authors labelled the samples into five sub-categories: pedestrian, rider, sitting person, other person, and people group. The authors denoted the hard negative samples as 'ignore', and this information can be taken into training. This dataset also has a larger person per image ratio than most datasets, offers more samples in each image, and is more challenging.

The ECP dataset [80] was published in 2019, which covers all the seasons, dry and wet weather conditions, and daytime and night vision in the dataset. The ECP dataset was collected in 12 countries which can help the dataset have a better diversity.

2.2.2 Evaluation Metrics

We will introduce the basic evaluation metrics in the detection area, then give the evaluation metrics specifically for the pedestrian detection area.

True positive (TP) is the target correctly detected by the detector. True negative (TN) is the background (non-target) correctly classified as the background by the detector, which means it is not in the detection results. False positive (FP) means the detector detects the background as a target. In the pedestrian detection area, this is very common, since the person reflected in the window and the model on the billboard have a high similarity with the real pedestrians in appearance. False negatives (FN) happen when the detector should detect an object but does not and causes a miss in detection results. In pedestrian detection, this frequently happens when the pedestrian is very far away, or the pedestrian is occluded or deformed.

Precision (PR) is the ratio of TP compared to all the detection results, given by

$$PR = \frac{TP}{TP + FP}. \tag{2.4}$$

Recall (RC) is the ratio of TP to all the targets including detected and not detected as given by

$$RC = \frac{TP}{TP + FN}. \tag{2.5}$$

Miss rate (MR) is the ratio of FP to all the potential detection targets, given by

$$MR = \frac{FN}{TP + FN} = 1 - RC. \tag{2.6}$$

False positive per image (FPPI) is the ratio of the FP to all detected objects, given by

$$FPPI = \frac{FP}{TP + FP} = 1 - PR. \tag{2.7}$$

Average Precision (AP) is the averaged PR for the same class category, and mAP is the mean of AP for every class category. AP and mAP are usually commonly used in general object detection that has many class categories.

In the pedestrian detection area, the evaluation metric that is commonly used to evaluate the pedestrian detector's performance is MR^{-2} (the lower, the better), which was introduced in [102]. MR^{-2} is the mean value of nine derived miss rates, with the corresponding FPPIs evenly located in $[10^{-2}, 10^{0}]$ within the log-space. The miss rate and FPPI are calculated by varying the threshold of detected bbox confidence score. When we lower the threshold, more detection results are taken, so there are more false positives but fewer chances to miss the object. Thus, when the FPPI increases, the miss rate decreases.

2.2.3 Inherent Attributes of Pedestrian Detection

In the early stage of the deep learning methods' appearance, the traditional machine learning methods still serve as the mainstream of research directions. There are many traditional methods to design the hand-crafted feature and leverage these pre-defined features, such as [95] and [250]. Some researchers hybridize traditional hand-crafted features with the features learned from deep learning methods, such as [239]. The accuracy performance is greatly increased when the researchers adopt the deep learning-based methods and detect the features learned by the CNN models. Pedestrian detection, as an application of object detection, has gained significant attention in recent years [172, 184, 83, 218, 128, 251, 180, 183, 202, 28].

Pedestrian detection has common characteristics with object detection, but also has unique attributes. We summarize the most common inherent attributes of the pedestrian detection task in Fig. 2.2. Occlusion happens very often, and the pedestrians can be occluded by the background, such as other vehicles, buildings, and mailboxes on the road. People are social animals, so they usually stay with other persons, which could cause the occlusion issue in the crowd.

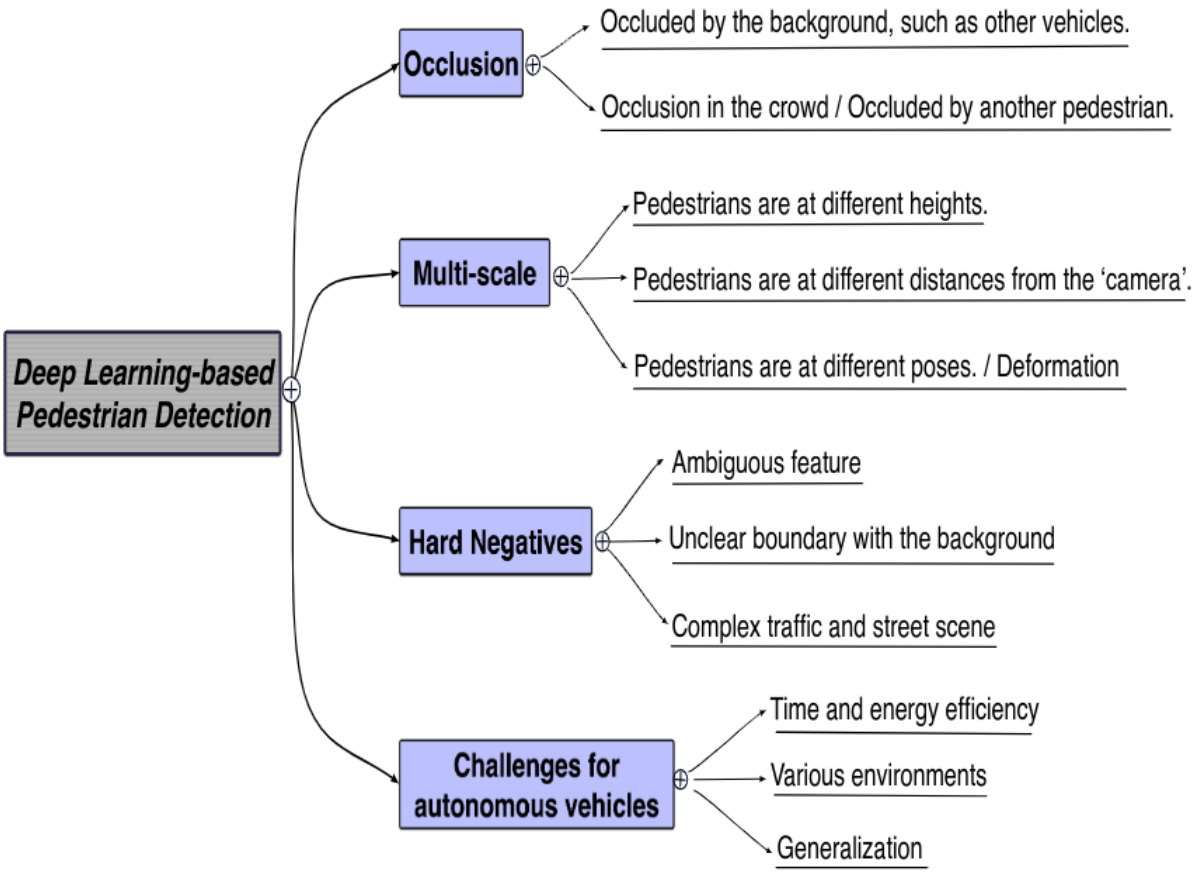


Figure 2.2: The inherent attributes and major challenges of pedestrian detection.

The pedestrians are at different heights and at different distances from the vehicular camera. They may be in different poses, which causes a multi-scale handling problem. It is very common that an image contains pedestrians of different scales. The accuracy of pedestrian detectors varies significantly over scales, which cause a major bottleneck in this area. Specifically, it is hard to discriminate the small-scale pedestrians from the low resolution feature maps. In addition, the bbox aspect ratio is fixed at 0.41 in the pedestrian detection task, which is different from flexible aspect ratios in general object detection.

Pedestrian detectors should be able to successfully detect people in complex traffic scenarios, and features of a background may be highly similar to the pedestrians'. This type of background is a hard example that the deep learning-based detectors may not be able to precisely distinguish. Similarly, a pedestrian sample will be considered a hard example if the detector cannot successfully detect this pedestrian after attempts.

Pedestrian detection is an essential component of the pedestrian protection system that provides information to the autonomous vehicles and other participants of the ITS, and it has many challenges. Time and energy efficiency, stable precision in various environments, and generalization ability in practical use are the most urgent challenges that it must overcome to ensure the safety of pedestrians and passengers.

Table 2.3 lists the optimization methods of deep learning-based detectors in different development steps in order to provide the readers with a clear overview of existing optimization techniques.

Table 2.3: Optimization methods overview.

Pre-processing	Data processing	Data cleaning, data transformation, data integration, and data normalization.
	Data augmentation	Geometric transformation, flipping, color shifts, rotation, and cropping.
		Deep learning method: GAN [112] [104]
Model Construction	Convolutional methods	Group convolution [237], dilated convolution [243], and various convolution blocks.
	Normalization layers	Batch normalization [136], group normalization [235], and layer normalization [24].
	Pooling layers	Average/ Max/ Overlapping/ Spatial [123]/ ROI [115]/ Corner pooling [150]
	Attention mechanism	Attention-domain (spatial [138], channel [130], and mixed domain [234]), source-target relationship (self-attention [88] vs. guided-attention [186])
	Regularization	Addressing overfitting (e.g., dropout [126] and weight decay)
	Others	Loss function, gradient optimizer, and activation function, etc.
Post-processing	NMS	Greedy-based NMS: basic NMS, soft-NMS [29], softer-NMS [125], adaptive-NMS [159], and ID-NMS [141]
		Learning NMS and NMS network.
	Other	Markov Random Field, etc.

Chapter 3

Related Work

In this chapter, we will analyze what hinders the high accuracy performance of pedestrian detection, and introduce the representative deep learning-based pedestrian detectors from the aspect of occlusion handling, multi-scale feature extraction, data utilization from different scopes, and hard negatives handling.

3.1 Occlusion Handling

It is very common that pedestrians are under occlusion in the captured images, and it is reported that more than 70% of pedestrians are occluded in the CityPersons dataset [249]. The occlusion of pedestrians increases the difficulty of detection. If a detector is trained to learn the partial features of pedestrians, it may result in more false positives if the detector considers that other things fit the partial features of pedestrians. However, if the detector is not designed to handle the occluded pedestrians, the detector will have a high rate of missed false negatives. Some representative methods to address occlusion are listed in Table 3.1.

3.1.1 Part Information Benefits the Occlusion Handling

Part-based detectors such as [182] and [164] work to detect various types of features of the partial region for each pedestrian, and they are the mainstream methods for improving the detection accuracy of the occluded pedestrian. Some part detectors are designed to only detect a specific part of the human body, such as the head detector [225]. [256] jointly learned the part detectors to investigate their correlation further. The part detectors can be integrated together to improve the full body human detection of partially

Table 3.1: Representative pedestrian detectors for occlusion handling.

Method	***Backbone***	***Framework***	***Anchor***	***Highlight***
DeepParts [219], 2015	AlexNet	AlexNet	×	Trained multiple part detectors by the data and chose the top-6 detectors to fuse the detection results during the inference step.
PCN [230], 2018	VGG-16	Faster R-CNN	√	Utilized three parallel branches– basic detection branch, part semantic branch, and context branch.
Occlusion-aware score [174], 2018	-	one-stage detectors	√	Designed a mini-net to generate a final confidence score based on part scores, and can be applied to all anchor-based one-stage detectors.
Bi-box [257], 2018	VGG-16	Fast R-CNN	×	Fused the full body bbox and the visible part bbox.
RepLoss [233], 2018	ResNet-50	ResNet	√	Proposed a loss function that considered the crowd scenes.
OR-CNN [253], 2018	VGG-16	Faster R-CNN	√	Proposed a loss function for the crowd group.
adaptive-NMS+RFBNet [159], 2019	VGG-16	RFBNet	√	A novel NMS that produced a dynamic threshold according to the predicted density.

occluded pedestrians, and the authors proposed 20 pre-defined types of part pools with the prior knowledge. DeepParts in [219] trained 45 part pool prototypes and constructed an automatic data-driven part selection, the fused score of the complementary parts was used for the final detection results. [181] was proposed to handle the feature extraction, deformation handling, occlusion handling, and the classification jointly in one CNN model to find the correlation among these tasks. One-stage detectors such as SSD [160] and YOLO-v2 [192] output the fixed-size tensors, which contain bbox locations and classification scores. In [174], Noh et al. proposed the occlusion-aware detection score that utilized the predicted part scores to formulate the final confidence scores. They first max-pooled the part scores on the part confidence map, and fed the max part score to multiple occlusion weighting masks, which were end-to-end trained. A score weight was then generated by a hidden layer and ReLU layer, which was finally used to re-scale the predicted confidence score in the output tensor.

Bi-box regression [257] (Bi-box) utilized the visible part information. The motivation of Bi-box is to deal with the occlusion problem by using the model to predict two bboxes for each pedestrian: one is the full body bbox, and the other is the visible part bbox. Thus, Bi-box can simultaneously perform the pedestrian detection and the occlusion estimation for each pedestrian. The Bi-box model took the VGG-16 [208] as the backbone in the feature extraction step, and two parallel sub-networks are used in the detection head: one for the full body bbox prediction, and the other for the visible part bbox prediction. P represents a proposal generated in the first stage, and G represents the ground truth annotation. P is matched to G if Eq. (3.1) is met, where F is the predicted full body bbox, and V is the predicted visible part bbox. $C(P, V)$ is the ratio of the area of V covered by P.

$$IoU(P, F) \geq \alpha \; and \; C(P, V) \geq \beta, \tag{3.1}$$

During the experiment, the hyper-parameters $\alpha = 0.5$ and $\beta = 0.5$ give the best result. The authors considered the detection results as negative if $IoU(P, F) < 0.5$. Final detection results are produced by fusing confidence scores that generated by two parallel branches. Designing parallel branches in the detection head is effective. For example, PCN [230] utilized two extra parallel branches besides the basic branch: one is the part semantic branch, which adopts the LSTM method for the information communication; the other is the context branch to select the region scales.

3.1.2 Addressing the Occlusion Caused by the Crowd

Pedestrians often gather together, and sometimes the occlusion is caused by other pedestrians. The authors of [217] came out with a double-pedestrian detection method, which

focused on the occlusion pattern that occludes by each other. Wang et al. claimed that many occlusion cases are actually caused by the crowd in [233], where one pedestrian is occluded by other pedestrians. After applying NMS, detection results that are close to each other may be eliminated because of a high IoU. To cope with the occlusion in the crowd, the authors of [233] proposed a novel regression loss function named Repulsion Loss (RepLoss) that inspired by magnets' attraction and repelling attributes. The regression loss function is given by,

$$RepLoss = L_{att} + \alpha L_{RepGT} + \beta L_{RepBox}. \tag{3.2}$$

In Eq. (3.2), L_{att} is the attraction term to enable the proposal to approach its target, L_{RepGT} is the repel term to make the proposal repel other surrounding ground truth targets except its own target, L_{RepBox} is another repel term to enable the proposals for different targets far away, and can make the detection results less sensitive to NMS. $\alpha = 0.5$ and $\beta = 0.5$ gives the best result in the experiment.

Zhang et al. proposed OR-CNN in [253] and aimed to improve pedestrian detection under occlusion. They proposed a novel loss function called aggregation loss function (AggLoss) and a new part occlusion-aware RoI pooling unit. The proposed two-stage detector OR-CNN adopts Faster R-CNN's framework [195]. The motivation of the AggLoss is to ensure that the anchors that match the same ground truth target can be close to each other. The AggLoss consists of the SmoothL1 regression loss and a β-balanced compactness loss L_{com}, which is defined as follows:

$$L_{com} = \frac{1}{N_{com}} \sum_{i=1}^{p} \Delta(t_i^* - \frac{1}{sumA_i} \sum_{j \in A_i} t_j), \tag{3.3}$$

where N_{com} is the number of ground truth objects that are assigned with more than one anchor, t_i^* is the ground truth object assigned with multiple anchors, $sumA_i$ is the number of anchors assigned with the i-th object, and t_j is the predicted bbox j. The working flow of the part occlusion-aware ROI pooling unit in [253] is to divide every proposal into five pre-defined hand-crafted parts, each of which is fed into the ROI pooling to generate the feature. These five individual features F_{in} are sent to five corresponding occlusion processing units to predict visibility scores, and the scores will be fused with corresponding input features F_{in}. The final feature is obtained by adding the outputs from these five occlusion processing units to the proposal's after-pooling feature, and following classification and regression tasks are performed on the final feature.

Adaptive-NMS [159] was introduced in 2019. The authors claimed the NMS is not an optimal solution in some conditions, especially in crowd scenes, because every target is close to the others and has a high overlap. Adaptive-NMS dynamically improves the

threshold for the NMS when a predicted pedestrian is in a crowd. This is achieved by an additional parallel sub-network in the detection head, which can predict the density. The density's definition is given as follows:

$$d_i := \max_{b_j \in \mathcal{G}, i \neq j} IoU(b_i, b_j), \tag{3.4}$$

where $\mathcal{G}$ is a set of ground truth bboxes, and the density of the object is defined as the maximum IoU with other objects in the ground truth set. The dynamic threshold of Adaptive-NMS is the maximum of the density of the crowd or the pre-defined threshold. The rest of the Adaptive-NMS is based on Soft-NMS, which will reduce the confidence score for the detection results instead of eliminating it.

3.1.3 Comparisons and Conclusions

The most common occlusion handling method is the part-based methods that we introduced in Section 3.1.1. This type of method utilizes partial features, which can be visible or representative parts of the pedestrian. Another type of method in Section 3.1.2 is to deal with the occlusion caused by other people in the crowd and take into consideration the person's relationship to the group. In Section 3.3.3, we will introduce methods that generate occlusion during the data preparation step, and help the model learn how to handle the occlusion. In Section 3.4.2, we will discuss the attention mechanism-based methods to force the detector to focus on specific regions, such as the occlusion region-based attention network introduced in [252].

From evaluation results provided in [233] [253] [159], the detection performance of occluded pedestrians are far from promising with such a high miss rate, and this can be very dangerous. It is hard to improve detection accuracy and eliminate the occlusion problem from existing methods. Taking the Internet of Things (IoT) and vehicular networks into consideration can innovate the new mode to detect the pedestrians in traffic scenarios. For example, the cellphones carried by pedestrians can work as sensors, which can work with distances detected by the Lidar or other 3D detection methods, and the integrated information will enable autonomous vehicles to figure out the pedestrians that are occluded by obstacles. In [185], the authors proposed an occlusion-aware sensor fusion framework to enable information obtained by different vehicles to be shared to predict pedestrians crossing the road.

3.2 Multi-scale Feature Extraction

The pedestrians far away from the camera, the little children, and the sitting persons are smaller in scale and occupy a smaller number of pixels in the image. The small-scale pedestrians are very easy to be ignored by detectors; they do not have a clear boundary with the background, and the location accuracy may be very low, since a minor offset can cause a dramatic shift for the small-scale object. An image often contains pedestrians of different scales, and these multi-scale pedestrians have different proportions on a feature map, which can cause difficulties in designing the network and setting anchor box scales. An overview of the methods that we will introduce to address multi-scale feature extraction is listed in Table 3.2.

Table 3.2: Representative pedestrian detectors for addressing multi-scale feature extraction.

Method	***Year***	***Backbone***	***Framework***	***Anchor***	***Highlight***
RPN+BF [195]	2016	VGG-16	RPN	√	Used boosted forest to replace the second stage of Faster R-CNN.
MS-CNN [82]	2016	VGG-16	Faster R-CNN	√	Performed the detection at different levels to generate multi-scale proposals.
SADR [259]	2016	VGG-16	Faster R-CNN	√	Used different levels of features to regress the pedestrian's location according to the height.
SAF R-CNN [154]	2017	VGG-16	Faster R-CNN	√	Gated function of two sub-branches, one for large scale objects and the other for small objects.
TLL [209]	2018	ResNet-50	ResNet	×	One-stage network that predicted the top-bottom point and the topological line, increasing the performance of small-scale pedestrians.
ALFNet [161]	2018	ResNet-50	ResNet	√	One-stage detector, cascaded the prediction blocks at four feature levels by higher IoU thresholds to generate fine results.

After the publication of the general object detector Faster R-CNN [195], Zhang et al. found that Faster R-CNN did not perform well when applying to pedestrian detection in [248]. However, Zhang et al. found the RPN could perform well alone. The reason why adding the second-stage classifier would harm the performance might be the low resolution of feature maps, which could affect the detection of small-scale objects. To address this, Zhang et al. took the first-stage network– RPN as the backbone to obtain the bbox, confidence scores, and features. The outputs are fed into the cascaded boosted forest classifier to eliminate the confusion of the hard background instances. The RPN+BF method uses the single ratio anchors (0.41) with 9 different scales to deal with the multi-scale pedestrians.

To address the multi-scale feature extraction, using multi-scale inputs is a widely-used but cost-intensive method. Another method is to utilize different levels of features. Higher-level features contain rich semantic information, but the spatial location is not accurate because of the low resolution of the feature map. Lower-level features are more precise in terms of locating. In [259], the authors proposed Scale Adaptive Deconvolutional Regression (SADR) Network, which works to flexibly choose the feature from different levels according to its height to regress the target's location. Similarly, Feature Pyramid Network [156] also took advantage of multi-scale features from different levels. Feature maps from different levels are fused together before and after the deconvolution operations for final detection. The Active Detection Module (ADM) was introduced in [254], which utilized a series of coordinate transformation actions to obtain accurate pedestrian locations. SAF R-CNN [154] was designed to cope with multi-scale pedestrian detection by taking two parallel sub-networks after generating the proposals; the first is large-size sub-network, and the other is small-size sub-network. Both of these two sub-networks predict confidence scores and bbox locations. These two sub-networks were applied to a scale-aware weighting layer (SAWL) proposed by authors. The mechanism is that if the object is large, the SAWL weight for the large-size sub-network is expected to be high, while the SAWL weight for the small-size sub-network is low. SAWL works as a gated function. The SAWL weight for the large-size sub-network is given by,

$$W_{large} = \frac{1}{1 + \alpha exp^{-\frac{\Delta h}{\beta}}}, \tag{3.5}$$

where α and β are trainable parameters, Δh is the difference between the height of the object and the averaged height of all objects. The W_{small} can be obtained by $1 - W_{large}$. The SAWL works as a soft activation, and the final detection results will depend more on one of the sub-networks' outputs according to the predicted W_{small} and W_{large}.

In contrast to previously mentioned methods like multi-scale inputs and multi-scale features fusion, Cai et al. in MS-CNN [82] utilized multiple detectors at different levels.

Similar to Faster R-CNN, MS-CNN has two sub-networks. The first sub-network is a multi-scale object proposal network, and Cai et al. performed the detection at four branches at different levels. Each branch can only detect objects with one scale by sliding anchor windows. The second sub-network is an accurate detection network. Cai et al. used a deconvolution layer to upsample the feature maps in order to increase the accuracy of small-scale object detection. Liu et al. proposed ALFNet [161], which is a cascade one-stage detector that performs detection at four different feature levels to utilize multi-scale feature maps. At each level of feature maps, three Convolutional Predictor Blocks (CPB) are stacked and cascaded. Each CPB works to translate anchor boxes to detection results. In the experiment, more anchor boxes with higher IoU were sent to the later CPB, which proved the effectiveness of ALFNet in improving the accuracy of bboxes' locations.

By using keypoints and a topological line to detect the pedestrian, TLL [209] was proposed to enhance the detection performance of small-scale pedestrians by Song et al., where TLL stands for the topological line localization. TLL is a one-stage anchor-free detector. Song et al. took the ResNet-50 [124] as the backbone and concatenated the deconvolutioned feature maps of Conv3, Conv4, and Conv5 layers. The concatenated feature maps are from different levels, and can enrich the feature semantics and the resolution. There are three parallel branches in the detection head, each with a 1×1 Conv layer to produce the top point, bottom point, and the topological line for each pedestrian, respectively. The bbox can be generated by combining the information extracted from the top-bottom topological line. Song et al. adopted the Markov Random Field (MRF) in the post-processing step to improve the detection accuracy.

3.2.1 Comparisons and Conclusions

There are various methods for coping with the scale variation, and the most common attribute is to share and fuse the features from different levels such as [142] and [156]. MS-CNN [82] and ALF [161] perform the detection operations on multiple feature layers. SADR [259] and SAF-RCNN [154] can flexibly select features from different branches or different levels according to the height of the pedestrian, thereby producing corresponding results. Other methods, such as re-annotating the pedestrian by the keypoints and the topological line [209] and the scale-aware attention mechanism (e.g.,GDFL [155]), will be introduced in Section 3.3.2 and Section 3.4.2.

As for the experimental results provided in their work, the two parallel subnets of SAF R-CNN [154] perform well in detecting large objects, but the detection accuracy is not significantly improved when detecting small-scale objects. TLL [209], which uses another

format of annotations to denote the pedestrians, has an obvious improvement in small-scale pedestrians' detection on the Caltech dataset. The small-scale pedestrians can be children or persons sitting at a very near distance, or adults standing further away. For autonomous vehicles, it is more urgent to detect the small-scale pedestrians at near and medium distances, which may require depth information as an aid. In traffic scenarios and street scenes, sitting persons usually sit on chairs or benches, riders come with their vehicles, and children have a significant opportunity to stay with the adults. Taking the surrounding background information and spatial context features into consideration can benefit the detection performance.

3.3 Multi-scope Data Utilization

In addition to using the semantic and part information of each pedestrian, there are some other different scopes to utilize the dataset. For example, the traditional detection task is to denote the pedestrian with the bbox, Song et al. re-annotated the pedestrians by topological lines in TLL [209]; a novel way is to detect the keypoints of pedestrians, and bboxes can be transformed from the detected keypoints. This type of method belongs to the anchor-free detector, which was mentioned in Section 2.1.3, and we will introduce other information and features researchers utilized to denote pedestrians besides the keypoints in Section 3.3.2. Re-annotating datasets with information that benefits models to classify and detect pedestrians is an effective method that will be introduced in Section 3.3.3. An overview of the representative methods we will introduce in this Section is listed in Table 3.3.

3.3.1 Exploiting Semantic Features

The semantic information, such as human-annotated semantic labels and the pixel value for semantic segmentation task, can contribute to other related tasks such as detection. Some works have been designed to aid pedestrian detection performance by the semantic information, such as [220] and [96].

The detection can locate each object but have no idea about the clear boundary; the semantic segmentation task can extract the boundary through the pixels, but have a hard time figuring the object instances that belong to the same class. Simultaneous Detection and Segmentation (SDS) [121] was introduced in 2014, which combined the general object detection task with the pixel-level segmentation task. SDS has two pathways: path A operates on the cropped image of the object, and path B operates on the foreground of the

Table 3.3: Overview of the representative pedestrian detectors in Section 3.3.

Method	***Year***	***Backbone***	***Framework***	***Anchor***	***Highlight***
SDS[121]	2014	-	R-CNN	$\surd$	Fused object detection with segmentation task by using two parallel pathways: one operated on the cropped image, and the other operated on the region foreground.
SDS-RCNN [81]	2017	VGG-16	Faster R-CNN	$\surd$	Fused the semantic segmentation task in the network to aid the detection task, and the feature sharing can help the model train.
CSP [162]	2019	ResNet-50	ResNet	$\times$	One-stage detector based on the center keypoint and scale.
CSID [141]	2019	DLA-34	ResNet	$\times$	A novel NMS methods: ID-NMS, which took both identity and density of the pedestrian into consideration.
PedHunter [89]	2019	ResNet-50	Faster R-CNN	$\surd$	Randomly occluded one part of the pedestrian during the training step.

image in path A. These two parallel pathways do not share parameters, and training them as a whole directly produced better results than training them separately. The outputs of these two pathways were concatenated together to produce the final detection results.

SDS-RCNN [81] was proposed in 2017, which also combined pedestrian detection with segmentation tasks. SDS-RCNN was designed to improve pedestrian detection performance by taking advantage of segmentation supervision. SDS-RCNN has two sub-networks: the first network is RPN, and the second one is a binary classification network (BCN). The authors proposed the Segmentation Infusion Layer (SIL) and applied it to both sub-networks. This SIL only has a single layer and a 1×1 kernel to produce binary results – pedestrian samples and background. With the rough segmentation information transformed from bbox annotations, the authors generated the ground truth semantic mask $S \in R^{W \times H}$ and assigned the pedestrian at S_i with value 1, the background at S_i with value 0. The softmax loss function was used for the SIL. SILs share the parameters with the main network, and SILs can assist and benefit the shared parameters to be less sensitive in a natural way during the training phase.

Other existing methods, such as using the attention mechanism to fuse semantic features to the detection bboxes can make the detector focus on the trained 'attention' regions, which will be introduced in Section 3.4.2.

3.3.2 Using Keypoint Sets to Replace Bounding Boxes

It is pointed out that the bbox cannot represent the content feature of the object, while the keypoints are able to in [150] and [105]. The keypoint-based anchor-free detectors have recently become very popular. The aforementioned TLL [209] is an example of the anchor-free pedestrian detector, which is achieved by predicting the top and bottom keypoints and the topological line of each pedestrian to replace the traditional bbox. The anchor-free detector has some advantages, such as relieving the multi-scale feature problem by predicting the keypoints in the heatmap, rather than using anchor boxes with different scales. Thus, TLL does not need to consider the multi-scale anchor boxes and bboxes.

CSP [162] is a one-stage anchor-free detector proposed by Liu et al., which utilized the center keypoint and the scale of the pedestrian to make the prediction. The CSP model took ResNet-50 as the backbone and concatenated the feature maps from different levels to benefit the multi-scale object detection performance. The final feature map is applied to three parallel branches, which output center keypoints, scale predictions, and regression predictions through separate 1×1 Conv layers, respectively. Liu et al. changed the ground truth pedestrian annotations into three 2D masks, which contain center keypoint locations,

scales of the pedestrians, and offsets. The ground truth center keypoints are obtained by using the 2D Gaussian function, and the pedestrian center area with a radius of 2 in the mask is assigned with positive value 1 to indicate that is the center of a pedestrian. The visualized center keypoint mask is a heatmap of pedestrians on the corresponding image. On the scale mask, the log of the width and height for each pedestrian is assigned at the corresponding pedestrian's center location. In the area of pedestrian detection, the bbox ratio is fixed 0.41, which indicates the width of the bbox can be obtained from the height and vice versa. Liu et al. did an ablation study to find what is best choice among using height, width, height+width as the 'scale', and the experimental results indicated that using 'height' can produce the best accuracy. In the training phase, the center keypoint prediction is treated as a classification task and applied with the Focal Loss [157], and the height prediction and bbox offset prediction are treated as regression tasks by using the Smooth L1 loss. To convert these outputs into bbox results, take the centerness confidence scores higher than the threshold on the center keypoint mask as the center points of pedestrians, and multiply the predicted height of the corresponding pedestrian on the scale mask by the bbox ratio 0.41 to obtain the bbox width. The corresponding predicted offset could refine the bbox locations in horizontal and vertical directions.

CSID [141] was published in 2019 based on the working pipeline of CSP [162] and took [244] as the backbone. The authors found the Adaptive-NMS had a good impact on the crowd scene. However, it may not have been able to suppress the bbox for the same person. The authors proposed ID-NMS, which also considered the person's identity compared to Adaptive-NMS. The density value was assigned to the positive region of each pedestrian, which is a 2×2 area on the downsampled feature map. The authors took the euclidean distance in embedding spaces to denote the distance between two bboxes, which could further represent the identity information on the ID-Map. The purpose of ID-NMS was to design a usage condition to flexibly choose whether to use adaptive NMS or the original greedy-based NMS. When the distance between the highest-scoring bbox and other predicted bboxes is greater than the pre-defined identity threshold, the NMS threshold will be calculated according to Adaptive-NMS [159]: choose the larger one between the density value and original NMS threshold. Otherwise, when the distance is smaller or equal to the identity threshold, the NMS threshold will be used for post-processing. The intuition behind the ID-NMS is to find if the surrounding bboxes belong to the same identity. ID-NMS will choose to use the original NMS to suppress the bboxes blindly if the bboxes belong to the same pedestrian. Otherwise, ID-NMS will use Adaptive-NMS which could generate a dynamic threshold to preserve bboxes belong to different identities during the post-processing step.

3.3.3 Expanding the Data

Data augmentation is one of the widely-used methods to enlarge the amount of data during the training phase. Its basic operations include flipping, randomly cropping and rotating the images to increase the diversity of the dataset, in order to increase the generalization ability of the CNN model. Generative Adversarial Networks (GAN) is unsupervised learning, which can be trained to learn the distribution of the given data and produce similar data. GAN can also be adopted for data augmentation, as seen in [112] and [104]. In [232], Wang et al. adopted the GAN to occlude objects and make them difficult to classify. This increased the likelihood of objects being occluded in the dataset and facilitated the model to learn occluded hard examples. Wang et al. designed Adversarial Spatial Dropout Network (ASDN) in [232], which applies an occlusion mask according to the object and sets the feature value in the occlusion mask to zero. ASDN is trained to learn where occlusion makes recognition difficult, and enables the CNN model to learn more difficult occluded examples. PedHunter [89] was published by Cheng et al. in 2019, which utilized the occlusion-simulated data augmentation. Cheng et al. divided the pedestrian samples into five parts (i.e., the head, the left-upper body, the right-upper body, the left-bottom body, and the right-bottom body) and randomly occluded one of the four parts except the head of the pedestrian. The pedestrian samples are only occluded and fed into the model during the training stage. Besides, Cheng et al. annotated the head part of each pedestrian in order to guide the model to learn the features in the mask-guided module.

3.3.4 Comparisons and Conclusions

After observing the experimental results, we find feeding the semantic features into the model could aid the detection performance. Re-annotating the pedestrian with the keypoints enables the model to have a better representation ability and learn the essential features of the pedestrian. Keypoints including the center point and corner points are widely used. Novel keypoints combinations that can better represent pedestrians' features might improve the accuracy of pedestrian detection. Data augmentation is an effective step to increase the diversity of the dataset and enhance the detection performance, and it can be achieved by training GAN models.

3.4 Hard Negatives Processing

Hard negatives are the samples that detectors may regard as positive samples. A large amount of negative samples can overwhelm the training phase, because the negative sam-

Table 3.4: The overview of the representative detectors in Section 3.4.

Method	***Backbone***	***Framework***	***Anchor***	***Highlight***
RetinaNet [157], 2017	ResNet	FPN	√	Proposed the Focal Loss to re-weight samples.
Faster R-CNN+ATT [252], 2018	VGG-16	Faster R-CNN	√	Designed three attention modules and applied to the Faster R-CNN to test the performance, three modules are channel-wise self-attention net, visible bbox attention net, and part-region attention net.
GDFL [155], 2018	VGG-16	VGGNet	×	Proposed a scale-aware attention module.
SSA-CNN [88], 2019	VGG-16	Faster R-CNN	√	Used the semantic feature in a self-attention mechanism.
MGAN [186], 2019	VGG-16	Faster R-CNN	√	Predicted the possibility map of the visible part of the pedestrian and fused with the feature in a guided attention way.

ples take a considerable proportion of the loss. We list the representative detectors that we will introduce in Table 3.4. There are two methods to deal with hard negatives in the training phase that will be introduced in Section 3.4.1: hard sampling and soft sampling. To reduce the hard negatives, utilizing the semantic information to aid the model and adopting the attention mechanism to force the model to focus on some specific region is an option, which will be analyzed in Section 3.4.2.

3.4.1 Dealing with Hard Negatives by Sampling

The sampling methods can be divided into hard sampling methods and soft sampling methods, where the hard sampling methods aim to select a subset of hard examples to train the model, such as hard negative mining utilized in [115] and OHEM [205]. Soft sampling aims to assign new weights to hard examples and easy examples, such as the representative Focal Loss proposed in [157]. The intuition behind the Focal Loss is to improve the accuracy of OSDs. The essential difference between TSDs and OSDs is that TSDs have a proposal generation phase before generating the final bbox, while OSDs do not. Correspondingly, TSDs are usually slower than OSDs but with higher precision. Lin et al. [157] deduced that one reason behind OSDs' unsatisfactory precision is that OSD needs to classify approximately 100k candidate bboxes, while TSD only needs to process around 2000 candidate bboxes with the first stage's help. Consequently, most of the candidates that OSD needs to process are hard negative samples, which can overwhelm the entire training process. In essence, this is the imbalance between foreground and background classes.

$$FocalLoss = -\alpha \begin{cases} (1-p)^{\gamma} \log(p) & \text{if } y = 1 \ , \\ p^{\gamma} log(1-p) & \text{otherwise} \ . \end{cases} \tag{3.6}$$

To cope with class imbalance, Lin et al. proposed the Focal Loss [157] that defined in Eq. (3.6) to handle this problem. They also proposed a network called RetinaNet, which used the Focal Loss as the loss function. RetinaNet took the ResNet [124] and feature pyramid network (FPN) [156] as the backbone to extract features. In the experiment, the focal weight α in Eq. (3.6) was set to 0.25, and γ was set to 2. RetinaNet beats the accuracy of many TSDs, including Faster R-CNN, on almost all subsets of MS COCO dataset [158].

3.4.2 Adopting Attention Mechanisms

The concept of 'attention mechanism' has been proposed for image caption and natural language processing (NLP) tasks for more than 10 years. The authors of [195] regarded

the RPN as a practical use of attention mechanism that could contribute to the detection network. Generally speaking, when people look at a picture, there are certain areas that will hold their gaze first. Similarly, a model can be tuned to focus on a specific area by using a greater weight than other places.

The Attention Domain

In the computer vision image-based detection area, attention mechanisms can be classified into spatial domain attention, channel domain attention, and mixed domain attention from the aspect of 'attention domain'. To give an example, we take the input feature map of size $H \times W \times C$ for demonstration. Jaderberg et al. proposed Spatial Transformer Network (STN) [138] in 2015, which is an example of the spatial domain attention methods. The Spatial Transformer module [138] was designed to generate a mask of size $H \times W$, which contains the spatial weights of the input feature map. For the spatial domain attention mechanism, each channel of the input feature map is processed the same way by multiplying the same $H \times W$ attention mask to obtain the output feature maps. SENet [130] is an example of the channel domain attention methods. The input feature map of the SE module [130] is squeezed into the size $1 \times 1 \times C$ first by global average pooling, then the $1 \times 1 \times C$ channel-wise attention mask is trained by a series of layers, including the FC layer, ReLU function, FC layer, and Sigmoid function. The SENet [130] treats every spatial feature on the same feature map channel equally, and the output feature maps are generated by multiplying the input feature map F_i on the i-th channel by the attention scalar S_i of the corresponding channel. The mixed domain attention methods (e.g., CBAM [234]) produced the attention mask of size $H \times W \times C$, enabling each spatial feature on each channel to have different attention weights. The output feature maps are obtained by point-to-point multiplication of the input feature maps and the corresponding attention weights.

Some previous works have adopted the attention module to improve the detection accuracy. To investigate how attention mechanisms could benefit CNN models to focus on the key features, Zhang et al. did an ablation study of three attention modules and applied to the baseline detector Faster R-CNN to test the performance in Faster R-CNN+ATT [252]; these three modules are channel-wise self-attention net, visible bbox net, and part regions net, respectively.

Self-attention Mechanisms vs. Guided-attention Mechanisms

From the perspective of the source-target relationship, the computer vision-based attention mechanism can be roughly divided in two ways: the self-attention mechanism and guided-attention mechanism. The source and target of the self-attention mechanism are the same; the guided-attention mechanism exploits external features to guide the model.

As a representative example of self-attention methods, SSA-CNN [88] was proposed in 2019. Similar to the aforementioned SDS-RCNN [81], SSA-CNN takes advantage of the semantic features. In contrast, SSA-CNN uses the result of the semantic segmentation directly in the self-attention mechanism and forces the detector to focus on the illuminated 'attention' regions, while SDS-RCNN does not infuse the semantic feature into detection task. SSA-CNN takes the two-stage Faster R-CNN [195] detection framework. Instead of the re-weighting approach to 'attention' in SENet [130], the authors of SSA-CNN [88] extracted the semantic feature maps from different layers and concatenated these multi-scale semantic feature maps to the original feature maps in channel-wise to boost the detection accuracy. The extracted semantic feature maps are concatenated to their backbone's feature map at the corresponding conv4_3 and conv5_3 layer of the RPN. In the second stage, the extracted semantic feature map from conv4_3 layer is pooled and concatenated to conv5_3 layer semantic feature map first. Then, the combined semantic feature maps are served as the features in the backbone network for pedestrian classification.

Detectors that use the self-attention mechanism (e.g., SSA-CNN [88]) exploit the features generated in previous steps and feed back to the model after transforming. In contrast, guided-attention mechanisms exploit external features.

We introduced SAF R-CNN [154] in Section 3.2, which adopted two parallel subnetworks to detect large-scale pedestrians and small-scale pedestrians, respectively. Graininess-aware deep feature learning (GDFL) [155] used a similar idea to improve the multi-scale pedestrian detection by proposing the scale-aware guided-attention module: one guided mask for the small-scale pedestrian and the other for the large-scale pedestrian. MGAN [186] is a TSD that took Faster R-CNN's framework and adopted the guided-attention manner. The authors innovated a mask-guided attention module (MGA), which was used between the RoI Align of the first stage network–RPN and the first FC layer of the second stage network. The MGA aims to predict an attention map of the parts of pedestrians that will likely be visible by taking advantage of the external information–the visible part annotations provided by datasets, and the pipeline of MGA is shown in Fig. 3.1. The F_r is the input feature with dimension $H \times W \times C$ and F_{pm} is a generated probability map with the number of channel = 1. The output feature is given by,

$$F_m = F_{ri} \circ F_{pm}, \ \ for\ i = 0, 1, 2, ..C, \tag{3.7}$$

where $\circ$ denotes the element-wise dot product and i is the channel index. The loss function of this module is named L_{mask}, which is a per-pixel BCE loss for weak supervision tasks. The output of MGA F_m is a feature map, emphasizing the spatial features at the location where MGA thinks it could the visible part of the pedestrian. Generally speaking, the F_m is guided by the attention mask F_{pm} through the attention mechanism.

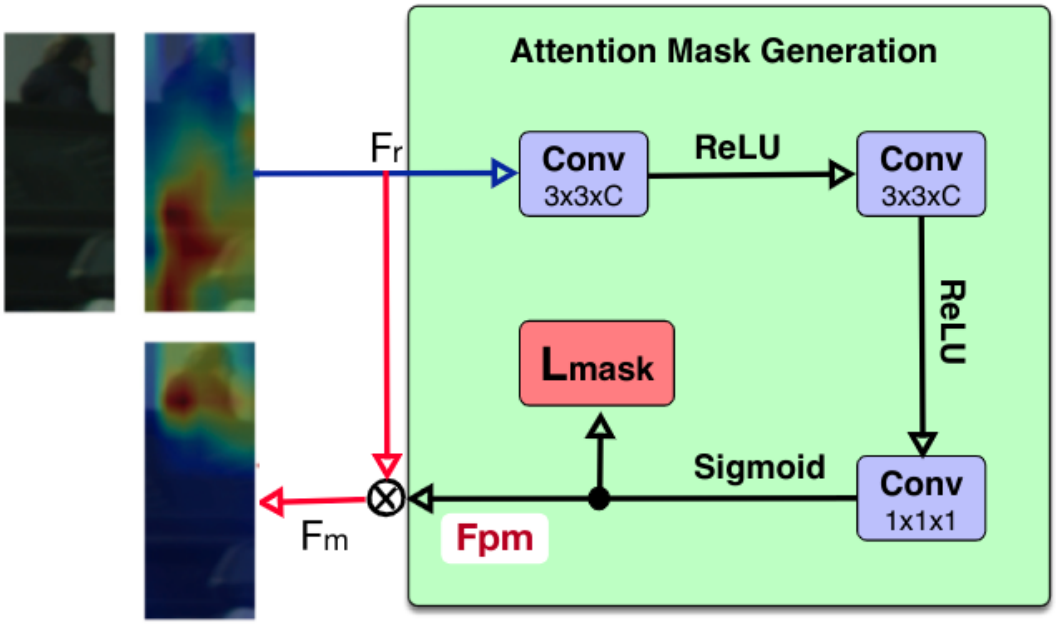


Figure 3.1: An illustration of the working pipeline of MGA module in [186].

3.4.3 Comparisons and Conclusions

The sampling is a classic manner to deal with hard samples, as we introduced in Section 3.4.1. The attention methods leverage external or contextual information to aid the detector with alleviating hard samples. For the application of pedestrian detection that applied to autonomous vehicles, the urban scene is complex. The persons reflected in shop windows and the models printed in the billboards are typical examples of hard negatives on the roadside. They are negative samples, but they look very similar to the positive samples and are therefore difficult to distinguish. The detectors may not have the knowledge to figure out they are not real pedestrians. One possible method for resolving this is to set the 'ignore' region and force the model to treat the features of 'ignore' regions as negative samples. Obviously, this method needs to be well-balanced during the training phase. For the person reflected in the window, it can be processed in the post-processing step, such as to find if the features inside two bboxes are horizontally symmetric. For the objects that

look similar to the real pedestrian, fusing the semantic feature into the detection results by an attention manner can be an asset. It is also possible to leverage the surrounding environment's contextual features to benefit hard samples' classification. For example, the detector could have the knowledge that the person's feature inside the billboard may be a negative sample.

Chapter 4

Proposed Methods

To detect pedestrians on a given image, the detector needs to pre-process the image firstly. The processed image is then transformed into a tensor before being fed into the Conv layers. The outputs of the detector contain the bboxes that enclose pedestrians and corresponding confidence scores, and we need to filter out the true positive predictions and reduce false positive samples in the post-process phase. In this chapter, we introduce our proposed BCNet [203] and its variants in detail from pre-processing, network structure and design, and post-processing three phases.

4.1 Pre-processing

Image-based pedestrian dataset provide the RGB images and the corresponding annotations of ground truth pedestrians. In the pre-process phase, we need to pre-process both the images and annotations.

4.1.1 Annotations

Taking the CityPersons dataset [249] as an example, the images are in the resolution of 1024×2048. The authors divided pedestrian instances into a negative class (ignore) and 5 positive classes: pedestrians, riders, sitting persons, other persons with unusual postures, and group of people. The annotation of each object instance are in the order of class label, horizontal ordinate of the bbox's left-top point, corresponding vertical ordinate, width, and height of the bbox. The bbox is the way to indicate the location of the pedestrian's full body, whether the pedestrian is occluded or fully visible. If the pedestrian is occluded, the annotated bbox is assigned by calculation and prediction. In addition to the bbox of

the pedestrian's full body, the CityPersons dataset also provides the bbox of visible part for each pedestrian.

Our proposed BCNet [203] is an anchor-free detector, and we take advantage of the center keypoint of object instance. Our detector is designed to predict the center keypoints of the full body and the visible part of each pedestrian, while predicting the height and offset. The annotations work as objectives, which can aid the detector during the training phase. Therefore, we need to transform the annotations to make it consistent with our detector's expected output in order to train our detector.

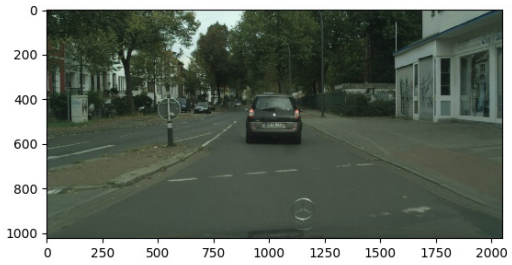
For each pedestrian's full body and visible part center keypoints, we produce a three-channel mask as the ground truth training data. We apply the 2D Gaussian function for each non-ignored ground truth pedestrian sample's bbox area, and make it the first channel, and we choose the higher value at each location if there is any union between different pedestrian instances. The second channel is made of values of 1 at the bbox area, and we also process all the ignored areas in the second channel and set that area to all 0. The third channel is made of values of 1 at center keypoints of the bbox, and the center keypoints are the central point of the bboxes.

The ground truth height mask is of two channels. We take the log value of the pedestrian's original height, and assign it to the 4×4 area of the center keypoint, which constructs the first channel. In the second channel, we set the central 4×4 area to value of 1. The ground truth offset mask has three channels. When calculating the center keypoints, we get the smallest integer value bigger than or equal to value of the coordination. We use the difference between the float value and the integer value as the ground truth offset. The first channel is the vertical offset, the second channel is the horizontal offset, and the third channel is set to 1 as before. These values are all assigned to the center keypoint's location in each channel.

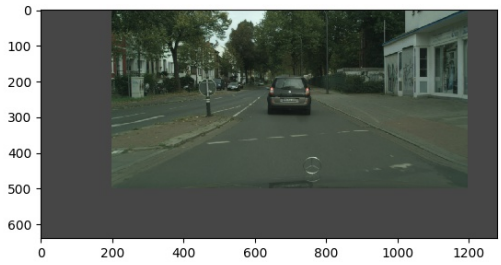
4.1.2 Images

The image pre-processing step includes image normalization, parameters modification, and data augmentation in common. When normalizing the RGB images, the mean values of each channel is 0.485, 0.456, and 0.406, and the standard deviation value is 0.229, 0.224, and 0.225, respectively. This is a common practice to normalize the input images, the values are calculated based on the ImageNet dataset [100] and suggested by PyTorch[1]. We jitter brightness by 0.5, and contrast, saturation, and hue of input images remain unchanged.

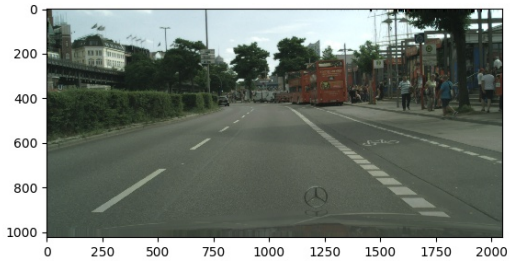
[1]https://pytorch.org/docs/stable/torchvision/models.html



(a) The original image A.



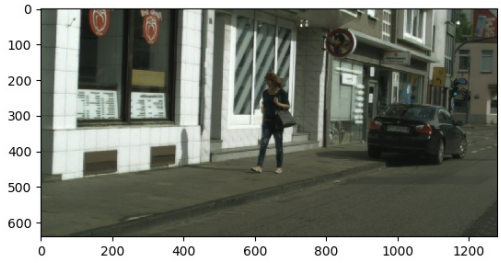
(b) The image A after pre-processing.



(c) The original image B.

(d) The image B after pre-processing.

(e) The original image C.



(f) The image C after pre-processing.

Figure 4.1: Images visualization.

Data augmentation is a popular manner to reduce overfitting by enlarging the diversity of data. As we listed in Table 2.3, data augmentation techniques include geometric transformation, flipping, and cropping. In our pre-process step, we first resize the input image with a random ratio varying from 0.4 to 1.5. For each resized image, we have a possibility of 0.5 to flip it horizontally, and a possibility of 0.5 to keep the image unchanged. The corresponding annotations are modified to be consistent with the images.

Most CNN-based models have a strict limitation on the size of input images and they require a uniform size of input images. Therefore, before we feed the images into our CNN-based model, we need to ensure the images are in the same size. We pave the smaller images to a random position and fill the rest positions with values of 1. We crop the image from a random position if the image is larger than our unified image size. Cropping and paving make the images into a unified size during training our model.

We visualize some training images with both original images we obtain from the CityPersons dataset [249] and the images after pre-processing in Figure 4.1 to better understand what we did in the data augmentation step. Sub-figure 4.1(b) was pre-processed by resizing to a smaller ratio and image pave, sub-figure 4.1(d) was enlarged and cropped into the unified image resolution, and sub-figure 4.1(f) was horizontally flipped and enlarged before image cropping.

4.2 Proposed Networks

In this section, we will introduce details of our BCNet [203], which takes advantage of the semantic information of each pedestrian. We exploit the fusion of visible part semantic feature and full body semantic feature. We further propose different types of methods to fuse semantic features.

4.2.1 Model Overview

We propose a detector called BCNet in this paper, and our model is a one-stage anchor-free detector that adopts the CSP [162] as the baseline. CSP is an anchor-free detector which achieves promising accuracy. We note that a drawback of the CSP is that it does not fully use he visible part semantic features for each pedestrian. To cope with this issue, we introduce an extra parallel branch in our detector to predict the center keypoint of the corresponding pedestrian's visible part.

As demonstrated in Figure 4.2, we design our model to predict the center keypoint of the visible part (CKVP) in addition to the prediction of the center keypoint of the full

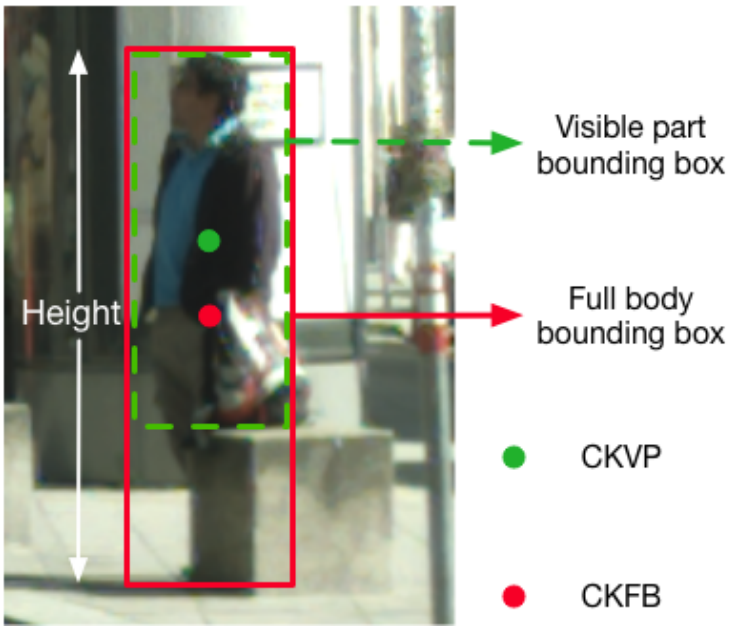


Figure 4.2: An illustration of CKVP and CKFB. The cropped image is from the CityPersons dataset [249].

body (CKFB) for each pedestrian. Our detector predicts the height of each pedestrian, and the offsets of the CKFB. These predictions are combined to convert into a bbox for each pedestrian, and the ratio of the pedestrian bbox is uniformly fixed at 0.41.

Table 4.1: The structure of ResNet-50 [124].

	conv2_x layer	***conv3_x layer***	***conv4_x layer***	***conv5_x layer***
Block Structure	$1 \times 1 \times 64$ $3 \times 3 \times 64$ $1 \times 1 \times 256$	$1 \times 1 \times 128$ $3 \times 3 \times 128$ $1 \times 1 \times 512$	$1 \times 1 \times 256$ $3 \times 3 \times 256$ $1 \times 1 \times 1024$	$1 \times 1 \times 512$ $3 \times 3 \times 512$ $1 \times 1 \times 2048$
Repeat	3	4	6	3

[1] The ResNet-50 begins with a $3 \times 3 \times 64$ Conv layer, a batch normalization layer, and a ReLU function before conv2_x layer.

[2] The block structure of the corresponding layer is repeated by multiple times.

The pipeline of our model is demonstrated in Fig. 4.3. We adopt the ResNet-50 [124] as the backbone in ConvNet to extract feature maps from four different layers (i.e., conv2_x layer, conv3_x layer, conv4_x layer, and conv5_x layer in ResNet-50 as listed in Table 4.1).

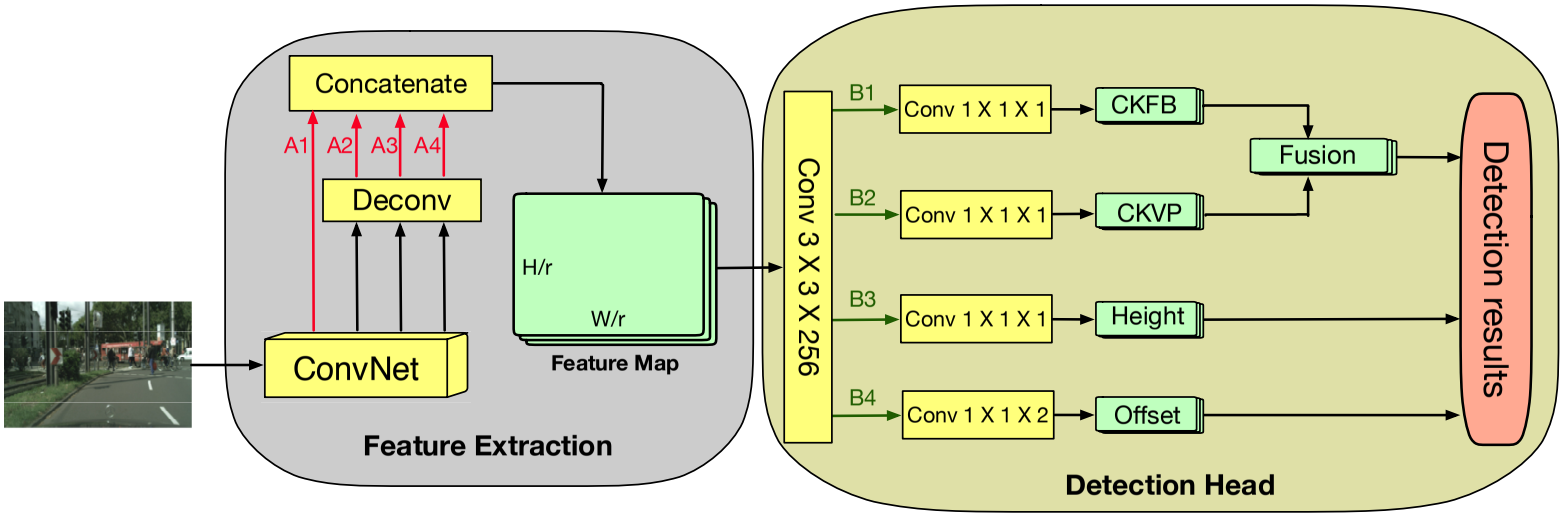


Figure 4.3: The architecture of BCNet. We take ResNet-50 [124] as the backbone in ConvNet. A1-A4 and B1-B4 are used to denote the locations.

We further apply the transposed convolution layers in order to rescale these four feature maps to the same dimension of $H/r \times W/r \times C$, where H is the height of the input image, W is the width of the input image, r is the pre-defined downsampling ratio, and $C = 256$ is the number of feature maps' channel.

After applying the L2 normalization for each feature map, these four feature maps are concatenated in channel-wise. Concatenating feature maps that extracted from different levels is widely used to boost the model's performance because the feature map from a deeper layer has a lower resolution but a higher semantic level. The feature maps with low resolution can result in a loss in location accuracy as small offsets on low-resolution feature maps can cause large differences in location results; the feature maps with high-level semantics could enable the detector to find targets and make more accurate classification results. The final feature map that generated in the feature extraction phase is of size $H/r \times W/r \times 4C$, and we use r = 4 as this is the downsampling ratio suggested in [209].

In our BCNet, after reducing the channel of the feature map from 1024 to 256 by applying a 3×3 Conv layer, four parallel branches are appended in the detection head. The CKFB branch is used to predict a heatmap which indicates all the center keypoints of the full body for every possible pedestrians on the entire input image; the CKVP branch works in a similar way as the CKFB branch, predicting a heatmap that indicates the center keypoints of the visible part for every pedestrian; the height branch is used to predict the corresponding heights of pedestrians in the image; the offset branch is processed by two 1×1 Conv kernels to generate the corresponding offsets to fine tune the CKFB's locations in width and height, respectively. The parameters of these four parallel branches are not shared.

As we mentioned, the outputs of the CKFB and CKVP branches are heatmaps, and the predicted pedestrians can be located through the high response points on the heatmap. We extract the detection results from the above four branches to generate the final detection bbox results: we filter out the response points that score above the pre-defined threshold on predicted heatmaps of the CKFB and CKVP branch, respectively. The high response points indicate the high degree of confidence in detecting the center keypoints of the full body and visible part for each pedestrian. We further fuse these two types of center keypoints by using the methods that we will introduce in Section. 4.2.3. The predicted offsets are used to fine-tune the coordination of the final center keypoints both vertically and horizontally. The bbox aspect ratio in the pedestrian detection area is set at 0.41 as defined in [102], which can be multiplied with the predicted height in order to obtain the bbox width.

4.2.2 Pedestrian Detection with the Semantic of the Visible Part

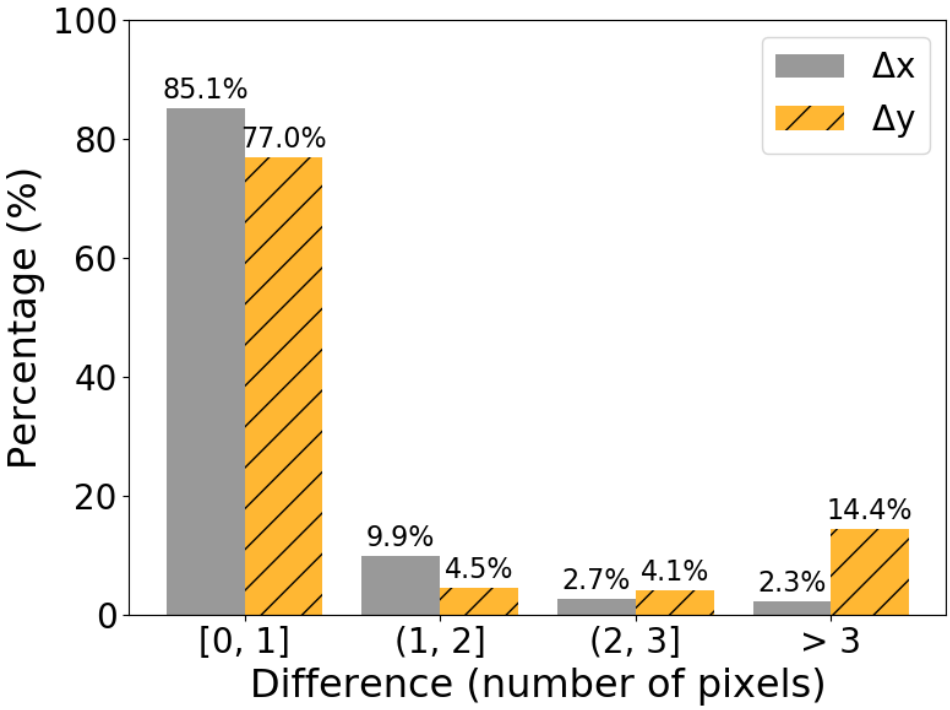


Figure 4.4: The distribution of distances between the CKVP and CKFB for each pedestrian in the CityPersons training set in the resolution of 256×512. Δx and Δy represent the horizontal and vertical distance, respectively.

To prove the practicality of our proposed semantic fusion method, we analyze the distribution of distances between the CKFB and CKVP for each annotated pedestrian sample in the training set of the Citypersons dataset [249], and we demonstrate the distribution in Fig. 4.4.

In the training set of the CityPersons dataset, there are 19654 annotated pedestrians in total. The resolution of the heatmaps generated in our BCNet's detection head are downsized with the ratio r=4, compared to the original resolution of input images. Accordingly, we shrink images and change the coordination of bboxes to the same resolution (i.e., 256×512). As demonstrated in Fig. 4.4, around 85% and 77% of the pedestrians have a distance of no larger than one pixel between their CKVP and CKFB horizontally and vertically, respectively.

For the pedestrians under reasonable occlusion (i.e., occlusion ratio ≤ 0.35), the CKVP and CKFB should be close to each other or even identical. For the heavily occluded pedestrians (i.e., occlusion ratio ≥ 0.35), two types of center keypoints are usually far

apart spatially, which prevents the location sensitive confidence scores of both CKFB and CKVP from being dramatically affected.

Based on our above analysis, the relationships and distances between the CKFB and CKVP enable the CKVP to aid the CKFB in locating the pedestrian on the fused heatmap; the final confidence score could be enhanced when the pedestrian is under reasonable occlusion because the CKVP is close to CKFB.

4.2.3 Fusing the Full Body Semantic and Visible Part Semantic

Fusing by Two Hyper-parameters

One natural way to combine the location sensitive confidence scores obtained from the full body center keypoint heatmap and the visible part center keypoint heatmap could be defined as in:

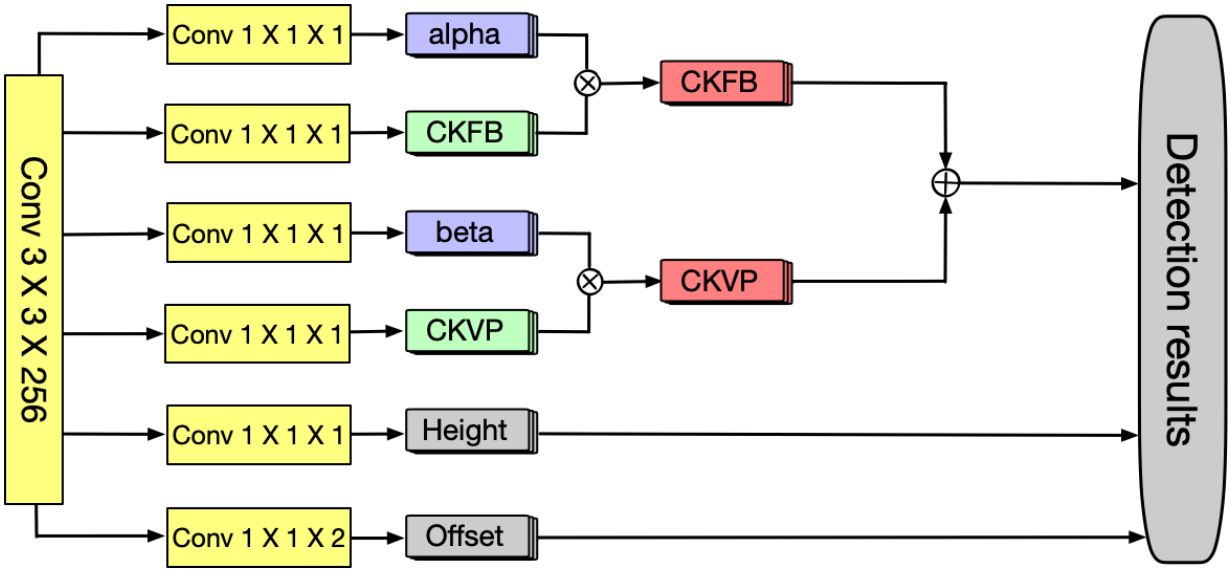
$$\alpha Heatmap_F + \beta Heatmap_V = Heatmap, \tag{4.1}$$

where $Heatmap_F$ and $Heatmap_V$ represent the heatmaps that contain confidence scores of the CKFB and CKVP, respectively. α and β are weighting factors $\in [0, 1]$. The final confidence scores on $Heatmap$ in Eq. (4.1) can be generated by different ratios of confidence scores on the CKFB heatmap and CKVP heatmap by tuning α and β. Hyper-parameters α and β are applied to the entire heatmap and the confidence scores at different locations are treated equally. As we introduced previously, the location sensitive confidence scores at different locations will not affect each other during the heatmap fusion. We will introduce more details about tuning hyper-parameters α and β in Section 5.3.1.

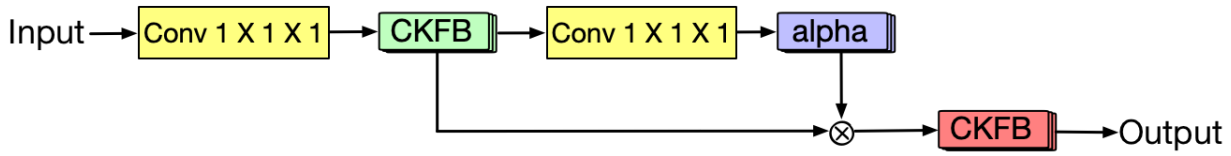
Fusing by Adopting Attention Modules

Adopting attention mechanisms could enable the detectors to focus on specific areas of interest. We adopt a naive attention module, a channel-domain attention module, and a spatial-domain attention module, respectively.

A naive attention module In addition to fusing the heatmaps by applying two pre-defined hyper-parameters α and β to the entire heatmap and treating every spatial feature equally, we can train the model flexibly to learn the parameters at corresponding locations. We design a naive attention module, which is applied to the last feature map that is used to predict the center keypoint heatmaps. This naive attention module is a special case of the mixed-domain attention mechanism because it is applied to a feature map with only 1 channel. The naive attention module is two $1 \times 1 \times 1$ Conv layers that are applied to

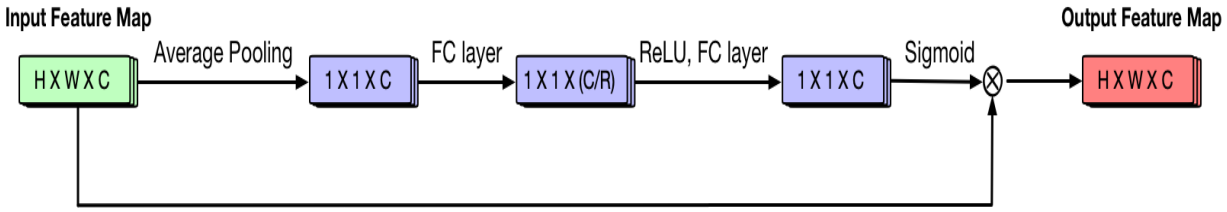


(a) The naive attention module that is only used in the detection head of our BCNet,which is named BCNet-Att-A1.

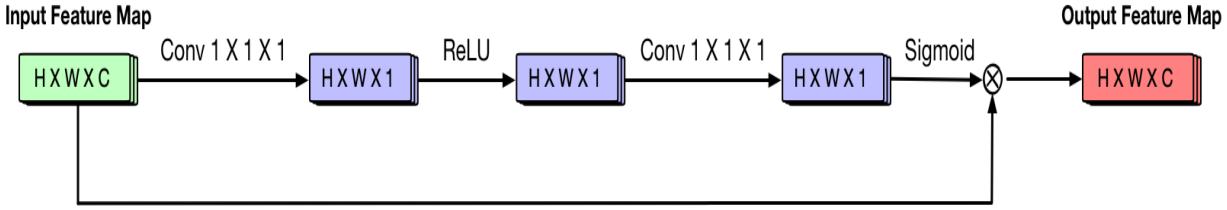


(b) We take the CKFB branch to illustrate the other variant, which is named BCNet-Att-A2.

Figure 4.5: The naive attention module. The green CKFB and CKVP are draft heatmaps, the red CKFB and CKVP are the re-weighted heatmaps that will be added up to produce the final heatmap.



(a) The channel-domain attention module we adopt is the SE layer that was proposed in [130].



(b) The spatial-domain attention module.

Figure 4.6: The dimensions of corresponding feature maps are denoted in height × width × channel format. The input feature maps' channel is 256 in the SE layer of our network. The output feature map is re-weighted by applying the attention scalar to the input feature map.

the feature extracted by the model, and they produce attention masks *alpha* and *beta*. *alpha* and *beta* are further fused with the CKFB heatmap and CKVP heatmap by dot product, respectively to generate the final heatmaps. The final heatmaps are re-weighted by the correspondingly trained attention masks *alpha* and *beta*. The activation function Sigmoid is applied to every heatmap in our model. This variant of the model illustrated in Fig. 4.5(a) is named BCNet-Att-A1. We also propose another variant BCNet-Att-A2 as illustrated in Fig. 4.5(b), which applies the attention module on the corresponding draft heatmaps, instead of the feature map produced in the feature extraction stage. The same operations will be applied to the CKVP branch in BCNet-Att-A2. In the naive attention modules that we propose, the final center keypoint heatmap is generated by adding the final CKFB and CKVP.

A channel-domain attention module We adopt the channel-domain attention mechanism that was proposed in SENet [130]. The working pipeline of the SE layer is illustrated in Fig. 4.6(a). We designed five variants of our BCNet that adopt SE layers [130] in different layers to investigate the effect of SE layers. We list the locations where we apply attention modules and the aliases of these variants in Table 4.2, and the locations are depicted in Fig. 4.4 in red and green color. The input feature map is first squeezed into the size $1 \times 1 \times C$ by applying a global average pooling, then the $1 \times 1 \times C$ channel-wise attention mask is obtained after applying a series of layers as shown in Fig. 4.6(a).

Table 4.2: Variants and aliases of BCNet that adopt attention methods in different locations.

Location	***Channel-domain attention***	***Spatial-domain attention***
B1,B2	BCNet-Att-B1	BCNet-Att-C1
B1-B4	BCNet-Att-B2	BCNet-Att-C2
A1-A4	BCNet-Att-B3	BCNet-Att-C3
A1-A4,B1,B2	BCNet-Att-B4	BCNet-Att-C4
A1-A4,B1-B4	BCNet-Att-B5	BCNet-Att-C5

SENet [130] treats each spatial feature on the same feature map channel equally by using a channel-sensitive weight; SE layer generates the output feature maps by multiplying the input feature map F_i on the i-th channel by the corresponding channel's attention scalar S_i.

A spatial-domain attention module In addition to the channel-domain attention mechanism, we adopt the spatial-domain attention mechanism in order to find the most effective method to fuse the full body semantic and the visible part semantic. We design a simple spatial-domain attention module as illustrated in Fig. 4.6(b), which contains a Conv layer, ReLU function, Conv layer, and Sigmoid function. The first Conv layer has only one 1×1 kernel, which could squeeze the input feature channel to 1. The second Conv layer has one 1×1 kernel and it works to produce a spatial-domain attention mask, which treats each channel of the input feature maps with the same attention scalar mask by dot product. The details of variants are listed in Table 4.2 in the same manner as the channel-domain attention module. The alias ending with 'B' indicates that our BCNet adopts the channel-domain attention method–SE layer in the corresponding locations; otherwise, the alias ending with 'C' indicates our BCNet adopts the spatial-domain attention method.

4.2.4 Loss Function

The loss function works to evaluate how far the prediction is away from the target, and provides the model with the direction. In our BCNet, the CKFB branch and CKVP branch are formulated as classification tasks, and we use L_{cls_f} and L_{cls_v} to denote the loss of the CKFB branch and CKVP branch, respectively. We apply the 2D Gaussian function in [162] to generate the ground truth heatmaps for CKFB and CKVP.

In the CKVP branch, y_{ij} denotes the $Score_{CKVP}$ on the ground truth CKVP heatmap,

$$L_{cls_v} = -\frac{1}{N}\sum_{i=1}^{H_r}\sum_{j=1}^{W_r}\begin{cases}(1-p_{ij})^{\gamma}\log(p_{ij}) & \text{if } y_{ij}=1\ ,\\ (1-y_{ij})^{\delta}(p_{ij})^{\gamma}log(1-p_{ij}) & \text{otherwise}\ .\end{cases} \tag{4.2}$$

and p_{ij} denotes the $\widehat{Score_{CKVP}}$ on the predicted CKVP heatmap. Both y_{ij} and p_{ij} are in the range [0, 1]. p_{ij} is the probability that our model predicts the likelihood of CKVP at location (i, j). Similar to [150] and [162], we adopt the Focal Loss function proposed in [157] and make some modifications. The Focal Loss variant we use is defined in Eq. (4.2), where N is the number of annotated pedestrians in the image. H_r and W_r represent the height and width of the images that are downsampled by the factor r, respectively. As suggested by Law et al. in [150], we set the focusing hyper-parameters $\gamma = 2$ and $\delta = 4$ to perform our experiments. In the CKFB branch, the same loss function in Eq. (4.2) is applied to calculate L_{cls_f}.

The height and offset branches are handled as regression tasks, and we apply the smooth L1 Loss function [115]. For the t-th pedestrian on the image, we use $log(h)$ to re-annotate the ground truth height h_t and assign h_t to locations within an area of radius 2. For the pedestrian with a height h and width w after downsampling by the factor r, we use the value $h/2-\lfloor h/2\rfloor$ and $w/2-\lfloor w/2\rfloor$ to represent the vertically offset and horizontally offset, respectively. The loss functions in the height branch and offset branch is given by,

$$L_{height} = \frac{1}{N}\sum_{t=1}^{N} SmoothL1Loss(h_t, \hat{h_t}), \tag{4.3}$$

$$L_{offset} = \frac{1}{N}\sum_{t=1}^{N} SmoothL1Loss(o_t, \hat{o_t}), \tag{4.4}$$

where h_t, $\hat{h_t}$, o_t, and $\hat{o_t}$ are the ground truth height, predicted height, ground truth offset, and predicted offset of pedestrian t, respectively.

The final objective function to be optimized during the training phase is defined as in:

$$Loss = \lambda_f L_{cls_f} + \lambda_v L_{cls_v} + \lambda_s L_{scale} + \lambda_o L_{offset}, \tag{4.5}$$

where λ_f, λ_v, λ_s, and λ_o are weighting factors for the corresponding loss, and we experimentally set them to 0.01, 0.01, 1, and 0.1, respectively.

4.3 Post-processing

Post-processing is the last processing step to polish our prediction results. We will introduce the NMS working scheme in this section. NMS is used to eliminate the duplicate and

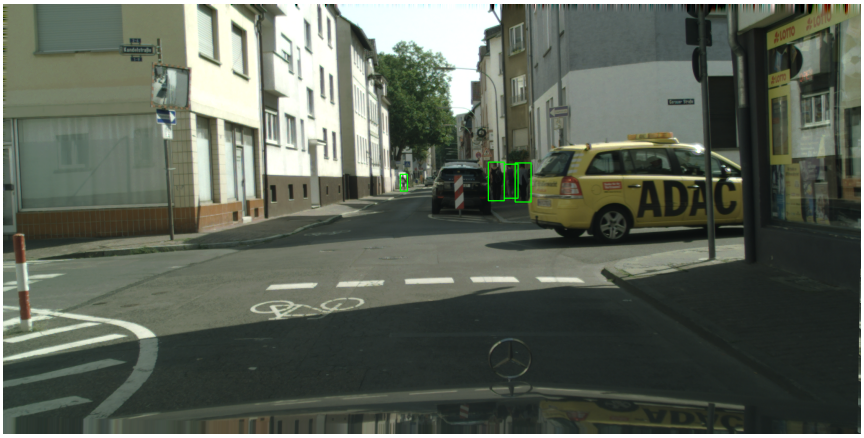
unnecessary prediction results.

4.3.1 NMS

NMS will be applied when testing the detector, but not during the training step. On the final fused heatmap, we keep the locations where the confidence scores are higher than 0.3 as the center keypoints to locate candidate pedestrians. The candidate bboxes are further generated by integrating with the corresponding heights and offsets.

The greedy based-NMS operation is applied to the candidate bboxes at a threshold of 0.5 to select the highest score among the candidates with the IoU overlap larger than the threshold. We visualize the comparison of before applying NMS and after applying NMS to detector's predictions in Figure 4.7.

(a) Before applying NMS.



(b) After applying NMS.

Figure 4.7: The comparison of before applying NMS and after applying NMS.

Chapter 5

Experiments

In this chapter we display our experimental settings of our proposed methods and comparison results with other existing methods. We will work on balancing the hyper-parameters α and β we introduced in Eq. (4.1), and we will do an ablation study to investigate how different attention mechanisms could affect the accuracy of detectors. We will then compare the experimental results on the CityPersons dataset [249] and the ETH dataset [110].

5.1 Datasets and Evaluation Metrics

We evaluate our BCNet on two datasets: the CityPersons dataset [249] and the ETH dataset [110]. The CityPersons dataset offers the visible part annotations of each pedestrian sample in addition to the full body annotation, which meets our BCNet's mandatory requirement. In addition, the images of the CityPersons dataset have a resolution of 1024 $\times$2048, and the images are collected in 3 different seasons and 27 cities. There are 2975 images in the training set and 500 images for evaluation in the validation set. Compared to the Citypersons dataset, the images of the ETH dataset have a resolution of 640 $\times$480, which is much lower. The CityPersons dataset and the ETH dataset have a large number of pedestrians per image, which is 7.0 and 6.1, respectively. Based on their attributes, we consider them to be representative, so we perform experiments on the CityPersons dataset and the ETH dataset.

The unified evaluation metric in the pedestrian detection area is MR^{-2}, which was introduced in [102]. MR^{-2} is the mean value of nine derived miss rates with the corresponding FPPIs evenly located in $[10^{-2}, 10^{0}]$ within the log-space, and a lower MR^{-2} indicates a higher accuracy. The corresponding miss rate and FPPI are calculated by

changing the threshold of bbox confidence score. When the threshold is lower, more detection results are taken, which could cause a relative higher false positives and lower miss rate. In our experiment we use MR^{-2} to evaluate detectors' performance in Fig. 5.1, Table 5.4, Table 5.5, Fig. 5.3, and Fig. 5.4.

Table 5.1: Evaluation Setups of the CityPersons dataset [249]

	Reasonable	***Bare***	***Partial***	***Heavy***	***Small***	***Medium***	***Large***
Height (in pixels)	[50, $+\infty$]	[50, $+\infty$]	[50, $+\infty$]	[50, $+\infty$]	[50, 75]	[75, 100]	[100, $+\infty$]
Visibility ratio	[0.65, 1]	[0.9, 1]	[0.65, 0.9]	[0, 0.65]	[0.65, 1]	[0.65, 1]	[0.65, 1]

5.2 Experimental Settings

In this section, we will provide detailed environmental settings of our proposed methods and our implementation details of some existing methods we compare with.

5.2.1 Baseline Model with Attention Mechanisms

The proposed BCNet variants and baseline model variants are all implemented in Python 2.7 and PyTorch 1.2.0. We train our BCNet and the baseline model CSP [162] on a single GPU (Nvidia GeForce GTX 1080 Ti).

We design and adopt different methods to fuse the full body semantic and visible part semantic in Section 4.2.3, thus we could further compare these methods and study the effect of different feature fusion methods. To demonstrate the effectiveness of our BCNet, we apply the same attention modules to the baseline model CSP [162], and the details are listed in Table 5.2. Similarly, the alias ending with 'B' represents the baseline model, adopting the channel-domain attention method–SE layer in the corresponding locations; otherwise, the alias ending with 'C' represents the baseline model that adopts the spatial-domain attention method. The ending numbers of the aliases are named in the same order as in Table 4.2. We train the baseline model CSP [162] on a single GTX 1080Ti GPU, rather than four GTX 1080Ti GPUs as in [162], and we study how much the experimental results will be affected when different numbers of GPUs are used.

Table 5.2: Variants and aliases of baseline model CSP [162] that adopt attention methods in different locations.

Location	*Channel-domain attention*	*Spatial-domain attention*
B1	CSP-Att-B1	CSP-Att-C1
B1,B3,B4	CSP-Att-B2	CSP-Att-C2
A1-A4	CSP-Att-B3	CSP-Att-C3
A1-A4,B1	CSP-Att-B4	CSP-Att-C4
A1-A4,B1,B3,B4	CSP-Att-B5	CSP-Att-C5

5.2.2 Existing Methods

Most of the proposed detectors were trained and tested in different environmental settings, such as GPUs with different computing power and additional amount of GPUs, which could have a huge impact on the detector's accuracy and efficiency. Using more GPUs with powerful computing ability can boost the detector's accuracy to some extend. However, this will introduce a huge cost, and consuming huge computing power may not be compatible with the autonomous vehicles' system requirements.

In our thesis, we will train some existing detectors by using the same environmental hardware on the CityPersons dataset, and we will compare our pedestrian detector BCNet [203] with these existing detectors locally.

We adopt Faster R-CNN [195] with backbone HRNet [211], Faster R-CNN with backbone ResNet-50 [124], RetinaNet [157], GA-RetinaNet [227], ALFNet [161], and Fully Convolutional One-stage (FCOS) [221] in our comparative experiments, and we list the environmental settings in Table 5.3.

- **Faster R-CNN [195]** It is one of the most representative two-stage detectors, and it is still widely used today. The concept of 'anchors' comes from Faster R-CNN. To better take advantage of the framework, we use FPN [156] to enrich the extracted feature. We apply both HRNet [211] and ResNet-50 as backbones to figure out what performance this classic two-stage detector could achieve.

- **RetinaNet [157]** The authors proposed the novel Focal loss to address the class imbalance problem. Correspondingly, RetinaNet beats many two-stage detector's accuracy. RetinaNet is a one-stage detector, it combines ResNet and FPN [156] to better extract features, which introduce a large computing complexity.

- **GA-RetinaNet** [227] Instead of generating anchors by sliding windows, the network is able to learn and predict where the anchor is and what the shape of the anchor box by applying the guided-anchor scheme.
- **ALFNet** [161] One-stage detector with cascaded prediction blocks applying at four different levels, and the detection results are filtered by adopting higher IoU thresholds.
- **FCOS** [221] FCOS is a one-stage detector. It is an anchor-free detector which works to predict the center-ness of the target, integrating with a 4D vector of left, top, right, bottom to generate the bbox prediction.

In the experiment, all the detection models are trained on a single Nvidia GeForce GTX 1080 Ti GPU, with 11GB GPU memory. We do not pre-train these models on any other datasets before training on the CityPersons training set. Training configurations of each detector are listed in Table 5.3. To fully utilize the computing ability of our GPU and obtain the best accuracy, we set the batch size to 2 if the GPU memory is sufficient. We have to set batch size to 1 for the detector Faster R-CNN and FCOS. We adjust the learning rate and the number of epochs by observing the convergence of the training loss. We reduce the training images to a different resolution during training detectors on the CityPersons dataset [249].

5.3 Experimental Results

There are 7 setups in our experiments to evaluate the detection accuracy on the CityPersons dataset; the details are given in Table 5.1. For example, a pedestrian on the Reasonable setup is no less than 50 pixels tall and at least 65% of the pedestrian's body is visible.

We train our BCNet variants and the baseline detector variants on the training set of the CityPersons dataset. The backbone model ResNet-50 was pre-trained on the ImageNet [100] dataset, and we load the pre-trained weights for initialization. We train detectors for 100 epochs, and the mini-batch size is 2. The learning rate for the first 50 epochs is 5×10^{-5}, and the learning rate is decreased to 2×10^{-5} for the remaining 50 epochs. Adam [101] is adopted to optimize the objective function we introduced in Eq. (4.5).

We test our BCNet variants on the validation set of the CityPersons dataset [249], and the mini-batch size is 1 during the inference phase. It takes 0.28s on average to infer a 1024×2048 image on a single GPU, which does not cost extra time compared to the baseline model variants.

5.3.1 The Ablation Study on Two Hyper-parameters α and β

To fuse the semantic of CKVP and CKFB, the first method we introduced in 4.2.3 was to adopt two hyper-parameters α and β in Eq. (4.1). By applying the pre-defined α and β value, the final heatmap can be combined by re-weighted CKFB and CKVP heatmaps. Thereby, the final confidence score on the fused heatmap that is used for producing detection results is combined by different proportions of confidence scores on two heatmaps. To find the best choice of hyper-parameters, we do an experiment on the Reasonable setup of the CityPersons dataset. We demonstrate the effect of varying α and β in Fig. 5.1. It is found that there is a specific area in which the MR^{-2} is promising and stable, and we conclude this happens when the ratio of α and β is around 2:1, and $\alpha \in [0.4, 1]$ and $\beta \in [0.2, 0.6]$. These combinations of α and β can result in a more accurate performance. A similar phenomenon also applies to most of the other setups we list in Table 5.1. To simplify, we take $\alpha = 1$ and $\beta = 0.5$ to conduct the following evaluation and comparisons.

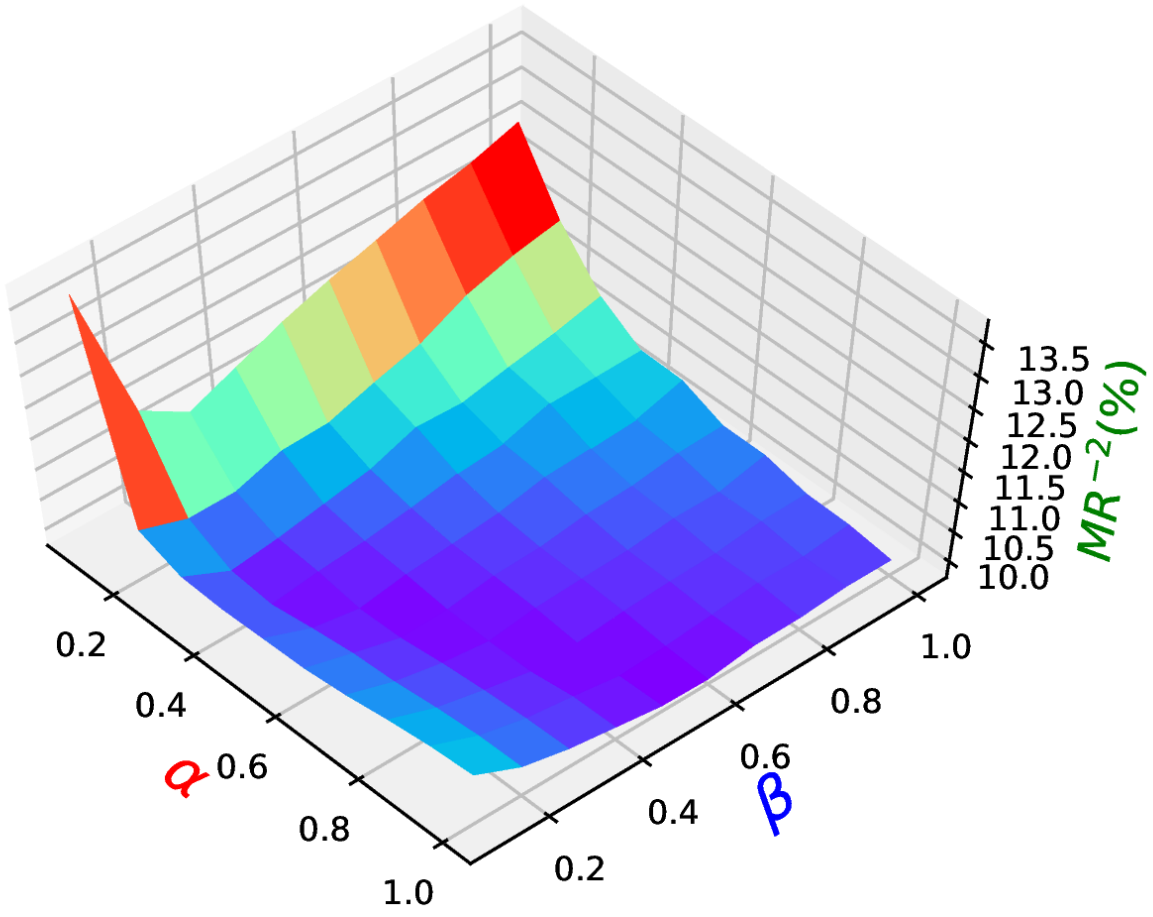


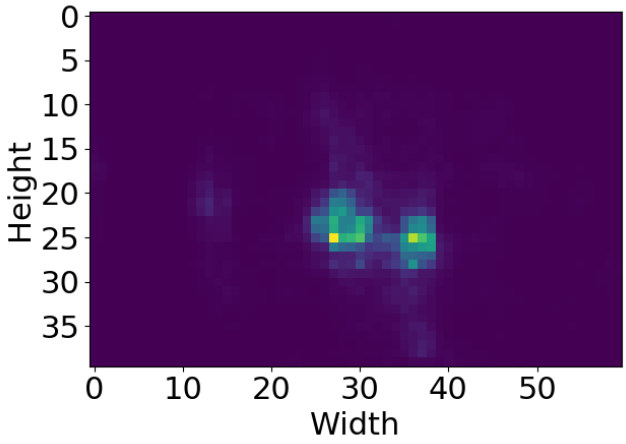
Figure 5.1: Experiments of varying α and β on Reasonable setup of CityPersons dataset, evaluated by MR^{-2}.

We visualize the CKFB heatmap, CKVP heatmap, and the heatmap after fusion (with $\alpha = 1$ and $\beta = 0.5$) in Fig. 5.2. The bright spots on the heatmaps indicate higher confidence scores, which means our detector is more likely to treat them as center keypoints. After the heatmap fusion, the final heatmap in Fig. 5.2(c) is enhanced by the semantic fusion. There is a stronger response on the final heatmap, as the center keypoint spot is brighter than the one before fusion. The visible part semantic remains on the final heatmap but would not become noise, because its response is weaker than the full body semantic with the help of weighting factors balance. In this way, the semantics of the visible part could aid the detection of each corresponding pedestrian's full body. Also, we observe a significant enhancement when the pedestrian is unoccluded or under slight occlusion, because the CKVP is spatially close to CKFB.

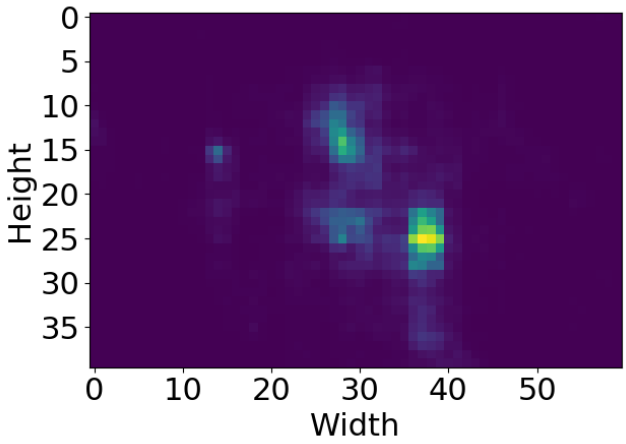
5.3.2 The Ablation Study on Attention Modules

We adopt three types of attention mechanisms that we introduced in Section 4.2.3 to perform an ablation study, and we list the experimental results of our BCNet and baseline model CSP in Table 5.4. Zhang et al. [252] applied the channel-domain attention module to the final feature maps generated in the first stage of the baseline model Faster R-CNN, however, the MR^{-2} dropped from 15.52% to 20.93% on the Reasonable setup of the CityPersons dataset [249]. One drawback is they only applied the attention modules to feature maps generated at the end of the RPN, which is not sufficient to further compare the effects. In our experiment, we apply attention modules to different locations, respectively. When the attention modules are applied to the feature extraction ResNet-50, we apply the attention modules to all four feature maps produced in four different layers, which enables the detector to focus on the area of interest in different levels, compared to Zhang et al.'s work [252].

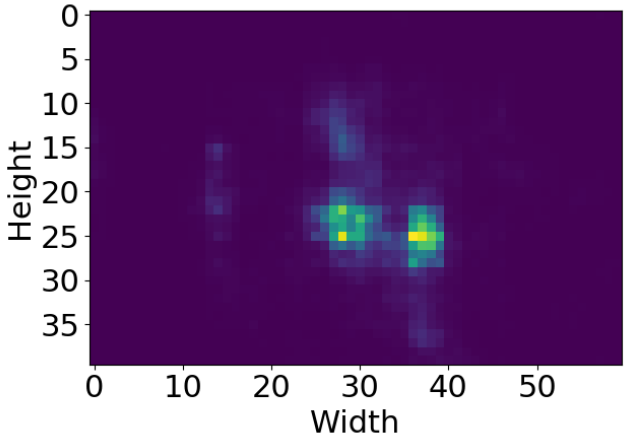
From Table 5.4 we can observe that after applying the channel-domain and spatial-domain attention modules to the baseline model CSP, the corresponding MR^{-2} values on the Reasonable setup vary from 11.72% to 13.20%, and the MR^{-2} of the baseline model CSP that we train on the same environment is 12.14%. Similar performances are also found on the other setups. We observe that CSP-Att-B3 and CSP-Att-C3 can yield modest improvement or comparable results compared to the baseline model. In contrast, the MR^{-2} values of all subsets generated by CSP-Att-B2 and CSP-Att-C2 are worse than the baseline model. The reason for this is that attention modules are only applied to the high semantic level feature maps in the detection head, where detectors may not be capable of extracting the effective area of interest. Even though CSP-Att-B1 and CSP-Att-C1 do not make an improvement on the Reasonable setup, they significantly improve the



(a) The CKFB heatmap.



(b) The CKVP heatmap.



(c) The fused heatmap.

Figure 5.2: The heatmaps are cropped into size 40×60 for visualization. Width and height are in pixels.

performance accuracy on the Medium setup, compared to the baseline model. For the CSP with attention variants, spatial-domain attention modules and channel-domain attention modules have little difference in MR^{-2} values on each setup.

As for the BCNet variants, we can observe that all variants of BCNet largely improve the MR^{-2} on almost all setups, compared to the baseline model and corresponding CSP variants. The results indicate the structure of our BCNet is more sensitive to attention modules, and this is because the BCNet is aided by the visibility semantic.

Adopting attention modules has become a popular approach, but during our experiments we find the attention modules may not be as powerful as we imagined. Applying two pre-defined balanced hyper-parameters to the entire heatmap produces the best MR^{-2} performance on the Reasonable, Bare, and Partial setups, and beats the baseline model on almost all setups except the Heavy setup.

5.3.3 Evaluation and Analysis

In table 5.4, we compare our proposed BCNet variants with the baseline detector and other existing detectors. From the experimental results of exisiting detectors, we can find FPN is effective to benefit the accuracy of detectors: RetinaNet was proposed and declared to beat the performance of Faster R-CNN. However, after applying FPN to Faster R-CNN, its accuracy is dramatically enhanced. The image resolution also has an impact on the detector's accuracy. Faster R-CNN + ResNet-50 is trained with a larger resolution comparing with the Faster R-CNN + HRNet that trained with the image resolution of 256×512, and they achieve a pretty similar performance, especially on Reasonable, Bare, and Partial setups. When Faster R-CNN* with backbones ResNet-50 is trained with the same resolution of 256×512, it is found that the HRNet backbone has a contribution to the accuracy. Therefore, we it is obvious that the backbone of detectors has a non-ignored effect on detector's accuracy.

However, there is a trade-off among backbone choices, image resolution, batch size, and some other hyper-parameters with a GPU which has the limited computing power and memory. This is an important issue that needs to be solved when applying deep learning-based detectors to vehicular GPUs to support autonomous vehicles in real-world scenarios.

Our BCNet variants produce overall better results than the CSP variants, which indicates that our BCNet structure has a strong ability to boost performance accuracy. We compare our BCNet with other state-of-the-art detectors in Table 5.5, which are trained with different GPU resource. On the one hand, our proposed detector BCNet achieves a

significant improvement in accuracy; on the other hand, our detector reduces the dependency on powerful GPUs. Considering the energy distribution of autonomous vehicles, the computing power and the number of vehicular GPUs are limited. Therefore, achieving promising performance accuracy with limited computing power is a critical prerequisite for applying autonomous vehicles to real-world scenarios.

The improvement of our BCNet lies in the utilization of the visible part semantic, which enables our detector to have more confidence to detect the pedestrians, especially when the pedestrian is under slight occlusion or unoccluded. Compared to the baseline model, our detector does not achieve improvement on the Heavy setup, but it is still better than some detectors that were designed for occlusion handling, such as RepLoss [233] and OR-CNN [253]. Moreover, autonomous vehicles are not required to detect heavily occluded pedestrians because the application of ITS enables all vehicles to exchange information.

By leveraging this capability of ITS, the images captured from different angles are shared, and for each pedestrian on the road, among all shared images, there may be at least one image where they are unoccluded or slightly occluded. The vehicle is capable of sending the detection results of pedestrians to all other nearby vehicles, so that every autonomous vehicle can be aware of pedestrians' locations even if it cannot detect the heavy occluded pedestrian from images captured by itself.

5.3.4 Test on the Additional Dataset

We apply our model to test on the ETH dataset without any other training or fine-tuning to study our model's generalization ability, and we plot the miss rate against FPPI in log scale in Fig. 5.3 and Fig. 5.4 by using the toolbox[1].

ETH works as an additional test dataset with 1804 test images. As shown in Fig. 5.3(a), the baseline model trained on 1 GPU scores 39.19% on MR^{-2}. The best performer is BCNet-ATT-C5, which scores 34.66%. BCNet-ATT-B3 obtains 35.92% on MR^{-2}, which is also better than the 36.77% obtained by the BCNet with pre-defined hyper-parameters on the ETH dataset. Three CSP variants beat the baseline model, with the help of attention modules. CSP-ATT-B1 performs even better than the CSP model trained on 4 GPUs.

The resolution of the images from the ETH dataset is only 640×480. It could harm the effectiveness of the CKVP because a minor offset can cause a considerable impact when generating the detection results. However, 8 BCNet variants perform better than the CSP with 1 GPU, and four among them perform even better than the CSP with 4 GPUs, which

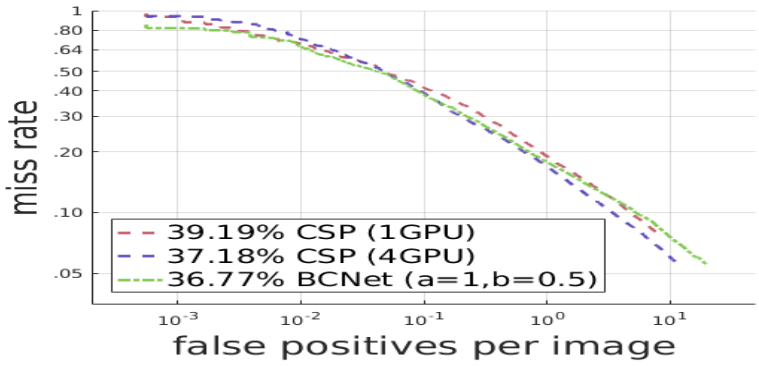
[1] www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/index.html

demonstrates our detectors' strong ability in boosting detection accuracy, even with less computing power.

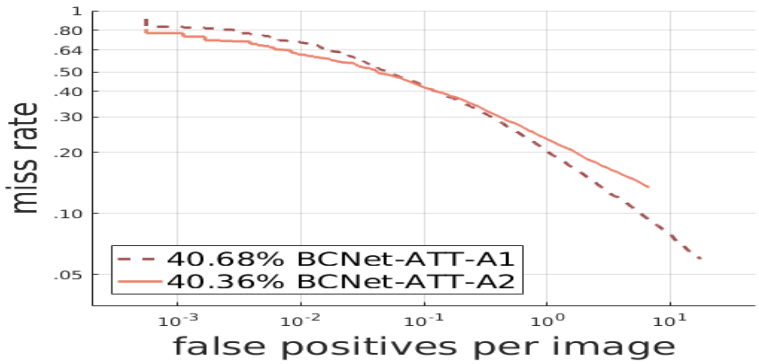
Our detector has considerable generalization ability and achieves a moderate performance on the ETH dataset. We consider that the performance could be significantly enhanced if the image resolution is large enough. With the progress of vehicular cameras, the quality and resolution of images will be enhanced, which will benefit our model to locate the CKFB and CKVP of each pedestrian.

5.4 Evaluation Demo

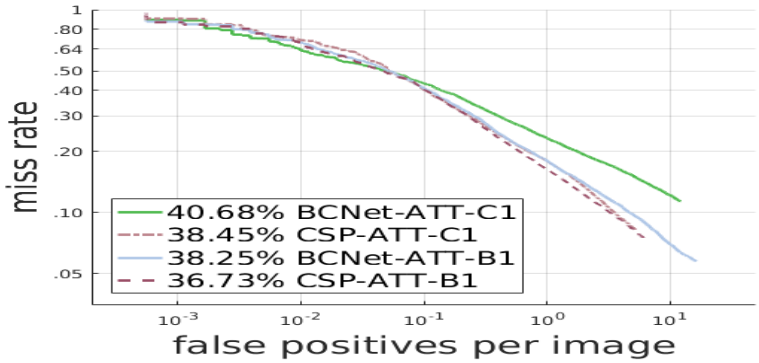
In order to demonstrate our detector's performance, we visualize some detection results by plotting the predicted bboxes on the test image, as illustrated on Figure 5.5, Figure 5.6, and Figure 5.7. We notice that the baseline model generate more false positive predictions, such as the window of the cab. Our proposed BCNet is able to detect very small-scale pedestrians in Figure 5.6(b),i.e., the person through the window, when we lower the score threshold to 0.1. Even if that detection result should be ignored if we consider context features: that person is inside a property, not on the road. It proves out BCNet has a strong feature extraction ability. Our BCNet is the only detector that does not predict bounding boxes in the background, as shown in Figure 5.7.



(a) ETH dataset 1

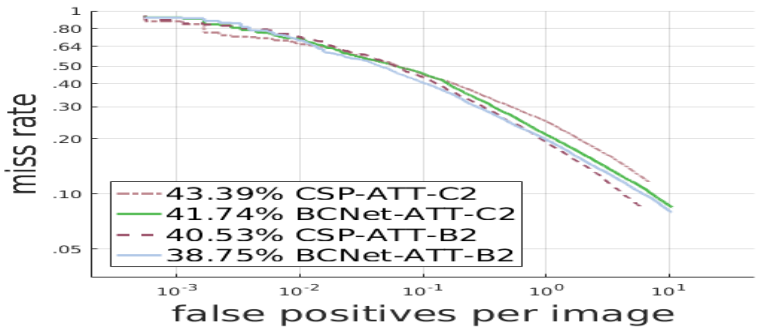


(b) ETH dataset 2

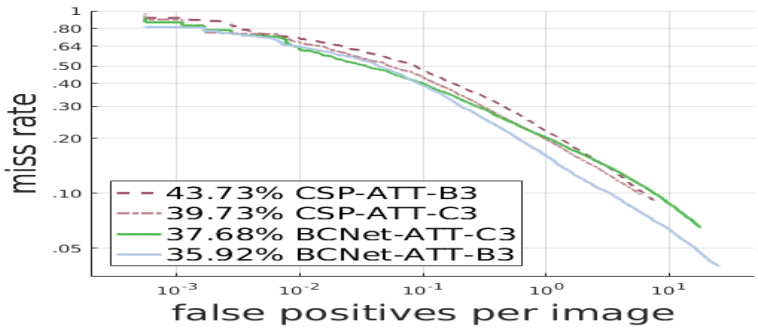


(c) ETH dataset 3

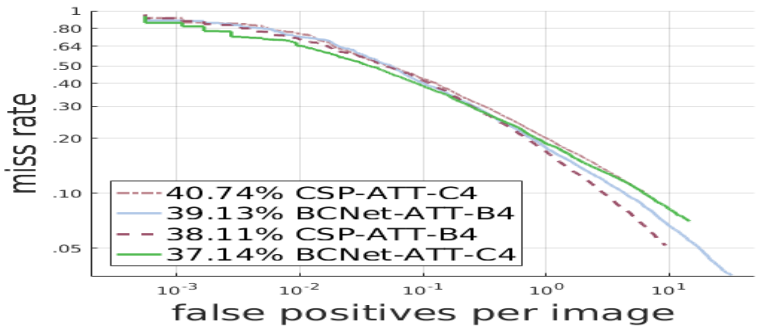
Figure 5.3: ETH dataset results



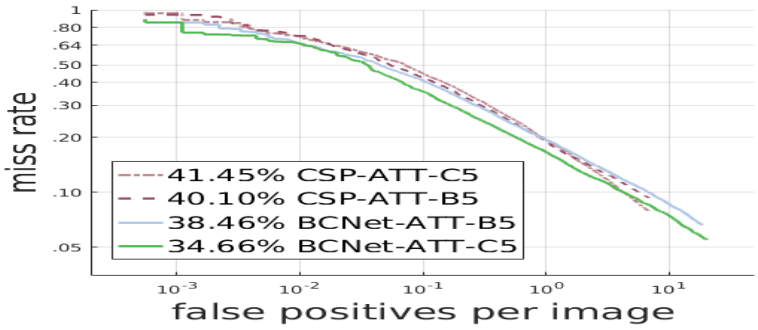
(a) ETH dataset 4



(b) ETH dataset 5

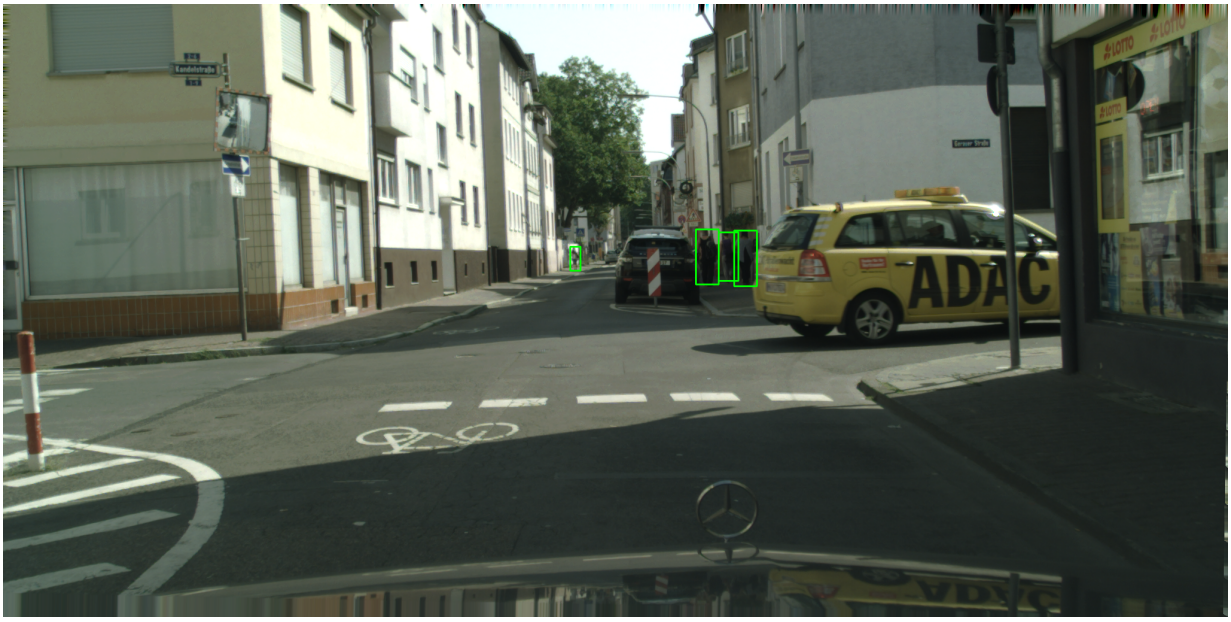


(c) ETH dataset 6

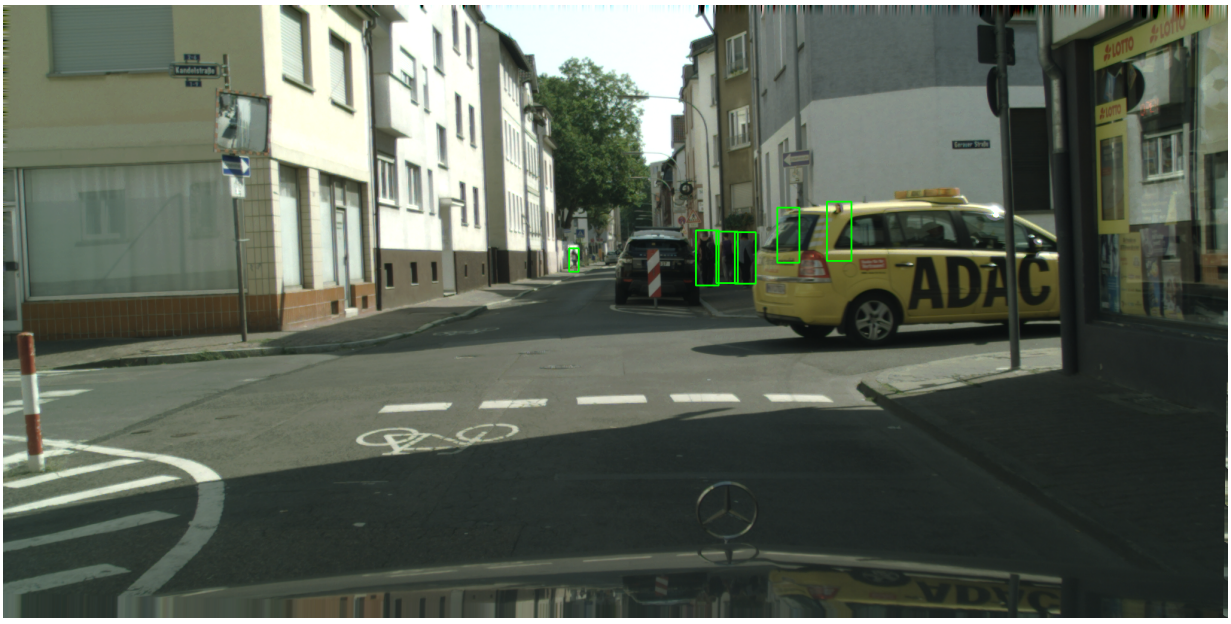


(d) ETH dataset 7

Figure 5.4: ETH dataset results



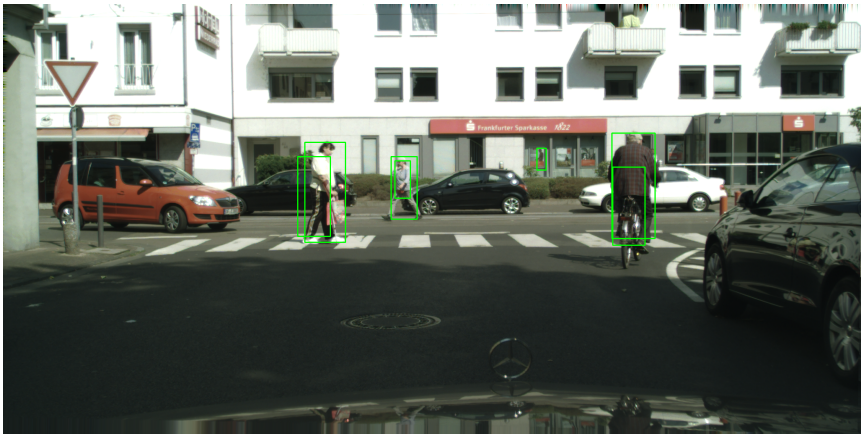
(a) The bboxes predicted by our BCNet.



(b) The bboxes predicted by the baseline model.

Figure 5.5: Plotting prediction results and comparison with the baseline model.

(a) The bboxes predicted by our BCNet with the score threshold of 0.3.



(b) The bboxes predicted by our BCNet with the score threshold of 0.1.

(c) The bboxes predicted by the baseline model.

Figure 5.6: Plotting prediction results and comparison with the baseline model.

(a) Faster R-CNN + FPN (HRNet)

(b) Faster R-CNN + FPN (ResNet-50)

(c) Faster R-CNN* + FPN (ResNet-50)

(d) RetinaNet

(e) GA-RetinaNet

(f) ALFNet

(g) FCOS

(h) BCNet (ours)

Figure 5.7: Plotting prediction results and comparison with the other models.

Table 5.3: Experimental settings of detectors' training.

	Backbone	***Training Image Resolution***	***Batch size***	***Epochs***	***Learning Rate***
Faster R-CNN+FPN	HRNet	256×512	1	60	0.00125
Faster R-CNN+FPN	ResNet-50	608×1216	1	60	0.002
Faster R-CNN*+FPN	ResNet-50	256×512	1	60	0.002
RetinaNet	ResNet-50	456×912	2	60	0.00125
GA-RetinaNet	ResNet-50	256×512	2	60	0.00125
ALFNet	ResNet-50	640×1280	2	100	0.0001
FCOS	ResNet-50	608×1216	1	60	0.0007
CSP	ResNet-50	640×1280	2	100	0.00005
BCNet and variants	ResNet-50	640×1280	2	100	0.00005

Table 5.4: Experimental results on the CityPersons validation set.

	Reasonable(%)	*Bare(%)*	*Partial(%)*	*Heavy(%)*	*Small(%)*	*Medium(%)*	*Large(%)*
Faster R-CNN+FPN(HRNet)	16.83	11.35	16.51	**49.65**	22.64	7.75	9.84
Faster R-CNN+FPN(ResNet)	16.23	11.16	16.58	51.91	24.64	9.42	8.76
Faster R-CNN*+FPN(ResNet)	17.64	12.09	18.30	54.96	28.32	8.99	9.45
RetinaNet	20.99	13.14	19.06	54.20	27.48	9.40	11.84
GA-RetinaNet	18.91	11.90	19.54	57.59	29.62	10.70	10.84
ALFNet	14.87	9.81	14.84	55.43	40.98	30.35	12.00
FCOS	20.03	14.12	20.37	56.69	32.46	10.42	11.72
CSP	12.14	8.31	11.08	50.53	15.95	5.17	6.64
BCNet ($\alpha = 1$, $\beta = 0.5$)	**9.82***	**5.83***	**9.23***	53.34	**12.98**	**3.30**	**6.10**
BCNet-Att-A1	**10.11**	**6.37**	**10.11**	59.04	**13.65**	4.86	**5.64**
BCNet-Att-A2	10.96	7.01	10.13	**58.58**	16.13	**4.16**	6.04
CSP-Att-B1	12.70	8.68	12.23	52.19	18.16	3.82	7.52
BCNet-ATT-B1	11.42	7.19	10.69	60.13	14.77	4.74	6.60
CSP-Att-C1	12.26	7.93	11.79	**50.51**	17.29	3.76	7.28
BCNet-Att-C1	**10.24**	**6.41**	**9.50**	59.25	**14.21**	**3.08**	**6.13**
CSP-Att-B2	13.06	9.36	11.82	**52.65**	16.93	5.31	7.55
BCNet-ATT-B2	**10.90**	**7.26**	**9.73**	59.05	**14.30**	4.46	**6.21**
CSP-Att-C2	13.20	9.12	12.41	53.51	18.14	5.39	7.89
BCNet-Att-C2	11.56	7.32	10.84	59.50	15.42	**3.57**	6.47
CSP-Att-B3	11.99	8.24	10.84	53.73	15.58	3.50	7.25
BCNet-ATT-B3	**10.41**	**6.84**	10.44	59.89	15.14	4.96	**5.50***
CSP-Att-C3	11.72	7.51	11.47	**49.58**	16.97	4.60	6.29
BCNet-Att-C3	10.92	7.01	**10.03**	60.08	**14.23**	**3.19**	5.87
CSP-Att-B4	12.44	8.18	12.17	**49.58**	17.51	5.11	7.32
BCNet-ATT-B4	**10.40**	6.65	**10.11**	58.96	**12.54***	**3.21**	6.82
CSP-Att-C4	12.42	8.28	12.16	53.66	17.74	4.92	6.84
BCNet-Att-C4	10.57	**6.57**	10.24	60.15	14.59	4.89	**5.78**
CSP-Att-B5	12.45	8.53	11.34	**50.09**	15.74	4.83	7.35
BCNet-ATT-B5	**10.10**	7.04	**9.81**	58.69	**13.85**	**2.51***	6.29
CSP-Att-C5	12.88	8.21	13.17	51.12	19.42	4.91	6.94
BCNet-Att-C5	10.64	**6.70**	10.13	58.89	16.13	4.02	**5.87**

[1] The results in boldface indicate the best performance of each block. The results end with * are the best on the corresponding subsets.

[2] All the models are trained and tested with the same GPU resource (Nvidia GeForce GTX 1080 Ti).

[3] The resolution of training images of Faster R-CNN*+FPN(ResNet) is 256×512.

Table 5.5: Experimental results on the CityPersons validation set. The results in boldface indicate the best performance.

	GPU	***Reasonable***	***Bare***	***Partial***	***Heavy***	***Small***	***Medium***	***Large***
Faster R-CNN [249] [195]	-	15.4	-	-	64.83	25.6	7.2	7.9
Faster R-CNN + Semantic [249]	-	14.8	-	-	-	22.6	6.7	8.0
Faster R-CNN + ATT-self [252]	-	20.93	-	-	58.33	-	-	-
TLL [209]	-	15.5	10.0	17.2	53.6	-	-	-
RepLoss [233]	4 GPUs	13.2	7.6	16.8	56.9	-	-	-
OR-CNN [253]	2 GPUs (Titan X)	12.8	6.7	15.3	55.7	-	-	-
ALFNet [161]	2 GPUs (1080Ti)	12.0	8.4	11.4	51.9	19.0	5.7	6.6
RFBNet + adaptive-NMS [159]	4 GPUs (Titan X)	12.7	7.6	11.7	51.9	-	-	-
MGAN [186]	1 GPU	11.5	-	-	51.7	-	-	-
CSP [162]	4 GPUs (1080Ti)	11.0	7.3	10.4	**49.3**	16.0	3.7	6.5
BCNet ($\alpha = 1$, $\beta = 0.5$)	1 GPU (1080Ti)	**9.8**	**5.8**	**9.2**	53.3	**13.0**	**3.3**	**6.1**

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis is a research on pedestrian detection for supporting autonomous vehicles. We proposed a deep learning-based detector BCNet to improve the detection accuracy. We introduced our methods from three main components: pre-processing with annotations and images, construct the detection model, and post-processing with the prediction results. We compared our BCNet with the baseline detector and other existing detectors in the same environment, and it is shown our BCNet achieved a better accuracy when the pedestrian is under reasonable occlusion.

In Chapter 1, we stated the background and motivation of our work. The appearance of ITS and autonomous vehicles is accompanied by the requirements for the efficiency of pedestrian detection. After analysis, we believe that improving the detection accuracy of slightly obstructed pedestrians is crucial. Correspondingly, we proposed a novel semantic fusion method to address this problem.

In Chapter 2, we introduced the preliminary knowledge and some basic concepts that used in the computer vision and pedestrian detection area before we introduced the most representative backbone models and classic detection frameworks. Then, we listed and compared 6 popular pedestrian detection datasets. We pointed out the inherent attributes of pedestrian detection which are different from general object detection.

In Chapter 3, we made a literature review from four aspects: occlusion handling, multi-scale feature extraction, multi-scope data utilization, and hard negative processing. We summarized the commonalities of the methods in each aspect, and pointed out the specific contributions and innovations of each method.

In Chapter 4, we explained our proposed detector BCNet from three designing phases. Our BCNet leverages the semantic feature fusion to boost the efficiency of the detector. We analyzed the effectiveness and benefits of infusing CKVP into CKFB. We further combine each pedestrian's CKFB and CKVP in two types of methods, one is adopting two hyper-parameters and the other one is adopting attention mechanisms.

In Chapter 5, we described the dataset setups and the evaluation metric before we displayed the experimental results on the CityPersons dataset and the ETH dataset. We compared our BCNet with the baseline detector CSP and other existing detectors, including Faster R-CNN, RetinaNet, GA-RetinaNet, FCOS, and ALFNet. From the experimental results, it is obvious that our BCNet achieved a promising improvement, especially when the pedestrians are under reasonable occlusion.

6.2 Future Work

In order to support autonomous vehicles to detect pedestrians in complex scenes in the real world, there are many challenges that need to be solved. Therefore, we list some open challenges in the pedestrian detection area to indicate the directions of future work.

Pedestrian detection aims to detect the pedestrians' locations accurately, efficiently, and robustly in real-time. This goal can be summarized into three sub-goals: accuracy, efficiency, and security. To find a better scope to contribute, we need to have a clear understanding of the challenges in this area. We will discuss some open challenges that might be helpful to guide the future work's directions in Section 6.2.1, Section 6.2.2, and Section 6.2.3 from accuracy, efficiency, and security these three aspects.

6.2.1 Accuracy-related open challenges

In this Subsection, we will discuss the open challenges that relate to accuracy from three aspects: facing various environmental conditions, lacking other information, and weak generalization ability.

Detection under various conditions

Generating fewer false positives and false negatives is a critical aspect of improving the accuracy of pedestrian detection. False positive indicates the detector detects something other than a pedestrian, and this will decrease the transporting efficiency. False negative

happens when the detector fails to detect a pedestrian, which will endanger the pedestrian because the vehicle cannot 'see' this pedestrian from the detection result. Other than the occlusion and complex street scenes that we introduced earlier, there are other factors can cause false positives and false negatives. When autonomous vehicles are applied to real-world scenarios, the bad weather, low illumination, and blurred images caused by high dynamic circumstances are not favorable conditions, but are inevitable.

Varying weather conditions can cause the background of the image to be different. For example, snowy weather makes the captured image whiter, and snow may occlude some parts of the pedestrian in the image [245]. There are other adverse weather conditions to consider, such as rain [127] [231] and fog [149] [153]. Emerging methods for eliminating rain and haze are proposed to restore the images, such as [132] and [246]. However, they did not conduct tasks related to pedestrian detection on the recovered images. In actual use, whether it is favorable or unfavorable weather, autonomous vehicles are required to have stable performance. Therefore, it is desirable to make more comparisons regarding pedestrian detection on both the restored image and the real image.

Autonomous vehicles need to deal with detection in poor light and even in the night. Images captured in the night are much darker than those captured in the daytime, and some features may be lost because the light is not sufficient, making it increasingly difficult to distinguish people from the background [146] [140]. Using the night-vision camera that generates multi-spectral images provides a solution for overcoming this issue, and some related work such as [120] and [148] have explored this. Image enhancement [117] is an effective method to improve the visibility of the images captured in low light. By far, the most common solution is to utilize the thermal-like images to detect the pedestrian at night or in low-light conditions. However, the features extracted between the RGB-image during the day and the nighttime thermal-image are different, and we need to think about how to integrate a model that is applicable both during the day and at night [143].

The highly dynamic property of autonomous vehicles is unavoidable. Moving cameras and pedestrians can cause blurred images [140]. The extracted features of a person in a blurred image are shifted and different from regular features. In order to detect pedestrians from a blurred image, the detector must pre-process and restore the image. It costs more time to compensate and process images before applying detection [169, 207]. Moreover, efficiently determining that the image is blurred before processing and recovering the image is another issue.

Predictions of pedestrians' other information

We cannot assume all the pedestrians are in the same position and facing the cameras. People can be in any position, which is the cause of the deformation problem. Pedestrians can move at different speeds, or they can stand still. Riders can take all different types of riding vehicles, and the features extracted from these various types of pedestrians can be very different from the standstill pedestrian without any occlusion. To cope with these attributes of pedestrians and to make a better prediction, it is very necessary to include the semantic information, contextual features, pose estimation [222], trajectory prediction [238] for information integration. In addition, pedestrians often gather together. Fusing the density of the person group, the distance from the person, and the age of the pedestrians to support ethical decision-making is one of the future directions of research [167]. Therefore, different information should be integrated and utilized in order to protect the pedestrians and perform other functions of autonomous vehicles.

Generalization ability

The strong generalization ability means the detector works well on the current dataset and should also have a similar performance on other datasets. Furthermore, the detector works under the current condition, and is expected to work under other conditions. However, this may not be as easy in the experiment as we think. When the dataset changes, the performance can drop a lot if we do not tune and re-train the model at all [203]. To cope with this issue, using data augmentation to increase the diversity of data is one method; transfer-learning methods can be an asset [137]. Furthermore, the model ensemble can benefit the detector's generalization. Finding effective methods to increase the generalization ability of CNN models is one future direction of research.

6.2.2 Efficiency-related open challenges

In this section, we will discuss open challenges in efficiency from two aspects: energy efficiency and time efficiency. Although vehicular GPUs are more powerful and energy-efficient than most desktop GPUs [204], improving pedestrian detectors' efficiency is still crucial [5, 11].

Energy efficiency

Deep learning-based pedestrian detectors achieve promising accuracy under many conditions, but they cost considerable computational resources and rely highly on computing

power. Reducing the dependency on computing hardware can help deep learning models be more acceptable for autonomous vehicles [200, 44, 108, 241, 37, 213]. Energy efficiency requires the model to be very light-scale, but the light-scale models usually cannot produce results as promising as the large-scale models under the same conditions [119, 84, 114]. How to shrink the model, reduce the computational complexity, and maintain the performance is a challenging problem. Some novel convolutional patterns have been introduced to reduce the computational cost, such as [135, 129]. Parameter pruning is another approach to compress models, which removes the unnecessary network connections by setting a predefined threshold, such as [87, 133]. Emerging methods aimed at reducing computational complexity and model scale are mainly designed and applied to generic CNN networks for classification or general object detection tasks. However, the pedestrian detectors designed for autonomous vehicles differ from general object detection tasks in terms of attributes and requirements. Therefore, there is no guarantee that these methods are effective and applicable to all pedestrian detectors that support autonomous driving systems. In addition, before adopting these approaches in pedestrian detectors, they need to be comprehensively and carefully adjusted to align with the systematical requirements of autonomous vehicles.

Vehicular computers such as NVIDIA DRIVE AGX Pegasus [175] have strong computing power and can handle multiple deep learning tasks simultaneously. Although Pegasus maintains an acceptable power consumption of 500 watts per hour [210], when the pedestrian detection system is applied to autonomous vehicles, it is still necessary to consider the energy distribution of each module in order to avoid affecting other fundamental functions such as driving and decision-making functions. There is very limited work on applying pedestrian detectors to physical autonomous vehicles or simulated platforms (e.g., [176] and [144]) to figure out the efficiency of pedestrian detectors and how they collaborate with other deep learning-based methods. Most of the existing experiments on physical vehicles are focused on data collection, but there is no actual on-board evaluation, such as [226]. Therefore, it is expected that more pedestrian detection methods will be practically tested on the actual autonomous vehicle environment.

Time efficiency

Time complexity is another limitation to consider, because autonomous vehicles need to know the locations of pedestrians in advance. The ideal autonomous vehicles are expected to have a shorter reaction time compared with human drivers (i.e., around 0.70 seconds as we introduced earlier). In practical scenarios, we also need to consider the time of image pre-processing, data transmission, result processing, and decision making [190, 242, 206, 74]. Based on the analysis, the necessity of reducing detectors' inference time consumption

is evident. In addition to improving the inference speed of pedestrian detectors, finding key-frames in video sequences can also enhance the overall efficiency of pedestrian detection by reducing the number of unnecessary detections [85] [199].

In urban scenarios, we can assume the average speed of the moving vehicle is 50 km/h. The safe braking distance is around 12.3 meters without taking the driver's reaction time into the calculation. This braking distance could vary with changes to road conditions or driving speeds. In order to detect pedestrians who are occluded and suddenly crossing the road in time, getting warning notifications through an inter-vehicle communication system is a practical approach [122, 106, 58, 66]. Connected and automated vehicles (CAV) [191, 9, 14] technologies enable surrounding autonomous vehicles to communicate with each other by sending and receiving messages [21, 18]. Building efficient data transmission protocols [92, 39, 42, 8] to reduce the congestion and latency [25, 43, 97, 26] can enhance the time efficiency of the pedestrian detection task. In addition, the pedestrian detection's latency should be considered systematically; tasks supporting autonomous vehicles can be affected by each other. Therefore, physical experiments that combine multiple related tasks, such as pedestrian detection and pedestrian tracking, can help analyze the time latency more reasonably [72, 189]. More simulation experiments for sharing and verifying pedestrian detection results between CAV are thus expected.

6.2.3 Security

One critical concern regarding autonomous vehicles is security [67, 36]. The sensors, cameras, vehicular communication networks, and other modules are not immune to cyber attacks [145, 197, 68, 53, 69, 55, 47, 107, 78, 178], which can cause serious traffic accidents by providing wrong information or commands to autonomous vehicles [4]. More details of the cyber attacks and countermeasures can be found in [118]. In this section, we would like to discuss the adversarial attacks and defense problems for pedestrian detection.

Adversarial attack

By adding minor perturbations to images, the detectors can be deceived and prompted to produce incorrect prediction results. Therefore, adversarial attacks can be used to evaluate the robustness of detectors, and studying adversarial attacks and defenses can help improve the robustness of pedestrian detectors [23]. The increasing attention has encouraged more researchers to make efforts for this issue [194].

Adversarial attacks can be categorized into digital adversarial attacks [236] and physical adversarial attacks [86]. The digital domain adversarial attackers can directly feed the digi-

tal version of adversarial examples to pedestrian detectors, which assumes the autonomous driving systems are under attackers' control. The adversarial attack in the physical world is a more realistic assumption. By taking a photo of the printed digital image and feeding this photo to GAN for simulating the nonlinear quantization effect of cameras, D2P [139] transfers digital examples to physical adversarial examples.

However, the various conditions in the physical world degrade the performance of physical adversarial examples explicitly, such as different perspectives and distances. Many adversarial attacks are proposed on object detectors, such as the traffic sign detector [86]. Currently, there is a lack of delicate adversarial example designs to deceive pedestrian detectors.

Defense

The adversarial attack and defense techniques are in a racing condition. The defense technique largely depends on adversarial examples. For example, adversarial training attempts to incorporate adversarial data in the training phase to ensure the robustness of the detector. Other defensive methods aim to randomize the adversarial effects with the intuition that CNN models are robust to random noises, such as [152]. The authors of [131] proposed a gradient-based method to detect if the image is modified by adversarial noises. These defense technologies are still in progress. Currently, there is no universal defense method that can defend against all adversarial attacks.

References

[1] Kaouther Abrougui, Azzedine Boukerche, and Richard Werner Nelem Pazzi. Design and evaluation of context-aware and location-based service discovery protocols for vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):717–735, 2011.

[2] Osama Abumansoor and A. Boukerche. A secure cooperative approach for nonline-of-sight location verification in vanet. *IEEE Transactions on Vehicular Technology*, 61:275–285, 2012.

[3] Noura Aljeri, Kaouther Abrougui, Mohammed Almulla, and Azzedine Boukerche. A performance evaluation of load balancing and qos-aware gateway discovery protocol for vanets. In *2013 27th International Conference on Advanced Information Networking and Applications Workshops*, pages 90–94, 2013.

[4] Noura Aljeri, Kaouther Abrougui, Mohammed Almulla, and Azzedine Boukerche. A reliable quality of service aware fault tolerant gateway discovery protocol for vehicular networks. *Wireless Communications and Mobile Computing*, 15(10):1485–1495, 2015.

[5] Noura Aljeri, Mohammed Almulla, and Azzedine Boukerche. An efficient fault detection and diagnosis protocolfor vehicular networks. In *Proceedings of the third ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, pages 23–30, 2013.

[6] Noura Aljeri and Azzedine Boukerche. Performance evaluation of movement prediction techniques for vehicular networks. In *Proceesdings of the IEEE International Conference on Communications*, pages 1–6, 2017.

[7] Noura Aljeri and Azzedine Boukerche. A predictive collision detection protocol using vehicular networks. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–5. IEEE, 2017.

[8] Noura AlJeri and Azzedine Boukerche. An efficient movement-based handover prediction scheme for hierarchical mobile ipv6 in vanets. In *Proceedings of the 15th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, pages 47–54, 2018.

[9] Noura Aljeri and Azzedine Boukerche. Mobility and handoff management in connected vehicular networks. In *Proceedings of the 16th ACM International Symposium on Mobility Management and Wireless Access*, pages 82–88, 2018.

[10] Noura Aljeri and Azzedine Boukerche. A dynamic map discovery and selection scheme for predictive hierarchical mipv6 in vehicular networks. *IEEE Transactions on Vehicular Technology*, 69(1):793–806, 2019.

[11] Noura Aljeri and Azzedine Boukerche. An efficient handover trigger scheme for vehicular networks using recurrent neural networks. In *Proceedings of the 15th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, pages 85–91, 2019.

[12] Noura Aljeri and Azzedine Boukerche. Movement prediction models for vehicular networks: an empirical analysis. *Wireless Networks*, 25(4):1505–1518, 2019.

[13] Noura Aljeri and Azzedine Boukerche. A novel online machine learning based rsu prediction scheme for intelligent vehicular networks. In *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8, 2019.

[14] Noura Aljeri and Azzedine Boukerche. An optimized link duration-based mobility management scheme for connected vehicular networks. In *Proceedings of the 16th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, pages 7–14, 2019.

[15] Noura Aljeri and Azzedine Boukerche. A probabilistic neural network-based road side unit prediction scheme for autonomous driving. In *Proceedings of the IEEE International Conference on Communications*, pages 1–6, 2019.

[16] Noura Aljeri and Azzedine Boukerche. A two-tier machine learning-based handover management scheme for intelligent vehicular networks. *Ad Hoc Networks*, 94:101930, 2019.

[17] Noura Aljeri and Azzedine Boukerche. An adaptive traffic-flow based controller deployment scheme for software-defined vehicular networks. In *Proceedings of the 23rd*

International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, pages 191–198, 2020.

[18] Noura Aljeri and Azzedine Boukerche. Advice-loc: An adaptive vehicle-centric location management scheme for intelligent connected cars. *Ad Hoc Networks*, 107:102223, 2020.

[19] Noura Aljeri and Azzedine Boukerche. Fog-enabled vehicular networks: A new challenge for mobility management. *Internet Technology Letters*, 3(6):e141, 2020.

[20] Noura Aljeri and Azzedine Boukerche. Mobility management in 5g-enabled vehicular networks: Models, protocols, and classification. *ACM Computing Surveys*, 53(5):1–35, 2020.

[21] Noura Aljeri and Azzedine Boukerche. A performance evaluation of time-series mobility prediction for connected vehicular networks. In *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pages 127–131, 2020.

[22] Thanasis Antoniou, Ioannis Chatzigiannakis, Georgios Mylonas, Sotiris Nikoletseas, and Azzedine Boukerche. A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range. In *Proceedings of the 37th Annual Simulation Symposium*, pages 43–52, 2004.

[23] Felix Assion, Peter Schlicht, Florens Greßner, Wiebke Gunther, Fabian Huger, Nico Schmidt, and Umair Rasheed. The attack generator: A systematic approach towards constructing adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1370–1379, 2019.

[24] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. [Online]. Available:https://arxiv.org/abs/1607.06450, 2016. Accessed on: Nov., 2019.

[25] S. Baidya, Y. J. Ku, H. Zhao, J. Zhao, and S. Dey. Vehicular and edge computing for emerging connected and autonomous vehicle applications. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 1–6, 2020.

[26] Athanasios Bamis, Azzedine Boukerche, Ioannis Chatzigiannakis, and Sotiris Nikoletseas. A mobility aware protocol synthesis for efficient routing in ad hoc mobile networks. *Computer Networks*, 52(1):130–154, 2008.

[27] Rodolfo Bezerra Batista, Azzedine Boukerche, and Alba Cristina Magalhaes Alves de Melo. A parallel strategy for biological sequence alignment in restricted memory space. *Journal of Parallel and Distributed Computing*, 68(4):548–561, 2008.

[28] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2903–2910, 2012.

[29] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5561–5569, 2017.

[30] A. Boukerch, L. Xu, and K. EL-Khatib. Trust-based security for wireless ad hoc and sensor networks. *Computer Communications*, 30(11):2413–2427, 2007.

[31] A. Boukerche. *Algorithms and protocols for wireless and mobile ad hoc networks*. John Wiley Sons, 2008.

[32] A. Boukerche, K. El-Khatib, Li Xu, and L. Korba. Sdar: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks. In *29th Annual IEEE International Conference on Local Computer Networks*, pages 618–624, 2004.

[33] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro. Localization systems for wireless sensor networks. *IEEE Wireless Communications*, 14(6):6–12, 2007.

[34] Azzedine Boukerche. A simulation based study of on-demand routing protocols for ad hoc wireless networks. In *Proceedings. 34th Annual Simulation Symposium*, pages 85–92, 2001.

[35] Azzedine Boukerche. *Handbook of algorithms for wireless networking and mobile computing*. CRC Press, 2005.

[36] Azzedine Boukerche, Noura Aljeri, Kaouther Abrougui, and Yan Wang. Towards a secure hybrid adaptive gateway discovery mechanism for intelligent transportation systems. *Security and Communication Networks*, 9(17):4027–4047, 2016.

[37] Azzedine Boukerche, Ioannis Chatzigiannakis, and Sotiris Nikoletseas. A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range. *Computer communications*, 29(4):477–489, 2006.

[38] Azzedine Boukerche, Xiuzhen Cheng, and Joseph Linus. Energy-aware data-centric routing in microsensor networks. In *Proceedings of the ACM international workshop on Modeling analysis and Simulation of Wireless Systems*, page 42–49. Association for Computing Machinery, 2003.

[39] Azzedine Boukerche, Xuzhen Cheng, and Joseph Linus. A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks. *Wireless Networks*, 11(5):619–635, 2005.

[40] Azzedine Boukerche and Amir Darehshoorzadeh. Opportunistic routing in wireless networks: Models, algorithms, and classifications. *ACM Computing Surveys*, 47(2):1–36, 2014.

[41] Azzedine Boukerche and Sajal K Das. Dynamic load balancing strategies for conservative parallel simulations. In *Proceedings 11th Workshop on Parallel and Distributed Simulation*, pages 20–28, 1997.

[42] Azzedine Boukerche, Sajal K Das, and Alessandro Fabbri. Analysis of a randomized congestion control scheme with dsdv routing in ad hoc wireless networks. *Journal of Parallel and Distributed Computing*, 61(7):967–995, 2001.

[43] Azzedine Boukerche, Sajal K Das, and Alessandro Fabbri. Swimnet: a scalable parallel simulation testbed for wireless and mobile networks. *Wireless Networks*, 7(5):467–486, 2001.

[44] Azzedine Boukerche, Sajal K Das, Alessandro Fabbri, and Oktay Yildiz. Exploiting model independence for parallel pcs network simulation. In *Proceedings Thirteenth Workshop on Parallel and Distributed Simulation.*, pages 166–173, 1999.

[45] Azzedine Boukerche, Yan Du, Jing Feng, and Richard Pazzi. A reliable synchronous transport protocol for wireless image sensor networks. In *2008 IEEE Symposium on Computers and Communications*, pages 1083–1089, 2008.

[46] Azzedine Boukerche and Caron Dzermajko. Performance evaluation of data distribution management strategies. *Concurrency and Computation: Practice and Experience*, 16(15):1545–1573, 2004.

[47] Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks. *computer communications*, 28(10):1193–1203, 2005.

[48] Azzedine Boukerche and Xin Fei. A coverage-preserving scheme for wireless sensor network with irregular sensing range. *Ad hoc networks*, 5(8):1303–1316, 2007.

[49] Azzedine Boukerche and Xin Fei. A voronoi approach for coverage protocols in wireless sensor networks. In *IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference*, pages 5190–5194, 2007.

[50] Azzedine Boukerche, Xin Fei, and Regina B Araujo. An optimal coverage-preserving scheme for wireless sensor networks based on local information exchange. *Computer Communications*, 30(14-15):2708–2720, 2007.

[51] Azzedine Boukerche, Sungbum Hong, and Tom Jacob. A distributed algorithm for dynamic channel allocation. *Mobile Networks and Applications*, 7(2):115–126, 2002.

[52] Azzedine Boukerche, Sungbum Hong, and Tom Jacob. An efficient synchronization scheme of multimedia streams in wireless and mobile systems. *IEEE transactions on Parallel and Distributed Systems*, 13(9):911–923, 2002.

[53] Azzedine Boukerche, Kathia Regina Lemos Jucá, Joao Bosco Sobral, and Mirela Sechi Moretti Annoni Notare. An artificial immune based intrusion detection model for computer and telecommunication systems. *Parallel Computing*, 30(5-6):629–646, 2004.

[54] Azzedine Boukerche and Xu Li. An agent-based trust and reputation management scheme for wireless sensor networks. In *GLOBECOM'05. IEEE Global Telecommunications Conference, 2005.*, volume 3, pages 5–pp, 2005.

[55] Azzedine Boukerche, Renato B Machado, Kathia RL Jucá, João Bosco M Sobral, and Mirela SMA Notare. An agent based and biological inspired real-time intrusion detection and security model for computer network operations. *Computer Communications*, 30(13):2649–2660, 2007.

[56] Azzedine Boukerche, Alexander Magnano, and Noura Aljeri. Mobile ip handover for vehicular networks: Methods, models, and classifications. *ACM Computing Surveys*, 49(4):1–34, 2017.

[57] Azzedine Boukerche, Anahit Martirosyan, and Richard Pazzi. An inter-cluster communication based energy aware and fault tolerant protocol for wireless sensor networks. *Mobile Networks and Applications*, 13(6):614–626, 2008.

[58] Azzedine Boukerche, Nathan J McGraw, Caron Dzermajko, and Kaiyuan Lu. Grid-filtered region-based data distribution management in large-scale distributed simulation systems. In *38th Annual Simulation Symposium*, pages 259–266, 2005.

[59] Azzedine Boukerche and Sotiris Nikoletseas. Protocols for data propagation in wireless sensor networks. In *Wireless communications systems and networks*, pages 23–51. 2004.

[60] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Secure localization algorithms for wireless sensor networks. *IEEE Communications Magazine*, 46(4):96–101, 2008.

[61] Azzedine Boukerche, Horacio A.B.F. Oliveira, Eduardo F. Nakamura, and Antonio A.F. Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer Communications*, 31(12):2838–2849, 2008.

[62] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo Freire Nakamura, and Antonio AF Loureiro. Dv-loc: a scalable localization protocol using voronoi diagrams for wireless sensor networks. *IEEE Wireless Communications*, 16(2):50–55, 2009.

[63] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina B Araujo. Hpeq a hierarchical periodic, event-driven and query-based wireless sensor network protocol. In *The IEEE Conference on Local Computer Networks, pages=560–567, year=2005,.*

[64] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina Borges Araujo. A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 157–164, 2004.

[65] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina Borges Araujo. Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments. *Journal of Parallel and Distributed Computing*, 66(4):586–599, 2006.

[66] Azzedine Boukerche, Richard WN Pazzi, and Jing Feng. An end-to-end virtual environment streaming technique for thin mobile devices over heterogeneous networks. *Computer Communications*, 31(11):2716–2725, 2008.

[67] Azzedine Boukerche and Yonglin Ren. A security management scheme using a novel computational reputation model for wireless and mobile ad hoc networks. In *Proceedings of the 5th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 88–95, 2008.

[68] Azzedine Boukerche and Yonglin Ren. A trust-based security system for ubiquitous and pervasive computing environments. *Computer communications*, 31(18):4343–4351, 2008.

[69] Azzedine Boukerche and Yonglin Ren. A secure mobile healthcare system using trust-based multicast scheme. *IEEE Journal on Selected Areas in Communications*, 27(4):387–399, 2009.

[70] Azzedine Boukerche, Cristiano Rezende, and Richard W Pazzi. Improving neighbor localization in vehicular ad hoc networks to avoid overhead from periodic messages. In *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*, pages 1–6, 2009.

[71] Azzedine Boukerche and E Robson. Vehicular cloud computing: Architectures, applications, and mobility. *Computer networks*, 135:171–189, 2018.

[72] Azzedine Boukerche and Amber Roy. Dynamic grid-based approach to data distribution management. *Journal of Parallel and Distributed Computing*, 62(3):366–392, 2002.

[73] Azzedine Boukerche, Amber Roy, and Neville Thomas. Dynamic grid-based multicast group assignment in data distribution management. In *Proceedings of the Distributed Simulation and Real-Time Applications*, pages 47–54, 2000.

[74] Azzedine Boukerche and Samer Samarah. An efficient data extraction mechanism for mining association rules from wireless sensor networks. In *2007 IEEE International Conference on Communications*, pages 3936–3941, 2007.

[75] Azzedine Boukerche and Samer Samarah. A novel algorithm for mining association rules in wireless ad hoc sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(7):865–877, 2008.

[76] Azzedine Boukerche and Carl Tropper. A static partitioning and mapping algorithm for conservative parallel simulations. In *Proceedings of the eighth workshop on Parallel and distributed simulation*, pages 164–172, 1994.

[77] Azzedine Boukerche and Carl Tropper. A distributed graph algorithm for the detection of local cycles and knots. *IEEE Transactions on Parallel and Distributed Systems*, 9(8):748–757, 1998.

[78] Azzedine Boukerche and Damla Turgut. Secure time synchronization protocols for wireless sensor networks. *IEEE Wireless Communications*, 14(5):64–69, 2007.

[79] Azzedine Boukerche, Anis Zarrad, and Regina Araujo. A cross-layer approach-based gnutella for collaborative virtual environments over mobile ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 21(7):911–924, 2009.

[80] M. Braun, S. Krebs, F. Flohr, and D. M. Gavrila. Eurocity persons: A novel benchmark for person detection in traffic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1844–1861, 2019.

[81] Garrick Brazil, Xi Yin, and Xiaoming Liu. Illuminating pedestrians via simultaneous detection & segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4950–4959, 2017.

[82] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *Proceedings of the European Conference on Computer Vision*, pages 354–370, 2016.

[83] Zhaowei Cai, Mohammad Saberian, and Nuno Vasconcelos. Learning complexity-aware cascades for deep pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3361–3369, 2015.

[84] Clayson Celes, Fabrício A Silva, Azzedine Boukerche, Rossana Maria de Castro Andrade, and Antonio AF Loureiro. Improving vanet simulation with calibrated vehicular mobility traces. *IEEE Transactions on Mobile Computing*, 16(12):3376–3389, 2017.

[85] Mingju Chen, Xiaofeng Han, Hua Zhang, Guojun Lin, and MM Kamruzzaman. Quality-guided key frames selection from video stream based on object detection. *Journal of Visual Communication and Image Representation*, 65:102678, 2019.

[86] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–68, 2018.

[87] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Processing Magazine*, 35(1):126–136, 2018.

[88] Siew-Kei Lam Chengju Zhou, Meiqing Wu. SSA-CNN: semantic self-attention CNN for pedestrian detection. [Online]. Available:http://arxiv.org/abs/1902.09080, 2019. Accessed on: Nov., 2019.

[89] Cheng Chi, Shifeng Zhang, Junliang Xing, Zhen Lei, Stan Z. Li, and Xudong Zou. Pedhunter: Occlusion robust pedestrian detector in crowded scenes. In *Proc. AAAI*, pages 10639–10646, 2020.

[90] Sergio Correia, Azzedine Boukerche, and Rodolfo I Meneguette. An architecture for hierarchical software-defined vehicular networks. *IEEE Communications Magazine*, 55(7):80–86, 2017.

[91] R. W. L. Coutinho, A. Boukerche, L. F. M. Vieira, and A. A. F. Loureiro. Geographic and opportunistic routing for underwater sensor networks. *IEEE Transactions on Computers*, 65(2):548–561, 2016.

[92] Rodolfo WL Coutinho, Azzedine Boukerche, Luiz FM Vieira, and Antonio AF Loureiro. Gedar: geographic and opportunistic routing protocol with depth adjustment for mobile underwater sensor networks. In *Proceedings of the IEEE International Conference on communications*, pages 251–256, 2014.

[93] Rodolfo WL Coutinho, Azzedine Boukerche, Luiz FM Vieira, and Antonio AF Loureiro. Design guidelines for opportunistic routing in underwater networks. *IEEE Communications Magazine*, 54(2):40–48, 2016.

[94] Rodolfo WL Coutinho, Azzedine Boukerche, Luiz FM Vieira, and Antonio AF Loureiro. Underwater wireless sensor networks: A new challenge for topology control–based systems. *ACM Computing Surveys*, 51(1):1–36, 2018.

[95] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[96] Arthur Daniel Costea and Sergiu Nedevschi. Semantic channels for fast pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2360–2368, 2016.

[97] Amir Darehshoorzadeh and Azzedine Boukerche. Underwater sensor networks: A new challenge for opportunistic routing protocols. *IEEE Communications Magazine*, 53(11):98–107, 2015.

[98] Robson E De Grande and Azzedine Boukerche. Dynamic balancing of communication and computation load for hla-based simulations on large-scale distributed systems. *Journal of Parallel and Distributed Computing*, 71(1):40–52, 2011.

[99] Horacio Antonio Braga Fernandes De Oliveira, Azzedine Boukerche, Eduardo Freire Nakamura, and Antonio Alfredo Ferreira Loureiro. An efficient directed localization recursion protocol for wireless sensor networks. *IEEE Transactions on Computers*, 58(5):677–691, 2008.

[100] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[101] Jimmy Ba Diederik P. Kingma. Adam: A method for stochastic optimization. [Online]. Available:https://arxiv.org/abs/1412.6980, 2014. Accessed on: Sept., 2019.

[102] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–311, 2009.

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

[104] Fabio Henrique Kiyoiti dos Santos Tanaka and Claus Aranha. Data augmentation using gans. [Online]. Available:http://arxiv.org/abs/1904.09135, 2019. Accessed on: Nov., 2019.

[105] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6569–6578, 2019.

[106] Elie El Ajaltouni, Azzedine Boukerche, and Ming Zhang. An efficient dynamic load balancing scheme for distributed simulations on a grid infrastructure. In *12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications*, pages 61–68, 2008.

[107] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. Diagnosing mobile ad-hoc networks: two distributed comparison-based self-diagnosis protocols. In *Proceedings of the 4th ACM international workshop on Mobility management and wireless access*, pages 18–27, 2006.

[108] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. Performance analysis of a distributed comparison-based self-diagnosis protocol for wireless ad-hoc networks. In *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, pages 165–172, 2006.

[109] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. A distributed fault identification protocol for wireless and mobile ad hoc networks. *Journal of parallel and distributed computing*, 68(3):321–335, 2008.

[110] A. Ess, B. Leibe, and L. Van Gool. Depth and appearance for mobile scene analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[111] L. Figueiredo, I. Jesus, J. A. T. Machado, J. R. Ferreira, and J. L. Martins de Carvalho. Towards the development of intelligent transportation systems. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 1206–1211, 2001.

[112] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan. Synthetic data augmentation using gan for improved liver lesion classification. In *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pages 289–293, 2018.

[113] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[114] Baraq Ghaleb, Ahmed Y Al-Dubai, Elias Ekonomou, Ayoub Alsarhan, Youssef Nasser, Lewis M Mackenzie, and Azzedine Boukerche. A survey of limitations and enhancements of the ipv6 routing protocol for low-power and lossy networks: A focus on core operations. *IEEE Communications Surveys & Tutorials*, 21(2):1607–1635, 2018.

[115] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[116] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[117] X. Guo, Y. Li, and H. Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26(2):982–993, 2017.

[118] Rajesh Gupta, Sudeep Tanwar, Neeraj Kumar, and Sudhanshu Tyagi. Blockchain-based security attack resilience schemes for autonomous vehicles in industry 4.0: A systematic review. *Computers & Electrical Engineering*, 86:106717, 2020.

[119] Hadi Habibzadeh, Tolga Soyata, Burak Kantarci, Azzedine Boukerche, and Cem Kaptan. Sensing, communication and security planes: A new challenge for a smart city system design. *Computer Networks*, 144:163–200, 2018.

[120] Hangil Choi, S. Kim, Kihong Park, and K. Sohn. Multi-spectral pedestrian detection based on accumulated object proposal with fully convolutional networks. In *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 621–626, 2016.

[121] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 297–312, 2014.

[122] A. Hbaieb, J. Rezgui, and L. Chaari. Pedestrian detection for autonomous driving within cooperative communication system. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, pages 1–6, 2019.

[123] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[124] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[125] Yihui He, Xiangyu Zhang, Marios Savvides, and Kris Kitani. Softer-nms: Rethinking bounding box regression for accurate object detection. [Online]. Available:http://arxiv.org/abs/1809.08545, 2018. Accessed on: Nov., 2019.

[126] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. [Online]. Available:http://arxiv.org/abs/1207.0580, 2012. Accessed on: Nov., 2019.

[127] Mazin Hnewa and Hayder Radha. Rain-adaptive intensity-driven object detection for autonomous vehicles. In *Proceedings of the SAE Technical Paper*, 2020.

[128] Jan Hosang, Mohamed Omran, Rodrigo Benenson, and Bernt Schiele. Taking a deeper look at pedestrians. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4073–4082, 2015.

[129] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. [Online]. Available:http://arxiv.org/abs/1704.04861, 2017. Accessed on: Nov., 2019.

[130] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[131] Shengyuan Hu, Tao Yu, Chuan Guo, Wei-Lun Chao, and Kilian Q Weinberger. A new defense against adversarial images: Turning a weakness into a strength. In *Proceedings of the Conference and Workshop on Neural Information Processing Systems*, pages 1635–1646. 2019.

[132] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[133] Qiangui Huang, Kevin Zhou, Suya You, and Ulrich Neumann. Learning to prune filters in convolutional neural networks. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 709–718, 2018.

[134] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and baselines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[135] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. [Online]. Available:http://arxiv.org/abs/1602.07360, 2016. Accessed on: Nov., 2019.

[136] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, pages 448–456, 2015.

[137] Jinpeng Li Saad Ullah Akram Ling Shao Irtiza Hasan, Shengcai Liao. Pedestrian detection: The elephant in the room. [Online]. Available:https://arxiv.org/abs/2003.08799, 2020. Accessed on: Nov., 2020.

[138] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Proceedings of the Conference and Workshop on Neural Information Processing Systems*, pages 2017–2025, 2015.

[139] Steve TK Jan, Joseph Messou, Yen-Chen Lin, Jia-Bin Huang, and Gang Wang. Connecting the digital and physical world: Improving the robustness of adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 962–969, 2019.

[140] I. Jegham and A. Ben Khalifa. Pedestrian detection in poor weather conditions using moving camera. In *Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications*, pages 358–362, 2017.

[141] Yun-chen Chen Yao Hu Steven C.H. Hoi Jianke Zhu Jialiang Zhang, Lixiang Lin. CSID: center, scale, identity and density-aware pedestrian detection in a crowd. [Online]. Available:https://arxiv.org/abs/1910.09188, 2019. Accessed on: Nov., 2019.

[142] Xiaoheng Jiang, Yanwei Pang, Xuelong Li, and Jing Pan. Speed up deep neural network based pedestrian detection by sharing features across multi-scale models. *Neurocomputing*, 185:163–170, 2016.

[143] Shu Wang Jingjing Liu, Shaoting Zhang and Dimitris Metaxas. Multispectral deep neural networks for pedestrian detection. In *Proceedings of the British Machine Vision Conference*, pages 73.1–73.13, 2016.

[144] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi. Autoware on board: Enabling autonomous vehicles with embedded systems. In *Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems*, 2018.

[145] Kevin Heaslip Ryan Gerdes Kaveh Bakhsh Kelarestaghi, Mahsa Foruhandeh. Survey on vehicular ad hoc networks and its access technologies security vulnerabilities and countermeasures. [Online]. Available:https://arxiv.org/abs/1903.01541, 2019. Accessed on: Nov., 2020.

[146] Jong Hyun Kim, Ganbayar Batchuluun, and Kang Ryoung Park. Pedestrian detection based on faster r-cnn in nighttime by fusing deep convolutional features of successive images. *Expert Systems with Applications*, 114:15 – 33, 2018.

[147] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Conference and Workshop on Neural Information Processing Systems*, pages 1097–1105, 2012.

[148] S. S. S. Kruthiventi, P. Sahay, and R. Biswal. Low-light pedestrian detection from rgb images using multi-modal knowledge distillation. In *Proceedings of the IEEE International Conference on Image Processing*, pages 4207–4211, 2017.

[149] M. Kutila, P. Pyykönen, H. Holzhüter, M. Colomb, and P. Duthon. Automotive lidar performance verification in fog and rain. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 1695–1701, 2018.

[150] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision*, pages 734–750, 2018.

[151] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[152] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. In *Proceedings of the Conference and Workshop on Neural Information Processing Systems*, pages 9464–9474, 2019.

[153] G. Li, Y. Yang, and X. Qu. Deep learning approaches on pedestrian detection in hazy weather. *IEEE Transactions on Industrial Electronics*, pages 1–1, 2019. Early Access.

[154] Jianan Li, Xiaodan Liang, ShengMei Shen, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Scale-aware fast r-cnn for pedestrian detection. *IEEE Transactions on Multimedia*, 20(4):985–996, 2017.

[155] Chunze Lin, Jiwen Lu, Gang Wang, and Jie Zhou. Graininess-aware deep feature learning for pedestrian detection. In *Proceedings of the European Conference on Computer Vision*, pages 732–747, 2018.

[156] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[157] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[158] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014.

[159] Songtao Liu, Di Huang, and Yunhong Wang. Adaptive nms: Refining pedestrian detection in a crowd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6459–6468, 2019.

[160] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, pages 21–37, 2016.

[161] Wei Liu, Shengcai Liao, Weidong Hu, Xuezhi Liang, and Xiao Chen. Learning efficient single-stage pedestrian detectors by asymptotic localization fitting. In *Proceedings of the European Conference on Computer Vision*, pages 618–634, 2018.

[162] Wei Liu, Shengcai Liao, Weiqiang Ren, Weidong Hu, and Yinan Yu. High-level semantic feature detection: A new perspective for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5187–5196, 2019.

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

[164] P. Luo, Y. Tian, X. Wang, and X. Tang. Switchable deep network for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 899–906, 2014.

[165] Abdelhamid Mammeri, Azzedine Boukerche, and Zongzhi Tang. A real-time lane marking localization, tracking and communication system. *Computer Communications*, 73:132–143, 2016.

[166] Anahit Martirosyan, Azzedine Boukerche, and Richard Pazzi. A taxonomy of cluster-based routing protocols for wireless sensor networks. In *International Symposium on Parallel Architectures, Algorithms, and Networks*, pages 247–253, 2008.

[167] S. K. Maurya and A. Choudhary. Deep learning based vulnerable road user detection and collision avoidance. In *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety*, pages 1–6, 2018.

[168] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer. Uav-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Communications Magazine*, 55(3):22–28, 2017.

[169] Tsubasa Minematsu, Hideaki Uchiyama, Atsushi Shimada, Hajime Nagahara, and Rin-ichiro Taniguchi. Adaptive background model registration for moving cameras. *Pattern Recognition Letters*, 96(C):86–95, 2017.

[170] Ahmed Mostefaoui, Mahmoud Melkemi, and Azzedine Boukerche. Localized routing approach to bypass holes in wireless sensor networks. *IEEE transactions on computers*, 63(12):3053–3065, 2013.

[171] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning*, pages 807–814, 2010.

[172] Woonhyun Nam, Piotr Dollár, and Joon Hee Han. Local decorrelation for improved pedestrian detection. In *Proceedings of the Conference and Workshop on Neural Information Processing Systems*, pages 424–432, 2014.

[173] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 483–499, 2016.

[174] Junhyug Noh, Soochan Lee, Beomsu Kim, and Gunhee Kim. Improving occlusion and hard negative handling for single-stage pedestrian detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 966–974, 2018.

[175] Nvidia. Nvidia drive agx pegasus. [Online]. Available:https://www.nvidia.com/en-us/self-driving-cars/drive-platform/hardware/. Accessed on: Nov., 2020.

[176] Nvidia. Nvidia drive sim and drive constellation. [Online]. Available:https://www.nvidia.com/en-us/self-driving-cars/drive-constellation/. Accessed on: Nov., 2020.

[177] Horacio ABF Oliveira, Azzedine Boukerche, Eduardo F Nakamura, and Antonio AF Loureiro. Localization in time and space for wireless sensor networks: An efficient and lightweight algorithm. *Performance Evaluation*, 66(3-5):209–222, 2009.

[178] Horacio ABF Oliveira, Eduardo F Nakamura, Antonio AF Loureiro, and Azzedine Boukerche. Error analysis of localization systems for sensor networks. In *Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pages 71–78, 2005.

[179] Renê Oliveira, Carlos Montez, Azzedine Boukerche, and Michelle S Wangham. Reliable data dissemination protocol for vanet traffic safety applications. *Ad Hoc Networks*, 63:30–44, 2017.

[180] W. Ouyang and X. Wang. A discriminative deep model for pedestrian detection with occlusion handling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3258–3265, 2012.

[181] W. Ouyang, H. Zhou, H. Li, Q. Li, J. Yan, and X. Wang. Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(8):1874–1887, 2018.

[182] Wanli Ouyang and Xiaogang Wang. A discriminative deep model for pedestrian detection with occlusion handling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3258–3265, 2012.

[183] Wanli Ouyang and Xiaogang Wang. Joint deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[184] Sakrapee Paisitkriangkrai, Chunhua Shen, and Anton Van Den Hengel. Strengthening the effectiveness of pedestrian detection with spatially pooled features. In *Proceedings of the European Conference on Computer Vision*, pages 546–561, 2014.

[185] A. Palffy, J. F. P. Kooij, and D. M. Gavrila. Occlusion aware sensor fusion for early crossing pedestrian detection. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1768–1774, 2019.

[186] Yanwei Pang, Jin Xie, Muhammad Haris Khan, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Ling Shao. Mask-guided attention network for occluded pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4967–4975, 2019.

[187] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 555–562, 1998.

[188] Constantine P. Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 555–562, 1998.

[189] Richard W Pazzi and Azzedine Boukerche. Propane: A progressive panorama streaming protocol to support interactive 3d virtual environment exploration on graphics-constrained devices. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 11(1):1–22, 2014.

[190] Richard WN Pazzi and Azzedine Boukerche. Mobile data collector strategy for delay-sensitive applications over wireless sensor networks. *Computer Communications*, 31(5):1028–1039, 2008.

[191] Joel Pereira, Cristiano Premebida, Alireza Asvadi, F Cannata, Luis Garrote, and UJ Nunes. Test and evaluation of connected and autonomous vehicles in real-world scenarios. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 14–19, 2019.

[192] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6517–6525, 2017.

[193] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[194] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346 – 360, 2020.

[195] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the Conference and Workshop on Neural Information Processing Systems*, pages 91–99, 2015.

[196] Y. Ren, R. Werner, N. Pazzi, and A. Boukerche. Monitoring patients via a secure and mobile healthcare system. *IEEE Wireless Communications*, 17(1):59–65, 2010.

[197] Yonglin Ren and Azzedine Boukerche. Modeling and managing the trust for wireless and mobile ad hoc networks. In *2008 IEEE International Conference on Communications*, pages 2129–2133, 2008.

[198] Cristiano Rezende, Azzedine Boukerche, Heitor S Ramos, and Antonio AF Loureiro. A reactive and scalable unicast solution for video streaming over vanets. *IEEE Transactions on Computers*, 64(3):614–626, 2014.

[199] Cristiano Rezende, Abdelhamid Mammeri, Azzedine Boukerche, and Antonio AF Loureiro. A receiver-based video dissemination solution for vehicular networks with content transmissions decoupled from relay node selection. *Ad Hoc Networks*, 17:1–17, 2014.

[200] Samer Samarah, Muhannad Al-Hajri, and Azzedine Boukerche. A predictive energy-efficient technique to support object-tracking sensor networks. *IEEE Transactions on Vehicular Technology*, 60(2):656–663, 2010.

[201] Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 92–101, 2010.

[202] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann Lecun. Pedestrian detection with unsupervised multi-stage feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[203] M. Sha and A. Boukerche. Semantic fusion-based pedestrian detection for supporting autonomous vehicles. In *Proceedings of the IEEE Symposium on Computers and Communications*, pages 618–623, 2020.

[204] Weijing Shi, Mohamed Baker Alawieh, Xin Li, and Huafeng Yu. Algorithm and hardware implementation for visual perception system in autonomous vehicle: A survey. *Integration*, 59:148–156, 2017.

[205] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016.

[206] Abdul Jabbar Siddiqui, Abdelhamid Mammeri, and Azzedine Boukerche. Real-time vehicle make and model recognition based on a bag of surf features. *IEEE Transactions on Intelligent Transportation Systems*, 17(11):3205–3219, 2016.

[207] Fabricio A Silva, Azzedine Boukerche, Thais RM Braga Silva, Linnyer B Ruiz, Eduardo Cerqueira, and Antonio AF Loureiro. Vehicular networks: A new challenge for content-delivery-based applications. *ACM Computing Surveys*, 49(1):1–29, 2016.

[208] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015. Accessed on: Nov., 2020.

[209] Tao Song, Leiyu Sun, Di Xie, Haiming Sun, and Shiliang Pu. Small-scale pedestrian detection based on topological line localization and temporal feature aggregation. In *Proceedings of the European Conference on Computer Vision*, pages 536–551, 2018.

[210] Jack Stewart. Self-driving cars use crazy amounts of power, and it's becoming a problem. [Online]. Available:https://www.wired.com/story/self-driving-cars-power-consumption-nvidia-chip/. Accessed on: Nov., 2020.

[211] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[212] Peng Sun, Noura AlJeri, and Azzedine Boukerche. A novel passive road side unit detection scheme in vehicular networks. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–5, 2017.

[213] Peng Sun, Noura AlJeri, and Azzedine Boukerche. An energy-efficient proactive handover scheme for vehicular networks based on passive rsu detection. *IEEE Transactions on Sustainable Computing*, 5(1):37–47, 2018.

[214] Peng Sun, Noura AlJeri, and Azzedine Boukerche. A fast vehicular traffic flow prediction scheme based on fourier and wavelet analysis. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2018.

[215] Peng Sun, Noura AlJeri, and Azzedine Boukerche. Dacon: A novel traffic prediction and data-highway-assisted content delivery protocol for intelligent vehicular networks. *IEEE Transactions on Sustainable Computing*, 5(4):501–513, 2020.

[216] Peng Sun, Noura Aljeri, and Azzedine Boukerche. Machine learning-based models for real-time traffic flow prediction in vehicular networks. *IEEE Network*, 34(3):178–185, 2020.

[217] Siyu Tang, Mykhaylo Andriluka, and Bernt Schiele. Detection and tracking of occluded people. *International Journal of Computer Vision*, 110(1):58–69, 2014.

[218] Yonglong Tian, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning strong parts for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[219] Yonglong Tian, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning strong parts for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1904–1912, 2015.

[220] Yonglong Tian, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Pedestrian detection aided by deep learning semantic tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5079–5087, 2015.

[221] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9627–9636, 2019.

[222] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[223] Lachlan Tychsen-Smith and Lars Petersson. Denet: Scalable real-time object detection with directed sparse sampling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 428–436, 2017.

[224] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of Computer Vision*, 104(2):154–171, 2013.

[225] Tuan-Hung Vu, Anton Osokin, and Ivan Laptev. Context-aware cnns for person head detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2893–2901, 2015.

[226] Heng Wang, Bin Wang, Bingbing Liu, Xiaoli Meng, and Guanghong Yang. Pedestrian recognition and tracking using 3d lidar for autonomous vehicle. *Robotics and Autonomous Systems*, 88:71–78, 2017.

[227] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin. Region proposal by guided anchoring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2960–2969, 2019.

[228] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2974, 2019.

[229] Renfei Wang, Cristiano Rezende, Heitor S Ramos, Richard W Pazzi, Azzedine Boukerche, and Antonio AF Loureiro. Liaithon: A location-aware multipath video streaming scheme for urban vehicular networks. In *2012 IEEE Symposium on Computers and Communications (ISCC)*, pages 436–441, 2012.

[230] Shiguang Wang, Jian Cheng, Haijun Liu, Feng Wang, and Hui Zhou. Pedestrian detection via body part semantic and contextual information with dnn. *IEEE Transactions on Multimedia*, 20(11):3148–3159, 2018.

[231] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson W.H. Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[232] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta. A-fast-rcnn: Hard positive generation via adversary for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2606–2615, 2017.

[233] Xinlong Wang, Tete Xiao, Yuning Jiang, Shuai Shao, Jian Sun, and Chunhua Shen. Repulsion loss: Detecting pedestrians in a crowd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7774–7783, 2018.

[234] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision*, pages 3–19, 2018.

[235] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision*, pages 3–19, 2018.

[236] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1369–1378, 2017.

[237] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[238] H. Xiong, F. B. Flohr, S. Wang, B. Wang, J. Wang, and K. Li. Recurrent neural network architectures for vulnerable road user trajectory prediction. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 171–178, 2019.

[239] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z Li. Convolutional channel features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 82–90, 2015.

[240] Maram Bani Younes and Azzedine Boukerche. Intelligent traffic light controlling algorithms using vehicular networks. *IEEE transactions on vehicular technology*, 65(8):5887–5899, 2015.

[241] Maram Bani Younes and Azzedine Boukerche. A performance evaluation of an efficient traffic congestion detection protocol (ecode) for intelligent transportation systems. *Ad Hoc Networks*, 24:317–336, 2015.

[242] Maram Bani Younes and Azzedine Boukerche. An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems. *Wireless Networks*, 24(7):2451–2463, 2018.

[243] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the International Conference on Learning Representations*, 2016. Accessed on: Nov., 2019.

[244] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018.

[245] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar. The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE Vehicular Technology Magazine*, 14(2):103–111, 2019.

[246] He Zhang and Vishal M. Patel. Densely connected pyramid dehazing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[247] J. Zhang, F. Wang, K. Wang, W. Lin, X. Xu, and C. Chen. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1624–1639, 2011.

[248] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. Is faster r-cnn doing well for pedestrian detection? In *Proceedings of the European Conference on Computer Vision*, pages 443–457, 2016.

[249] S. Zhang, R. Benenson, and B. Schiele. Citypersons: A diverse dataset for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4457–4465, 2017.

[250] Shanshan Zhang, Christian Bauckhage, and Armin B. Cremers. Informed haar-like features improve pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[251] Shanshan Zhang, Rodrigo Benenson, Bernt Schiele, et al. Filtered channel features for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1751–1760, 2015.

[252] Shanshan Zhang, Jian Yang, and Bernt Schiele. Occluded pedestrian detection through guided attention in cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6995–7003, 2018.

[253] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. Occlusion-aware r-cnn: Detecting pedestrians in a crowd. In *Proceedings of the European Conference on Computer Vision*, pages 637–653, 2018.

[254] Xiaowei Zhang, Li Cheng, Bo Li, and Hai-Miao Hu. Too far to see? not really!—pedestrian detection with scale-aware localization policy. *IEEE Transactions on Image Processing*, 27(8):3703–3715, 2018.

[255] Zhenxia Zhang, Richard W Pazzi, and Azzedine Boukerche. A mobility management scheme for wireless mesh networks based on a hybrid routing protocol. *Computer Networks*, 54(4):558–572, 2010.

[256] Chunluan Zhou and Junsong Yuan. Multi-label learning of part detectors for heavily occluded pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3486–3495, 2017.

[257] Chunluan Zhou and Junsong Yuan. Bi-box regression for pedestrian detection and occlusion estimation. In *Proceedings of the European Conference on Computer Vision*, pages 135–151, 2018.

[258] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 840–849, 2019.

[259] Yousong Zhu, Jinqiao Wang, Chaoyang Zhao, Haiyun Guo, and Hanqing Lu. Scale-adaptive deconvolutional regression network for pedestrian detection. In *Proceedings of the Asian Conference on Computer Vision*, pages 416–430, 2016.