Ad Hoc Networks 128 (2022) 102784

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Performance evaluation of CNN-based pedestrian detectors for autonomous vehicles

Mingzhi Sha [*], Azzedine Boukerche

Paradise Research Laboratory, School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Ave., Ottawa, K1N 6N5 ON., Canada

ARTICLE INFO

Keywords:
Intelligent transportation system
Autonomous vehicle
Pedestrian detection
Convolutional neural network

ABSTRACT

With the widespread application of deep learning methodologies, many fields including Intelligent Transportation Systems (ITS) have integrated neural network-based models. In return, the promising performance of neural network-based models attracts more research efforts being paid to the deep learning area. Autonomous vehicles, as the future ITS participants, have achieved tremendous development with the help of convolutional neural network (CNN)-based detectors (e.g., vehicle detectors, lane detectors, pedestrian detectors, traffic sign detectors, etc.) in recent years. However, we have noticed that researchers leveraged different computing power when publishing their experimental results, which could lead to unfair comparisons. In this paper, we focus on CNN-based pedestrian detectors. We conduct a comprehensive comparative study of the representative CNN-based pedestrian detectors, aiming to investigate the influence of experimental settings by eliminating the bias of different experimental environments.

1. Introduction

The emergence of the Intelligent Transportation System (ITS) requires improvement of existing methods in communication, resource allocation, and coordination, so as to improve the efficiency and quality of ITS services to accommodate the increasing number of traffic participants. Meanwhile, the extensive research on the ITS domain provides the possibility for the application of autonomous vehicles, e.g., [1–4]. Both academia and industry show great interest in autonomous vehicles. Therefore, autonomous vehicles attract huge attention recently. Accordingly, many research efforts have been paid to solve the challenges that autonomous vehicles are facing, such as pedestrian detection.

Pedestrian detection is an indispensable task, and it is also a challenge that autonomous vehicles must overcome before being applied to real-world scenarios because safety is not negotiable. Therefore, improving the performance of pedestrian detectors is a hot research topic in academic and industrial fields. In recent years, many works have been proposed, such as [5–9].

The proposed work covers various aspects regarding the performance, for example, some researchers work to improve the detection accuracy when the pedestrian is occluded, such as [10–12]. Some researchers work to infuse parallel branches into the detection network to conduct additional tasks, such as [12–14]. Some researchers innovate novel features and approaches to represent the pedestrians, such as [15,16].

However, we found that the experimental environments of most pedestrian detectors are not unified or even unexplained. Training pedestrian detectors with different amounts of GPUs or different types of GPUs may introduce a significant bias, because computing power may have a non-negligible impact on the performance of the detector. Therefore, the various experimental conditions could have a considerable impact on the detectors' accuracy and efficiency, which makes readers unable to verify if the improvement in the experimental results is due to the effectiveness of the proposed methods or simply because of the powerful computing power. Furthermore, the GPUs are evolved much more powerful than before, making it unfair when researchers directly cite previous experimental results and make a comparison, especially comparisons between detectors published in different years.

The relationship between the GPUs and detectors' performance is the elephant in the room. Adopting a larger number of GPUs or choosing a GPU with higher computing capability will undoubtedly increase the total computational power, thereby boosting the performance of the detector from the perspective of accuracy and efficiency. However, the cost of using GPUs with large-scale computing capabilities is expensive. Besides, autonomous vehicles may not be able to afford such massive computing power consumption. Therefore, the improvement of CNN-based detectors should never rely solely on the use of greater computing power.

* Corresponding author.
E-mail addresses: msha096@uottawa.ca (M. Sha), boukerch@site.uottawa.ca (A. Boukerche).

https://doi.org/10.1016/j.adhoc.2022.102784
Received 15 March 2021; Received in revised form 23 November 2021; Accepted 12 January 2022
Available online 31 January 2022

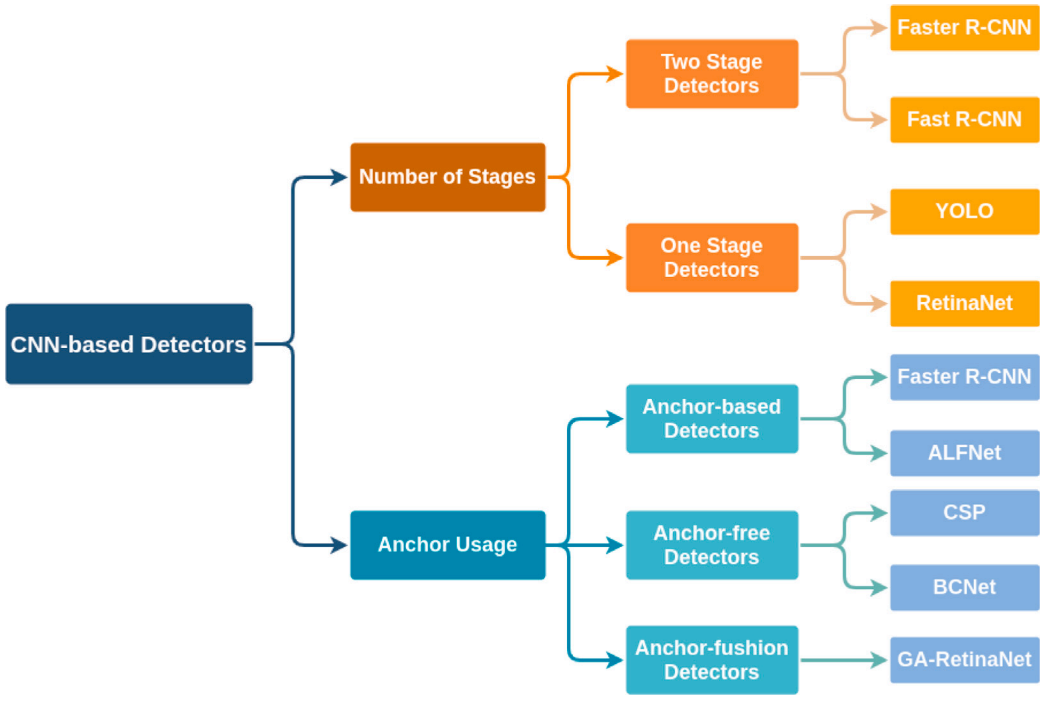


Fig. 1. Existing CNN-based detection methods.

Therefore, we aim to study the performance of existing pedestrian detectors by eliminating the bias of different computational power by training some representative CNN-based pedestrian detectors under the same experimental environment. In this paper, we adopt 7 existing pedestrian detectors, which cover a variety of different categories. We will fairly compare the performance of these detectors and investigate the impact of experimental hyper-parameters on the CityPersons [17] dataset and the ETH [18] dataset.

In the rest of this paper, we will first review the development of CNN-based detectors from two different aspects: the feature extraction backbone and the detection framework in Section 2. Next, we will present the 7 pedestrian detectors we choose and the processing steps in Section 3. Then, we conduct the experiments and evaluations on the CityPersons dataset and the ETH dataset in Section 4. Lastly, the conclusions and the future work will be given in Section 5.

2. Related work

The objective of detection 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 bounding boxes (Bboxes) of the objects, where the Bboxes work to denote the locations of the objects [15,19]. The CNN-based detectors generally have two major components. One is the feature extraction network (i.e., backbone), which contains stacked convolutional layers and pooling layers. The other one is the detection head, which produces the final Bboxes based on the feature maps obtained from the backbone. The detection framework includes the usage of backbone, and how feature maps are processed before and in the detection head. Therefore, in this section, we will review the milestone work from both the feature extraction backbone and the detection framework.

2.1. Feature extraction backbone

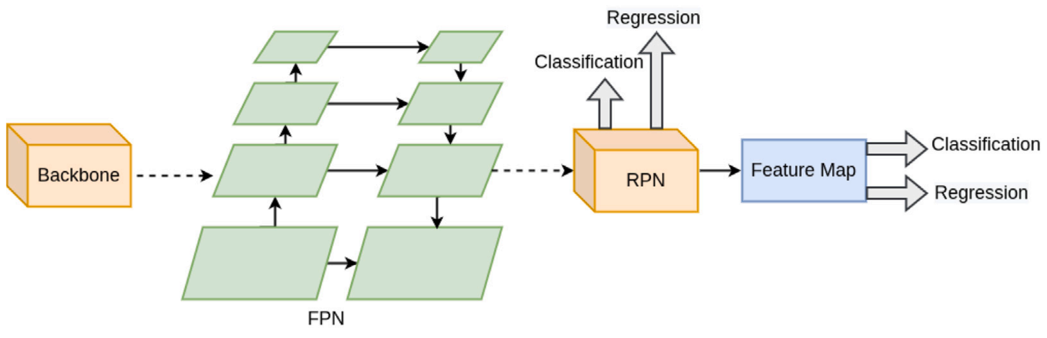
LeNet-5 [20] is an early-stage CNN model that was proposed in 1998, and it was proposed for handling digits recognition task on MNIST [20] dataset, which only contains 10 classes. Because of limitations in the computational power, neural network models were very small-scale and unable to handle hard tasks at that time. In 2012, AlexNet [21] came out and it was trained on two GPUs. The computations are separated equally by two GPUs, and each one is responsible for half of the kernels' computations. 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 [22], either large in sample amount or data size.

VGGNet [23] came out in 2014, which proposed to use 3×3 kernels instead of using the large kernel size like 5×5 or 11×11. The replacement helps to reduce the number of parameters and increases the neural network's depth. The non-linearity of deep layers enables the network model to learn more complicated patterns and thus precisely discriminate more sophisticated features.

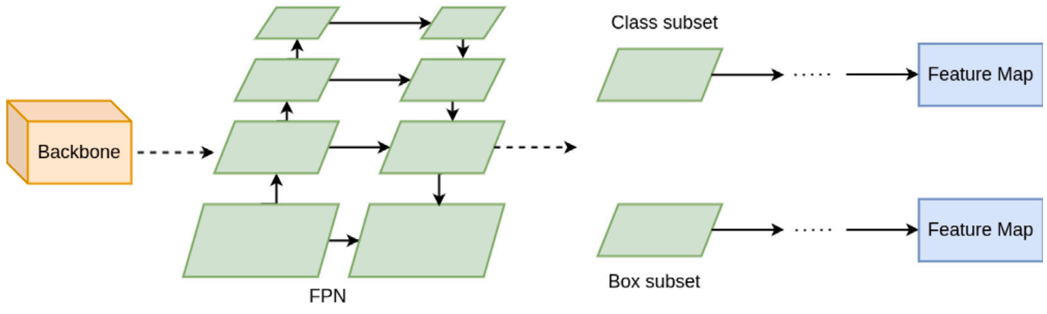
He et al. in [24] did an experiment to compare the performance of two plain networks, one with a depth of 20 and the other one with a depth of 56. Unexpectedly, the performance of the deeper network has a higher train error and test error compared with the shallow network. However, it can be inferred that the degradation performance given by the experiment in [24] is caused by the gradient vanishing and exploding, which makes the deep model hard to converge. To cope with this problem, He et al. proposed a novel shortcut connection by adding the input directly to the output, which makes the network easier to be trained.

The lower-level feature maps have higher resolution, which can provide more accurate information in localization; whereas higher-level feature maps contain stronger semantic values. To leverage both location accuracy and semantic richness, multi-scale feature representation methods are commonly used. FPN [25] was proposed in 2017, which introduced a novel network block to achieve multi-scale feature representation. FPN proposed a feature map top-down pathway by conducting upsampling, and the feature maps from the bottom-up pathway (i.e., the feature maps generated during the feed-forward computation of ResNet) at the corresponding level are further merged by element-wise addition. FPN has been widely adopted as the network neck to boost the accuracy and semantic values since its appearance.

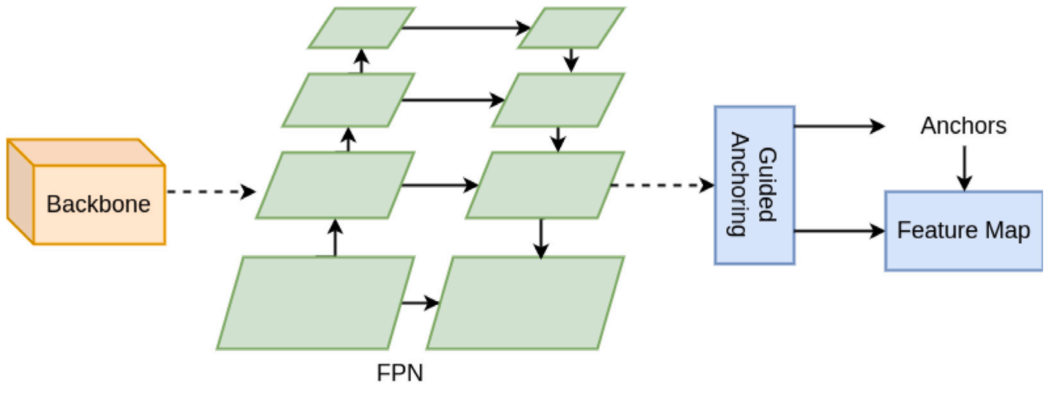
HRNet [26] was designed to maintain localization precise and semantic richness. Different from FPN which recovers the feature map resolutions from low-resolution feature maps, HRNet connects the



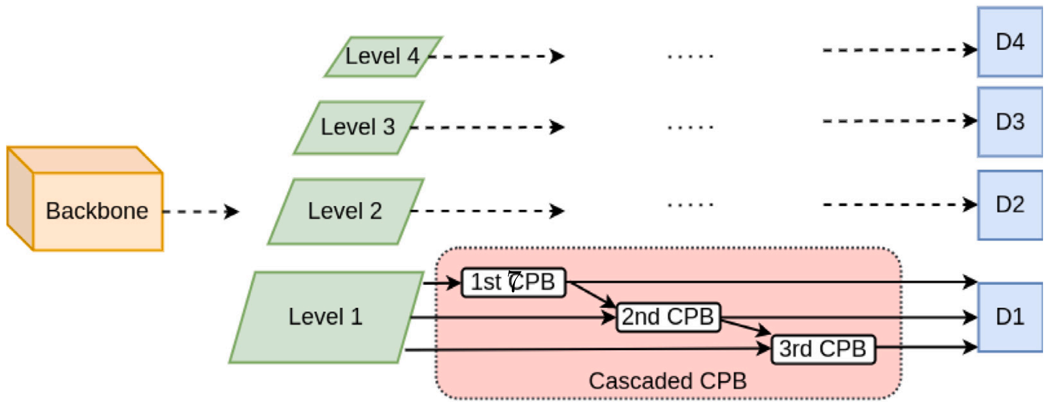
(a) Faster R-CNN + FPN



(b) RetinaNet

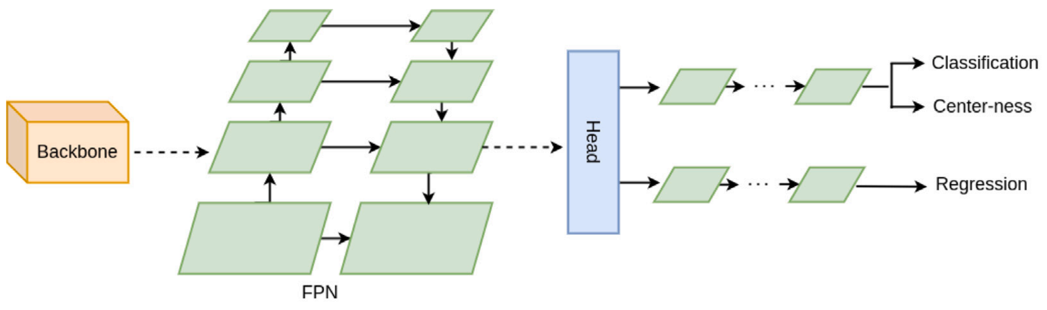


(c) GA-RetinaNet

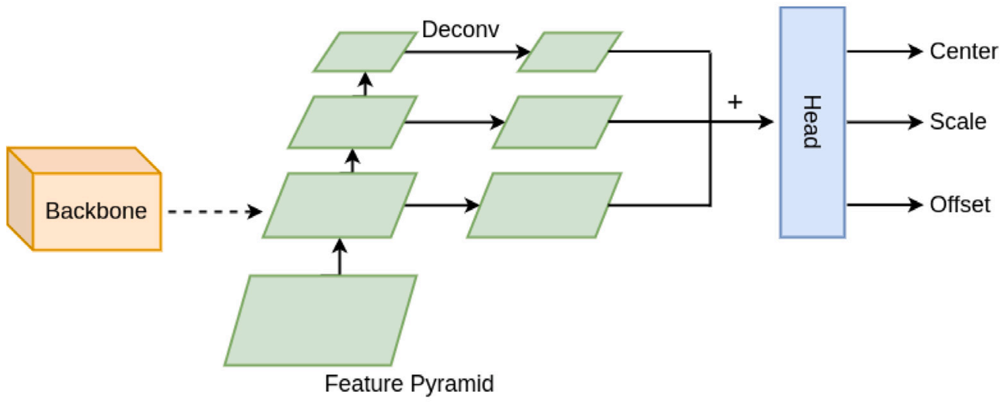


(d) ALFNet

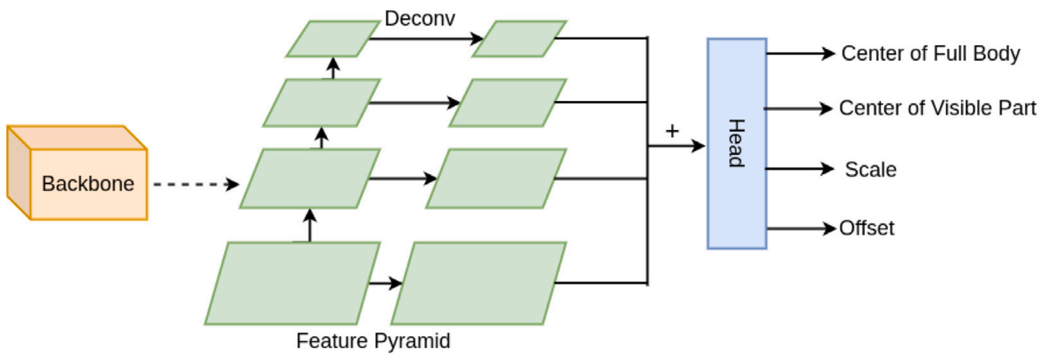
Fig. 2. Model structure illustrations of the chosen detectors.



(e) FCOS



(f) CSP



(g) BCNet

Fig. 2. (*continued*).

multi-resolution convolutional streams in parallel and keeps fusing the feature maps of different resolutions to output high-resolution final feature maps and enrich the semantic features at the meanwhile.

2.2. Detection framework

Pedestrian detection highly relied on traditional machine learning methods before Faster R-CNN achieved promising performance in 2015. Traditional machine learning methods apply the pre-defined hand-crafted feature descriptor to the entire image and perform pattern matching, such as [27] and [28]. Differently, CNN-based methods enable the detector to learn the feature of the target from the given dataset by itself. Powerful feature extraction and learning capabilities have made CNN-based detectors popular, especially since 2015, many innovative methods have emerged.

As shown in Fig. 1, we review the existing detectors from two aspects, one is from the number of stages, and the other is from the usage of anchors. Detectors can be divided into one-stage detectors and two-stage detectors. The detection task is a combination of classifying and locating.

The most classic and representative two-stage detectors include the R-CNN family [29–31], which search for the region of interest (ROI) at the first stage and perform classifying and locating base on the ROI results at the second stage. Though the appearance of two-stage detectors have promising performance in accuracy, they are slow in inference speed. To improve the inference speed, the first one-stage detector–YOLO-v1 [32] was proposed in 2016. Different from two-stage detectors, one-stage detectors predict the class and location simultaneously, without searching ROI proposals first. YOLO-v1 was proposed to detect in real-time, the downside is the poor accuracy caused by the overwhelming negative samples. In order to obtain satisfactory accuracy and speed, the aforementioned FPN was adopted in the one-stage detector RetinaNet [33], and the novel loss function-Focal Loss is used for training, which could alleviate class imbalance.

The concept of ‘anchor’ was proposed in RPN [31], and it has been widely adopted since the appearance. The anchor mechanism replaces blindly searching the windows on the entire image by generating anchor boxes with pre-defined scales and ratios at each fixed anchor point. The anchor mechanism is used to predict ROI proposals in two-stage detectors (e.g., Faster R-CNN [31] and Mask R-CNN [34]) and candidate Bboxes in one-stage detectors (e.g., SSD [35] and YOLO-v2 [36]).

Instead of applying anchors into usage, YOLO-v1 [32] achieved an anchor-free detecting pipeline by dividing the images into grids to generate corresponding predictions. DeNet [37] avoids the use of the anchor mechanism by predicting the four keypoints of the target. Another anchor-free pedestrian detector TLL [38] was designed to predict top-bottom points and the topological line. Since 2019, adopting anchor-free detectors has been a trend, especially keypoint-based detectors, such as CSP and BCNet.

3. Methods

In the experiment, we choose the most representative and trending detector to cover the diversity of different aspects. We adopt Faster R-CNN [31] + Feature Pyramid Network (FPN) [25], RetinaNet [33], RetinaNet with Guided Anchoring (GA-RetinaNet) [39], Asymptotic Localization Fitting Network (ALFNet) [40], Fully Convolutional One-Stage (FCOS) object detector [41], Center and scale prediction-based detector (CSP) [15], and Bi-Center Network (BCNet) [8] in our comparative experiments. They differ not only in the structure of the model, the number of stages, and the use of anchors as shown in Table 1, each detector has very different design purpose. The details of each detector will be given in 3.1, and the model structures are demonstrated in Fig. 2.

3.1. Discussion on the detectors

Before we present the experimental settings and results, we would like to introduce the detectors we choose and explain why we choose them.

3.1.1. Faster R-CNN with FPN

Faster R-CNN [31], as one of the most well-known two-stage detectors, is still widely used today. One contribution of [31] is Region Proposal Network (RPN), which is a sub-network and works to generate object proposals. RPN was proposed to replace generating proposals by selective search [42]. Differently, RPN predicts 3 pre-defined scales and 3 pre-defined aspect ratios at each anchor point, which is the center of each sliding window, yielding 9 anchor boxes at each anchor center. The purpose of RPN is to select the foreground from the input images, filtering the background. The fixed number of proposals (i.e., top N) that are generated by RPN will be fed into ROI pooling and the detection head for classification and Bbox offsets tuning purposes.

The lower-level feature maps have higher resolution, which can provide more accurate information in localization; whereas higher-level feature maps contain stronger semantic values. To leverage both location accuracy and semantic richness, multi-scale feature representation methods are commonly used. FPN [25] was proposed in 2017, which introduced a novel network block to achieve multi-scale feature representation. FPN proposed a feature map top-down pathway by conducting upsampling, and the feature maps from the bottom-up pathway (i.e., the feature maps generated during the feed-forward computation of ResNet) at the corresponding level are further merged by element-wise addition,

As demonstrated in Fig. 2(a). we adopted FPN in our experiment and the fused feature maps from four different levels are used as the RPN input. The network introduced in this subsection will be called Faster R-CNN + FPN in the rest of this paper. As for the backbone choices, we use both ResNet-50 [24] and HRNetV2-W32 [26] to extract the features, respectively, which helps to study the influence of backbones and better understand the performance of this classic two-stage detector.

3.1.2. RetinaNet

Unlike two-stage detectors, one-stage detectors need to predict the class of proposals directly, without any filtering steps. Therefore, one challenge that one-stage detectors are facing is the majority of candidate proposals are not ground truth target but the background. The imbalanced number of false examples will overwhelm the loss-guided training phase. To cope with the class imbalance problem, the authors of RetinaNet [33] proposed a novel loss function called Focal Loss. Focal Loss applied soft sampling concept with a α-balanced modulating factor $\alpha \times (1-p_t)^\gamma$ to the plain cross-entropy (CE) loss function (i.e., $-\log(p_t)$), where p_t is the possibility predicted by the model for the foreground target, otherwise it equals to 1 — the possible predicted by the model.

The modulating factor helps to control the influence of easy examples, both for positive and negative samples. For example, when the model predicts the easy example with $p_t = 0.9$, and suppose γ is set to 2, the loss contribution will be 100 times less than the baseline CE loss. In the experiment, the focal weight α was set to 0.25, and γ was set to 2.

RetinaNet is a one-stage detector; it combines ResNet and FPN [25] to extract richer features, as shown in Fig. 2(b). Taking advantage of the Focal Loss, RetinaNet achieves competitive accuracy and beats many two-stage detectors' accuracy on almost all subsets of MS COCO dataset [43].

3.1.3. GA-RetinaNet

Instead of generating anchors by sliding windows, the GA-RetinaNet [39] in Fig. 2(c) can learn and predict where the anchor is and the shape of the anchor box by applying the guided-anchor scheme. Guided-anchoring proposed a novel anchor-fusion pipeline, that is using the anchor-free method to predict the anchor points and corresponding anchor shapes, and finally generating the anchor boxes at the predicted locations.

Guided-anchoring is trained by the multi-task way, combining the branches of anchor locations and anchor shapes jointly, in addition to the basic classification branch and regression branch. The anchor location branch is modeled as a classification problem, and it is handled by Focal Loss, with a binary label map as the ground truth, where 1 indicates the correct location for the anchor center. The anchor shapes branch is treated as the regression problem, and the authors facilitated the intersection-over-union (IoU) overlap to evaluate the anchor shapes. To handle the various anchor shapes, the authors applied a 3×3 deformable convolutional layer to conduct the feature adaption.

The design of the guided-anchoring scheme is different from blindly generating dense anchor boxes at each anchor point, but is an alternative efficient anchoring scheme and can be simply adopted by many detectors. The proposed guided-anchoring can replace the RPN to generate proposals for two-stage detectors and can be adapted into the detection head for one-stage detectors.

3.1.4. ALFNet

RPN-based two-stage detectors greatly improve the accuracy, but they slow down the inference speed, making them not practical for real-time detection purposes. Differently, one-stage detectors are generally faster than two-stage detectors but their accuracy is not as competitive as two-stage detectors. In addition to Focal Loss, there are many other researchers working on improving the detection quality of one-stage detectors, such as ALFNet [40].

As demonstrated in Fig. 2(d), ALFNet improved the pedestrian detection accuracy by applying a series of cascaded Convolutional Predictor Blocks (CPB) at four different levels. Each CPB is regarded as a detection head, and the detection results are filtered by adopting higher IoU thresholds at later predictors. The anchor boxes are optimized with a strict threshold, resulting in more precise detection results.

Table 1
Detectors comparison.

Detectors	Year	# of stages	Anchor
Faster R-CNN	2015	Two-stages	Yes
RetinaNet	2017	One-stage	Yes
GA-RetinaNet	2019	One-stage	Yes
ALFNet	2018	One-stage	Yes
FCOS	2019	One-stage	No
CSP	2019	One-stage	No
BCNet	2020	One-stage	No

3.1.5. *FCOS*

Fully Convolutional One-Stage (FCOS) detector [41], namely, is a one-stage detector and the authors did not use any fully connected layers. As shown in Fig. 2(e), FCOS works to predict the target's centerness, integrating with a regressing branch to predict the 4-way distance to the Bbox border. For example, the ground truth Bbox with coordinates (x_0, y_0) and (x_1, y_1), the regression target at the coordinate (x, y) should be the 4D vector $(\hat{l}, \hat{t}, \hat{r}, \hat{b})$ where $\hat{l} = x - x_0$, $\hat{t} = y - y_0$, $\hat{r} = x_1 - x$, and $\hat{b} = y_1 - y$. The ground truth centerness is formulated as $\sqrt{\frac{min(\hat{l},\hat{r})}{max(\hat{l},\hat{r})} \times \frac{min(\hat{t},\hat{b})}{max(\hat{t},\hat{b})}}$.

The authors predicted the object centerness in a per-pixel prediction fashion, which greatly reduces the number of hyper-parameters and network outputs. In addition, the centerness also provides the semantic features of the object.

3.1.6. *CSP*

Representative anchor-free detectors such as CornerNet [44] took advantage of two corner keypoints to predict the target. In addition to two corner keypoints, CenterNet [45] predicted the center point of the corresponding target, and the success of CenterNet indicated the effectiveness of the center keypoint. By predicting the centerness of the pedestrian and the corresponding scales, the authors of CSP [15] designed this one-stage anchor-free detector, as demonstrated in Fig. 2(f). The Bboxes are not directly generated by predicting the top-right and bottom-left coordinates, but generated by transforming the predicted scale (i.e., height and width) at each predicted center point, and refined with the predicted offset.

3.1.7. *BCNet*

In addition to the centerness prediction used by CSP, Bi-Center Network (BCNet) [8] proposed to utilize the semantic features of pedestrians' visible parts. As presented in Fig. 2(g), BCNet fuses the semantic features of the centerness of the full body and the centerness of the visible part for each pedestrian, and predicts the height and offset for the corresponding pedestrian simultaneously. The detection head outputs the center keypoint 2D mask for the center of full body (CKFB) branch and the center of visible part (CKVB) branch, and the feature fusion of these two centerness masks is guided by two hyper-parameters α and β, and we adopted the same experimental setting with $\alpha = 1$ and $\beta = 0.5$.

3.1.8. *Summary*

We summarize key characters of the detection frameworks we use in Table 1. It can be found that more detectors are designed as one-stage detectors, especially after the design of Focal Loss. Moreover, many detectors have been designed as anchor-free pipelines, especially since 2019. These selected detectors in this paper can cover a variety of different categories.

3.2. *Data processing*

3.2.1. *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.

Taking the CityPersons dataset [17] as an example, the images are in the resolution of 1024 × 2048. All pedestrian instances are divided 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.

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 [22] and suggested by PyTorch.[1] We jitter brightness by 0.5, and contrast, saturation, and hue of input images remain unchanged.

Data augmentation is a popular manner to reduce overfitting by enlarging the diversity of data, 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 [17] and the images after pre-processing in Fig. 3 to better understand what we did in the data augmentation step. Fig. 3(d) was pre-processed by resizing to a smaller ratio and image pave, Fig. 3(e) was enlarged and cropped into the unified image resolution, and Fig. 3(f) was horizontally flipped and enlarged before image cropping.

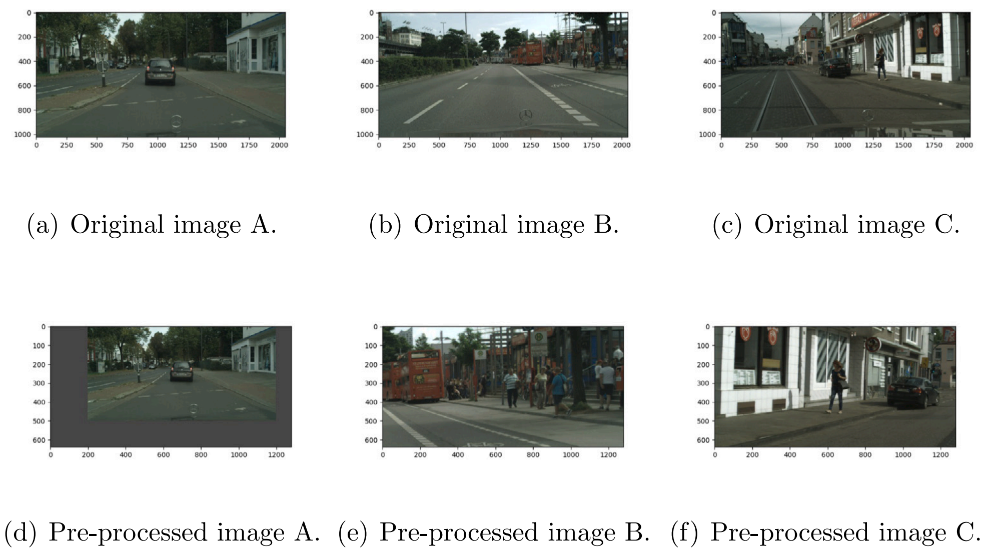
3.2.2. *Post-processing*

Post-processing is the last processing step to polish our prediction results. We will introduce the non-maximum suppression (NMS) working scheme in this section. NMS is used to eliminate the duplicate and unnecessary prediction results. NMS will be applied when testing the detector, but not during the training step. 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.

4. Experiment

In this Section, we will walk through experiment details. Before introducing the experimental settings, we will compare the datasets and explain why we adopt them. We evaluate the pedestrian detectors on both the CityPersons [17] dataset and the ETH [18] dataset for a comprehensive comparison.

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


(a) Original image A. (b) Original image B. (c) Original image C.

(d) Pre-processed image A. (e) Pre-processed image B. (f) Pre-processed image C.

Fig. 3. Images pre-processing visualization.

Table 2
Comparisons between the CityPersons dataset and the ETH dataset.

	ETH	CityPersons
# of images	2303	5000
# of persons	~ 14,000	35,016
Persons per image	6.1	7.0
Resolution	640 × 480	2048 × 1024
Train-val-test split (%)	20/0/80	60/10/30
# of seasons	1	3
# of cities	1	27
# of countries	1	3
Occlusion tags	×	√
# of pedestrian labels	1	5
Ignore region	×	√
Year	2007	2017

Table 3
Evaluation Setups of the CityPersons dataset [17].

(a) Setups with different visibility ratios.

	Reasonable (R)	Bare (B)	Partial (P)	Heavy (H)
Height (in pixels)	[50, +∞]	[50, +∞]	[50, +∞]	[50, +∞]
Visibility ratio	[0.65, 1]	[0.9, 1]	[0.65, 0.9]	[0, 0.65]

(b) Setups with different heights.

	Small (S)	Medium (M)	Large (L)
Height (in pixels)	[50, 75]	[75, 100]	[100, +∞]
Visibility ratio	[0.65, 1]	[0.65, 1]	[0.65, 1]

Table 4
Experimental settings of detectors' training.

	Backbone	Image resolution	Batch size
Faster R-CNN+FPN	HRNet	256 × 512	1
Faster R-CNN+FPN	ResNet-50	608 × 1216	1
Faster R-CNN*+FPN	ResNet-50	256 × 512	1
RetinaNet	ResNet-50	456 × 912	2
GA-RetinaNet	ResNet-50	256 × 512	2
ALFNet	ResNet-50	640 × 1280	2
FCOS	ResNet-50	608 × 1216	1
CSP	ResNet-50	640 × 1280	2
BCNet	ResNet-50	640 × 1280	2

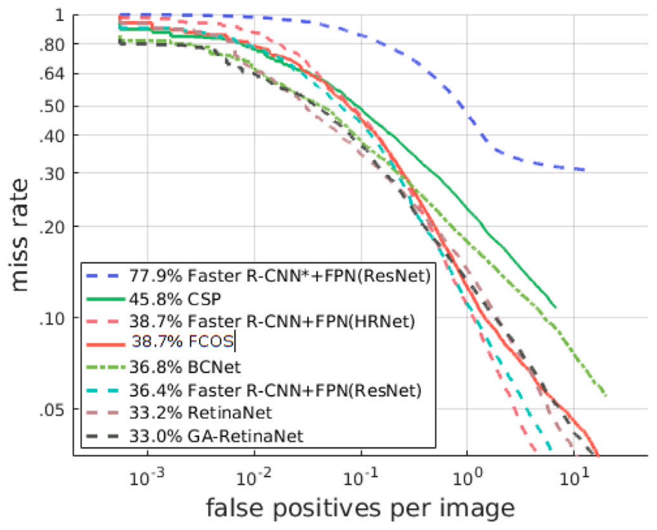


Fig. 4. Evaluation on the ETH dataset.

4.1. Datasets for evaluation

We choose the CityPersons dataset and the ETH dataset to conduct our experiments, because they are very different in many aspects as displayed in Table 2.

The ETH dataset [18] 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. The CityPersons Dataset [17] 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 a Bbox of the visible part for a pedestrian if he/she is occluded. The authors labeled 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 CityPersons dataset offers semantic information. The detailed configurations of each setup are given in Table 3.

The ETH dataset and the CityPersons dataset are different in many aspects, including the density of people on the image, the resolution of

Table 5
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	12.82	8.27	12.75	51.14	19.23	4.78	7.27
BCNet	**9.82**	**5.83**	**9.23**	53.34	**12.98**	**3.30**	**6.10**

[1] The results in boldface indicate the best on the corresponding subsets.
[2] The resolution of training images of Faster R-CNN*+FPN (ResNet) is 256 × 512.

Table 6
We cite the experimental results on the CityPersons validation set from original authors. The results in boldface indicate the best performance.

	GPU	Reasonable	Bare	Partial	Heavy	Small	Medium	Large
Faster R-CNN [17]	–	15.4	–	–	64.83	25.6	7.2	7.9
Faster R-CNN + Semantic [17]	–	14.8	–	–	–	22.6	6.7	8.0
Faster R-CNN + ATT-self [10]	–	20.93	–	–	58.33	–	–	–
ALFNet [40]	2 GPUs (1080Ti)	12.0	8.4	11.4	51.9	19.0	5.7	6.6
CSP [15]	4 GPUs (1080Ti)	11.0	7.3	10.4	**49.3**	16.0	3.7	6.5
BCNet	1 GPU (1080Ti)	**9.8**	**5.8**	**9.2**	53.3	**13.0**	**3.3**	**6.1**

the image, and the diversity of street views. We adopt the CityPersons dataset as the main dataset, we will train the detectors on the training set and test them on the validation set. In addition, we believe using these two very different datasets to conduct our experiments can test the generalization ability of detectors by training them on the CityPersons dataset and applying to the ETH dataset without any fine-tuning.

4.2. *Experimental settings*

In the experiment, all the detection models are trained on a single Nvidia GeForce GTX 1080 Ti GPU, with 11 GB GPU memory. We adopt the official work provided by authors of [15] and the MMDetection toolbox [46] to build up the experimental models and configurations. 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 4. To fully utilize our GPU's computing ability and obtain the best accuracy, we set the batch size to 2 if the GPU memory is sufficient. We have to set the batch size to 1 for the detector Faster R-CNN + FPN and FCOS. We adjust the learning rate and the number of epochs by observing the convergence of the training loss. We reduce each detector's training images to a different resolution to make it compatible with GPU memory. In order to investigate the impact of training image resolution, we reduce the training image resolution of Faster R-CNN* + FPN to 256 × 512, as shown in Table 4.

The unified evaluation metric is MR^{-2} (the lower, the better), which is the mean value of nine derived miss rates with the corresponding FPPIs (false positive per image) evenly located in $[10^{-2}, 10^{0}]$ within the log-space.

4.3. *Evaluation on the CityPersons dataset*

After training detectors with the settings shown in Table 4, we test detectors on the CityPersons validation set with original image resolution 1024 × 2048. The evaluation results on the CityPersons dataset are shown in Table 5, and we plot the Bboxes on the same image from the CityPersons dataset in Fig. 5 for visualization.

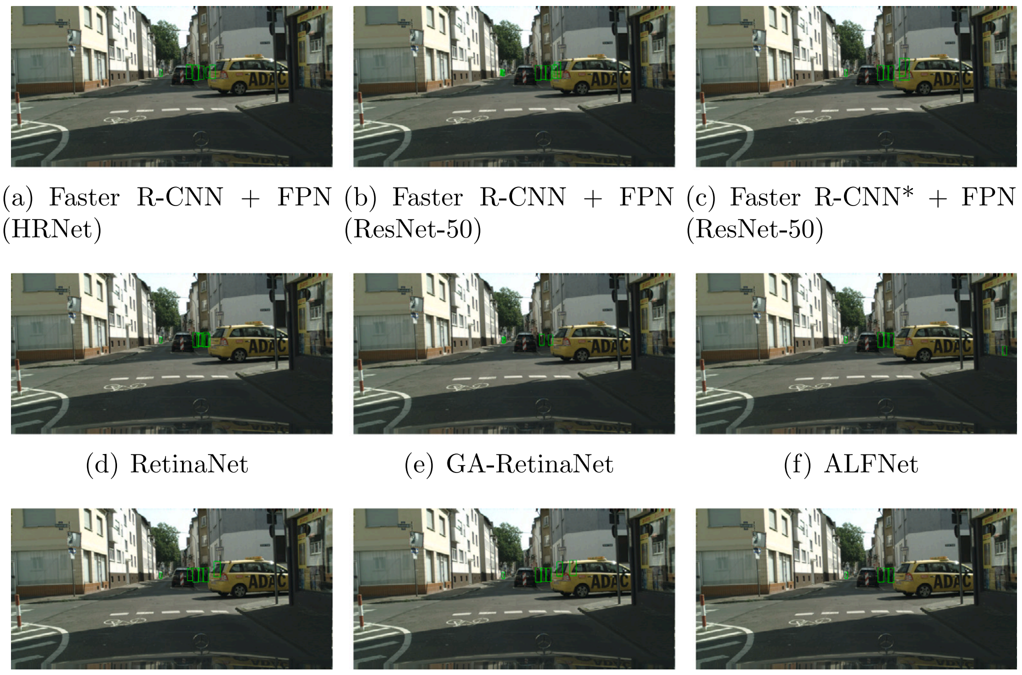
From the experimental results, we can find FPN is effective to benefit the accuracy of detectors: RetinaNet was proposed and declared to beat the performance of Faster R-CNN in [33]. However, after applying FPN to Faster R-CNN, its accuracy is dramatically enhanced. The image resolution also has an impact on the detector's accuracy. Faster R-CNN + FPN with ResNet-50 is trained with a larger resolution comparing with the Faster R-CNN + FPN with HRNet that is trained with the image resolution of 256 × 512, and they achieve pretty similar performance, especially on Reasonable, Bare, and Partial setups. When Faster R-CNN + FPN with ResNet-50 and HRNet detectors are both trained with the same resolution, it is found that the HRNet backbone contributes to the accuracy. Therefore, it is evident that the backbone of detectors has a non-ignored effect on the detector's accuracy.

We listed the experimental results from the authors and the GPU information in Table 6, if applicable. The authors of [17] did not provide clear information about the hardware used in the Faster R-CNN experiment, which prevented us from making comparisons. We found by using four GPUs, ALFNet obtained a MR^{-2} of 12.0% on the Reasonable setup, which is 2.87% better than the result obtained on one GPU. The performance of ALFNet on Small and Medium setups dramatically drops when it is trained with 1 GPU. However, the performance difference of CSP trained with 1 GPU and 4 GPUs is not very obvious on most setups, which could illustrate the effectiveness of CSP's network design.

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

4.4. *Evaluation results on the ETH dataset*

The ETH dataset [18] has 1804 images in total, and the image resolution is 480 × 640, which is much smaller compared with the image resolution of the CityPersons dataset. In order to compare the generalization ability of each detector, we directly apply the trained detectors to the ETH dataset without any other training or fine-tuning. We plot the evaluation results on Fig. 4. It can be observed that GA-RetinaNet achieves the best performance, and BCNet achieves a moderate performance. The performance of CSP drops significantly, compared with BCNet. Faster R-CNN* + FPN with ResNet-50 obtains the worst performance among these detectors, which could further underline the impact of training image resolution. The generalization ability of pedestrian detectors is critical when they are applied to autonomous driving system, enabling them predict correct results on unseen data. Currently, most pedestrian detectors do not have strong generalization ability, and this is one huge challenge in the pedestrian detection area.



(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) CSP (i) BCNet

Fig. 5. Visualization results on the CityPersons dataset.

5. Conclusion

In this comparative study paper, we stated our motivation is to compare different detectors under the same computational environment. We selected various representative CNN-based pedestrian detectors that cover different perspectives, and we introduced and compared them in detail. From the evaluation results, we found that the feature extraction capability of backbones directly impacts the detector's accuracy; a powerful backbone may contribute to yield higher accuracy. Training CNN models with reduced image resolutions would slightly decrease accuracy, even if the images are reduced to the same ratio. The trade-off between hyper-parameter settings and network architectures determines the complexity of the model, and the complexity of the model may be limited by GPUs (especially on-board GPUs). As we demonstrated in the evaluation results on the ETH dataset, the generalization ability is one open challenge in the pedestrian detection area. In the future, we will work on improving the generalization ability of pedestrian detectors by increasing the dataset diversity.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R.I. Meneguette, L.H.V. Nakamura, A flow control policy based on the class of applications of the vehicular networks, in: Proc. ACM MobiWac, 2017, pp. 137–144.

[2] P.B. Bautista, L. Urquiza-Aguiar, M. Aguilar-Igartua, Evaluation of dynamic route planning impact on vehicular communications with SUMO, in: Proc. ACM MSWiM, 2020, pp. 27–35.

[3] R.W.L. Coutinho, A. Boukerche, X. Yu, Information-centric strategies for content delivery in intelligent vehicular networks, in: Proc. ACM MSWiM, 2018, pp. 21–26.

[4] A. Nahar, D. Das, S.K. Das, OBQR: Orientation-based source QoS routing in VANETs, in: Proc. ACM MSWiM, 2020, pp. 199–206.

[5] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: A benchmark, in: Proc. IEEE CVPR, 2009, pp. 304–311.

[6] H. Huang, S. Lin, WiDet: Wi-Fi based device-free passive person detection with deep convolutional neural networks, in: Proc. ACM MSWiM, 2018, pp. 53–60.

[7] P. Sun, A. Boukerche, Challenges and potential solutions for designing a practical pedestrian detection framework for supporting autonomous driving, in: Proc. ACM MobiWac, 2020, pp. 75–82.

[8] M. Sha, A. Boukerche, Semantic fusion-based pedestrian detection for supporting autonomous vehicles, in: Proc. IEEE ISCC, 2020, pp. 1–6.

[9] A. Boukerche, M. Sha, Design guidelines on deep learning–based pedestrian detection methods for supporting autonomous vehicles, ACM Comput. Surv. 54 (6) (2021).

[10] S. Zhang, J. Yang, B. Schiele, Occluded pedestrian detection through guided attention in CNNs, in: Proc. IEEE CVPR, 2018, pp. 6995–7003.

[11] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, C. Shen, Repulsion loss: Detecting pedestrians in a crowd, in: Proc. IEEE CVPR, 2018, pp. 7774–7783.

[12] C. Zhou, J. Yuan, Bi-box regression for pedestrian detection and occlusion estimation, in: Proc. ECCV, 2018, pp. 135–151.

[13] G. Brazil, X. Yin, X. Liu, Illuminating pedestrians via simultaneous detection & segmentation, in: Proc. IEEE ICCV, 2017, pp. 4950–4959.

[14] M. Sha, A. Boukerche, A novel visibility semantic feature-aided pedestrian detection scheme for autonomous vehicles, Comput. Commun. 179 (2021) 50–61.

[15] W. Liu, S. Liao, W. Ren, W. Hu, Y. Yu, High-level semantic feature detection: A new perspective for pedestrian detection, in: Proc. IEEE CVPR, 2019, pp. 5187–5196.

[16] J. Zhang, L. Lin, Y. Chen, Y. Hu, S.C.H. Hoi, J. Zhu, CSID: Center, scale, identity and density-aware pedestrian detection in a crowd, 2019, Available: https://arxiv.org/abs/1910.09188, Online: (Accessed November 2019).

[17] S. Zhang, R. Benenson, B. Schiele, CityPersons: A diverse dataset for pedestrian detection, in: Proc. IEEE CVPR, 2017, pp. 4457–4465.

[18] A. Ess, B. Leibe, L. Van Gool, Depth and appearance for mobile scene analysis, in: Proc. IEEE ICCV, 2007, pp. 1–8.

[19] C.P. Papageorgiou, M. Oren, T. Poggio, A general framework for object detection, in: Proc. IEEE ICCV, 1998, pp. 555–562.

[20] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[21] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proc. NIPS, 2012, pp. 1097–1105.

[22] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, Li Fei-Fei, ImageNet: A large-scale hierarchical image database, in: Proc. IEEE CVPR, 2009, pp. 248–255.

[23] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2015, Available: https://arxiv.org/abs/1409.1556v6, Online: (Accessed November 2019).

[24] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE CVPR, 2016, pp. 770–778.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proc. IEEE CVPR, 2017, pp. 2117–2125.

[26] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al., Deep high-resolution representation learning for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2020).

[27] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proc. IEEE CVPR, 2005, pp. 886–893.

[28] C.P. Papageorgiou, M. Oren, T. Poggio, A general framework for object detection, in: Proc. IEEE ICCV, 1998, pp. 555–562.

[29] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proc. IEEE CVPR, 2014, pp. 580–587.

[30] R. Girshick, Fast R-CNN, in: Proc. IEEE CVPR, 2015, pp. 1440–1448.

[31] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Proc. NIPS, 2015, pp. 91–99.

[32] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proc. IEEE CVPR, 2016, pp. 779–788.

[33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: Proc. IEEE ICCV, 2017, pp. 2980–2988.

[34] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proc. IEEE ICCV, 2017, pp. 2961–2969.

[35] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot multibox detector, in: Proc. ECCV, 2016, pp. 21–37.

[36] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, 2016, Available: http://arxiv.org/abs/1612.08242, Online: (Accessed October 2020).

[37] L. Tychsen-Smith, L. Petersson, DeNet: Scalable real-time object detection with directed sparse sampling, in: Proc. IEEE ICCV, 2017, pp. 428–436.

[38] T. Song, L. Sun, D. Xie, H. Sun, S. Pu, Small-scale pedestrian detection based on topological line localization and temporal feature aggregation, in: Proc. ECCV, 2018, pp. 536–551.

[39] J. Wang, K. Chen, S. Yang, C.C. Loy, D. Lin, Region proposal by guided anchoring, in: Proc. IEEE CVPR, 2019, pp. 2960–2969.

[40] W. Liu, S. Liao, W. Hu, X. Liang, X. Chen, Learning efficient single-stage pedestrian detectors by asymptotic localization fitting, in: Proc. ECCV, 2018, pp. 618–634.

[41] Z. Tian, C. Shen, H. Chen, T. He, FCOS: Fully convolutional one-stage object detection, in: Proc. IEEE ICCV, 2019, pp. 9627–9636.

[42] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.

[43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common objects in context, in: Proc. ECCV, 2014, pp. 740–755.

[44] H. Law, J. Deng, CornerNet: Detecting objects as paired keypoints, in: Proc. ECCV, 2018, pp. 734–750.

[45] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, Centernet: Keypoint triplets for object detection, in: Proc. IEEE ICCV, 2019, pp. 6569–6578.

[46] K. Chen, et al., MMDetection: Open MMLab detection toolbox and benchmark, 2019, Available: https://arxiv.org/pdf/1906.07155.pdf, Online: (Accessed October 2020).

Mingzhi Sha is currently a master student of computer science in Paradise Research Laboratory at the University of Ottawa. She received the B.Eng. degree in computer science from Sichuan University, Chengdu, China, in 2017. Her current research topic is using deep learning-based methods to improve the performance of pedestrian detection, which can be applied to autonomous vehicles.

Azzedine Boukerche (FIEEE, FCAE, FEIC, FAAAs) is a distinguished Professor and Senior Canada Research Chair Tier-1. He serves as Editor of several Journals including Elsevier Ad Hoc Network and IEEE Transactions on Cloud Computing. His Research of Interests Vehicular network, Sensor network and Mobile Systems.