Challenges of Designing Computer Vision-based Pedestrian Detector for Supporting Autonomous Driving

Peng Sun, Azzedine Boukerche
PARADISE Research Laboratory, EECS, University of Ottawa, Canada
Emails: psun044@uottawa.ca, boukerch@site.uottawa.ca

***Abstract*—In recent years, aiming to improve deriving safety and supporting autonomous driving, pedestrian detection has attracted considerable attention from both industry and academic. Moreover, by taking advantage of the powerful computational capacity of GPU and high-level feature learning ability of the deep convolutional neural network, tremendous image/video-based pedestrian detection methods have been proposed. However, most of the existing approaches are designed relying on the computer vision-based target detection techniques. Accordingly, the evaluation criteria they consider in the design are often from the computer vision research field. Therefore, these existing methods tend to focus on the improvement of accuracy and ignore some of the special requirements that need to be considered in the field of autonomous driving. In this paper, we will analyze and summarize the features of the state-of-the-art pedestrian detection methods in detail. Then, by considering the practical application scenarios of autonomous driving techniques, we further discuss the open challenges of designing a practical pedestrian detection method for supporting autonomous deriving.**

***Index Terms*—Autonomous driving; artificial intelligence; traffic condition sensing; computer vision; pedestrian detection.**

I. Introduction

In modern society, the transportation system is the essential infrastructure to maintain the normal functioning of society. Moreover, as the demand for transportation logistics increases, in order to maintain the efficiency of the transportation system, the requirements for traffic safety are getting higher and higher. However, the growing number of transportation system participants (i.e., number of vehicles, number of roads and bridges, etc.) has led to the increasing complexity of transportation systems, how to improve road traffic safety has become a serious issue that has to be solved.

To overcome this issue, relying on the maturing traffic condition sensing techniques, Level 1∼3 driving assistant systems/autonomous driving techniques [1], e.g., lane-keeping/departure assistant [2], driver drowsiness detection [3], dynamic routing plan [4], etc., are gradually beginning to be applied on newly manufactured vehicles for improving the road safety. Typically, the objective detection methods are essential for supporting these techniques, e.g., the road marking detection for lane keeping/departure assistant [5], pedestrian detection for improving the safety of transportation system participants [6], animal detection for avoiding Animal-Vehicle Collision [7], etc., and most of these methods are designed relying on the video or image captured by the on-board camera. Hence, by taking advantage of the powerful computational capacity of GPU, many machine learning-type approaches designed for computer vision-related tasks are normally adopted in the autonomous driving area, e.g., Convolutional Neural Network (CNN) [8], etc. However, most of the existing approaches are designed relying on the image/video-based target detection techniques. Accordingly, the criteria of these existing approaches considered in the design are often directly from the computer vision research field. Therefore, these existing methods tend to focus on the improvement of accuracy and ignores some of the special requirements that need to be considered in the field of autonomous driving.

Therefore, in this article, we will focus on the development of pedestrian detection technology. By analyzing the technical characteristics of existing pedestrian detection methods and the actual needs of autonomous driving, we will clearly point out the open technical challenges are still not effectively solved in this research field. More precisely, we will provide open challenges and potential solutions of designing a practical pedestrian detection method for supporting autonomous driving.

The remainder of this article is organized as follows: In Section II, we will discuss some preliminary concepts and a road map of the development of pedestrian detection techniques. Then, in Section III, we literature review and discuss the characteristics of existing computer vision-based pedestrian detection method. Accordingly, in Section IV, we point out the open challenges of designing a practical pedestrian detection method for supporting autonomous driving in detail. Finally, Section V concludes the paper.

II. Preliminaries and a road map of pedestrian detection

Achieving effective pedestrian detection is a vital issue for the application of autonomous driving. Because as an important participant in the transportation system, pedestrians

This work is partially supported by NSERC-SPG, NSERC-DISCOVERY, Canada Research Chairs Program, NSERC-CREATE TRANSIT Funds.

2155-6814/19/$31.00 ©2019 IEEE
DOI 10.1109/MASS.2019.00013

are often more vulnerable to traffic accidents than occupants in the vehicle. By exploiting the effective pedestrian detection technology, the autonomous driving vehicle can predict the possible collision events to a certain extent by detecting pedestrians and analyzing their movement trajectories in advance, and then actively pre-adjust the driving state of the vehicle itself to avoid possible collisions. Accordingly, considerable research efforts have been devoted to the related research fields.

In 2005, the famous Histograms of Oriented Gradients (HOG) descriptor was introduced by Dalal and Triggs in [9] for implementing the human detection task relying on a linear SVM classifier. Typically, the HOG can be considered an *edge detection-based* method. By dividing the input fixed-size image into multiple non-overlapping *cells*, a 9-bin histogram will be used to represent the feature contained with a cell (derived by two 1-D masks), where each column within the histogram represents the sum of the contributions of pixels within the cell to this specific angle. Moreover, to eliminate the effect of the illumination, the L2-norm-based normalization is typically adopted for each block of 2×2 or 3×3 cells of 6×6 or 8×8 pixels. In [9], the standard setting is the 2×2 block of 8×8-pixel cells. Therefore, for each block containing 16×16 pixels, the features of the block is described by a 36×1 vector concatenated by four 9-bin histograms. Finally, a complete HOG feature vector can be formed by concatenating all 36×1 block description vectors together. The SVM will be trained by the derived HOG feature vectors. Due to its high detection accuracy and relatively simple computational complexity, HOG is widely regarded as a benchmark and adopted by many existing applications in the field of pedestrian detection for a period of time.

For instance, Deformable Part-based Model (DPM) introduced by [10] in 2008 can be considered as an improved version of HOG. Unlike the original HOG identifying the object based on single HOG feature vector of the whole image, DPM recognizes the target within an image (represented an image pyramid) by a scanning window approach. Briefly, by exploiting the HOG features of different parts of the target, the target within the input image is detected in a divide-and-conquer manner. Then, the final score of the detection window can be calculated by considering the scores of root filter and parts filters and the cost of the deformation. Accordingly, comparing the conventional HOG, DPM achieved an important feature, i.e., deformation invariance.

From 2012, by exploiting the high computational capacity of GPU and the high-level feature learning ability of the CNN, Krizhevsky et al. in [8] introduced the well-known ImageNet/AlexNet model for solving the target detection problem based on the high-resolution images. Since then, the design of pedestrian detection by exploiting CNN has gradually become a trend. Typically, as shown in Fig. 1, CNN can be considered a two-part Neural Network. The first part is the convolution (Conv) layers. This part can be considered as the feature learning parts of CNN. Through this part, the features contained in the input image can be extracted by the convolution kernel/filters. Then the derived features will be flattened into a column vector and then forwarded to the second part of CNN, i.e., the fully-connected (FC) layers. This structure can be considered a conventional NN, in which, the non-linear patterns contained within the input features are learned by adjusting the weights between neurons and the thresholds of the activation functions. Finally, the derived values are fed to a classifier (e.g., the widely-used Softmax) to generate the final classification results. Currently, most of existing computer vision-based pedestrian detection methods are designed by reforming this general structure based on additional layers (e.g., ResNet [11], etc.) or components (e.g., RCNN [12], SPP-Net [13], etc.). In the following section, we will discuss the existing approaches in detail.

III. Theoretical Comparison of the State-of-the-Art

In this section, we do a survey of the existing computer vision-based pedestrian detection methods, and theoretically summarize their advantages and disadvantages.

As mentioned before, the pedestrian detection can be considered as an application of computer vision-based target detection method. The existing approaches are mainly designed based on some backbone networks by adding some specific components for identifying the pedestrian features from the input images. Therefore, before going to the specific pedestrian detection methods, we first discuss some widely used generic target detection approaches which are commonly adopted as backbone/core detecting unit for implementing pedestrian detection task.

A. Commonly adopted generic target detectors

Typically, the primary goal of the target detection method is to provide two essential pieces of information, i.e., the location information of the target and the nature/type of the target. In other words, the target detection is to answer two fundamental questions: "*Where is the target*?" and "*What is the target*?" Correspondingly, based on the process of solving these two sub-problems, the existing computer vision-based target/objective detection method can be roughly categorized into two types, i.e., the two-stage method and the single-stage method.

1) Two-stage detectors: Typically, the second-order detector solves the two fundamental problems we mentioned earlier in a step-by-step manner. The detector first derives the potential bounding box (Bbox) [or region of interest (RoI)] on the feature maps extracted from the input image. In this step, the detector derives the rough location information of all potential targets. Then, relying on the features within the Bbox/RoI, the Bbox regressor and the selected classification method (e.g., linear SVM in RCNN [12]) can further refine the Bbox and determine the class to which the target belongs, respectively. Here, we list some of the most widely used two-stage detectors.

- RCNN [12]: For each input image, 2K RoIs are determined by the selected proposal generation method (i.e.,

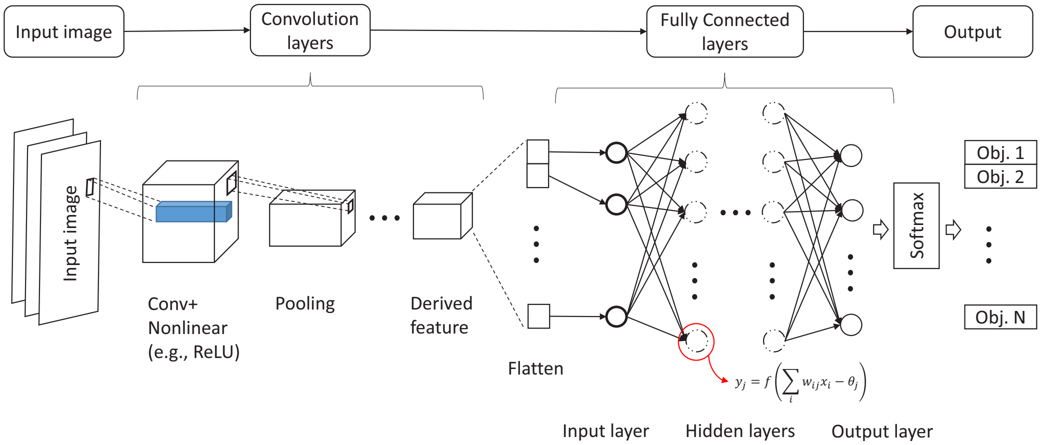

Fig. 1. An illustration of the general structure of CNN for implementing image-based target detection

selective search [14]). Every selected RoI within the input image will be wrapped to a fixed-size (i.e., 227×227 pixel size) and then fed into the CNN. Subsequently, the feature vectors extracted from each RoI by CNN will be forwarded to SVMs trained for different target classes for further implementing the classification task. By training on the well-known AlexNet [8] framework, compared with its competitors at the same time, RCNN achieved significant improvement regarding detection accuracy, i.e., improved the mean Average Precision (mAP) on PASCAL VOC 2010 from 33.4% (achieved by DPM [10]) to 53.7%. However, since the components of RCNN are trained in Ad hoc manner (i.e., every classifier and Bbox regressor have to trained separately). The training cost of RCNN in terms of time consumption is considerable, i.e., 84-hour. Also, the detection speed is slow, which is 47s/image with VGG16 [15] configuration.

- SPP-Net [13]: For the slow detection speed of RCNN, He et al. pointed out that the main reason is that, in order to adapt to the parameters of the FC layer in a given CNN, the RCNN must perform a large amount of preprocessing on the input image (i.e., wrapping 2K RoIs to a fixed-size) before feeding it to the network. Therefore, He et al. in [13] adjusted the processing flow of the input image and designed a new Spatial Pyramid Pooling (SPP) layer to generate feature vectors with fixed length to provide input for the FC layer. More precisely, without a large amount of image resizing, SPP-Net directly inputs the image into the Conv-layers for feature extraction. Then, through the newly designed SPP layer, a fixed-size feature vector that meets the requirements of the FC layer is generated by pooling the necessary information from the derived feature maps at multiple scales. Consequently, compared to RCNN, SPP-Net increases target recognition speed by 64 times and 102 times on PASCAL VOC 2007 by trained on ZF-Net [16] and ImageNet/AlexNet [8], respectively, while maintaining detection accuracy. However, due to the Pyramid Pooling feature, SPP-Net cannot be trained in the End-to-End manner but only be trained separately, i.e., the higher-layers within its system, and Conv-layers below the SPP layer. This also limits the further improvement of SPP performance regarding detection accuracy. Also, the training cost of SPP-Net is still high, i.e., 25-hour.
- Fast RCNN (FRCN) [17]: Aiming to overcome the drawbacks of RCNN and SPP-Net, Girshick introduced FRCN in [17]. Similar to SPP-Net, FRCN inputs the image directly to Conv-layers to derive a feature map for reducing the processing time. Meanwhile, to ensure that the entire network can be trained, FRCN projects the RoIs selected by the selected proposal method to the feature map extracted from the input image. Then, relying on the so-called RoI pooling layer, the desired size of the feature vector is pooled from the RoI projections on the feature map and fed in the FC layers. Moreover, unlike the RCNN training the SVM and Bbox regressors in separate stages, FRCN jointly fine-tunes/trains the Softmax classifier and Bbox regressor in a single step. Consequently, FRCN dramatically reduces the target detection speed and achieves higher detection accuracy. For PASCAL VOC 2012, based on the VGG16 configuration, FRCN achieves 213 times and 10 times faster detection speed than RCNN and SPP-Net, respectively. Meanwhile, the mAP is improve from 58.5% achieved by RCNN to 70.0% on PASCAL VOC 2007. Also, FRCN costs 9 times and 3 times less training time than RCNN and SPP-Net, respectively.
- Faster RCNN [18]: To further improve the performance of FRCN, Faster-RCNN was proposed by Ren et al. in [18]. To put it simply, Faster RCNN tries to improve the overall efficiency of FRCN by further integrating the proposal generation method into the network for improving the proposal generation efficiency (i.e., reducing the

proposal generation speed). To achieve this objective, by exploiting the deep convolutional network, the authors of Faster RCNN designed the *Region Proposal Networks* (RPN) to directly generate RoIs with corresponding objectiveness scores without using additional proposal methods. Generally, relying on the sliding window technique, a pyramid of k anchors with varies multiple scales and aspect ratios are predicted and centered at each sliding window. Then, relying on the features extracted from the input image, the classification and regression process of Bboxes can take the derived anchors as references. Further, by sharing the Conv-layers, the RPN is combined with FRCN to form a unified network. At this point, all the components required for image-based target detection are integrated into a unified network. Consequently, the efficiency of the target detection process is improved. As demonstrated by the authors, Faster RCNN (RPN+FRCN) achieved 5 frames per second (FPS) and 17 FPS image processing rate based on VGG16 and ZF-NET structure, respectively. Moreover, comparing with FRCN, the achieved detection accuracy is also slightly improved from 70.0% to 73.2% on PASCAL VOC 2007.
- Feature Pyramid Networks (FPN) [19]: Technically, FPN is not a "real" detector but an add-on feature extractor for a detector. Since the authors of FPN [19] combined it with Faster RCNN to form an actual FPN+Faster RCNN detector to implement the pedestrian detection task, we theoretically considered it as a two-stage detector in this paper. Briefly, by leveraging the inherent pyramidal feature hierarchy of Conv-layers, FPN builds a feature pyramid following the feed-forward computation procedure of Conv-layers. Within this derived feature pyramid, as the level increases, the resolution of the feature map reduces while semantic level increasing. Then, according to the top-down direction, through the upsampling way to provide more accurate spatial information contained by lower-level feature map for higher-level feature maps containing better semantic information but lacking spatial information, FPN can derive more precise feature maps. Accordingly, the FPN+Faster RCNN detector obtains the ability to detect targets with different scales. Meanwhile, as demonstrated by the authors, the FPN+Faster RCNN detector also achieved 5 FPS image processing rate based on VGG16 structure.

2) One-stage detectors: Through the above discussion of two-stage detectors, we can clearly see that the relatively slow detection speed is the main factor that restricts its application in the scene where the timeliness of detection is required, which is more critical in pedestrian detection for supporting autonomous driving. As pointed in [20], the main reason for the slow detection speed of two-stage detectors is that the target detecting in two-stage detectors is realized by two consecutive processes, i.e., positioning and recognition processes, and each process produces a corresponding delay, thus reducing the overall detecting speed of the detector. Therefore, to improve the detecting speed, the most straightforward method is to implement positioning and recognition processes simultaneously. Following this logic, many one-stage detectors have been proposed. Here, we discuss some well-known one-stage detectors in detail.

- YOLO [20]: YOLO (You Only Look Once) proposed in 2015 is a pioneer of one-stage detectors. In this work, instead of implementing localization and recognition/classification in two separated stage, by applying a single neural network, both Bboxes and corresponding class probabilities are predicted in a single stage. More precisely, YOLO process the input image as a $S \times S$ grids. For each grid, B Bboxes with different scales and corresponding class scores (i.e., the probability that the object contained by the Bbox belongs to a specific target class). Since, the Bbox may not be centered at the grid, it has to use four parameters (x, y, w, h) to denote its spatial information, where (x, y) denotes the center of the predicted Bbox and (w, h) represents the width and height of the Bbox relatively to the input image. While, the class probability is derived based on the features extracted by the Conv-layers[1]. Consequently, YOLO achieved 45 FPS image processing speed and Fast YOLO even achieved 155 FPS processing rate. However, due to the relatively higher localization error, compared with Faster RCNN, the detection accuracy of YOLO (with 45 FPS) dropped from 73.2% to 63.4% mAP and from 70.4% to 57.9% mAP on PASCAL VOC 2007 and VOC 2012, respectively. To overcome this drawback, YOLOv2 & YOLO9000 [22] and YOLOv3 [23] had been proposed by adopting more accurate Bbox prediction method and more accurate feature extractor (e.g., Darknet-19 and Darknet-53 used in YOLOv2 and YOLOv3, respectively).
- SSD [24]: Single Shot Detector (SSD) is another famous one-stage detector. Unlike YOLO generating Bboxes and their corresponding class scores only based on the final feature maps learned by Conv-layers, SSD exploits the inherent pyramidal feature hierarchy of Conv-layers. From each selected several feature layers with different resolution, SSD predicts a fixed set of default boxes (similar to the anchors in Faster RCNN) with corresponding class scores and shape related parameters (offsets). Then, by taking these default boxes with different resolutions as references, a non-maximum suppression step is adopted to produces final class scores and adjust the box to fit the object size. Relying on this multi-resolution detection technique, SSD obtains stronger ability to recognize target with varies scales than YOLO. As demonstrated by the authors, by training on VGG16 structure, SSD achieved 74.3% mAP on PASCAL VOC 2007 at 59 FPS on an Nvidia Titan X GPU, which outperform the 63.4% mAP and 45 FPS speed of YOLO.

[1] As mentioned by the authors, the network structure of YOLO is inspired by GoogLeNet [21].

- RetinaNet [25]: To further reduce the localization error and improve the ability to detect dense objectives, Lin et al. in [25] designed a new one-stage detector called RetinaNet. By taking FPN as the backbone feature extractor, for each layer of feature maps, the authors add two parallel convolutional subnets to implement both object classification and Bbox regression tasks. Meanwhile, by training the network with newly designed *focal loss*, RetinaNet achieved comparable accuracy to two-stage detectors while maintaining high detection rates. The focal loss is defined as follows,

  $$FL(p_t) = -(1-p_t)^{\gamma}\log(p_t), \tag{1}$$

  where p_t is truth estimated class probability and $\gamma \geq 0$ is a tunable focusing parameter for balancing the cross entropy.
- CornerNet [26]: CornerNet is a recently published one-stage detector. In this work, the authors focused on improving the training efficient of the detectors. As pointed by the authors, for many of existing detectors, the target localization is achieved by selecting the anchor box best fitting the target as the Bbox. Since the final Bbox is only selected from a tiny number of positive examples out of all generated anchors, in theory, a large number of extra hyperparameters used to define anchors are also need to be tuned during training, and these enormous amounts of hyperparameters severely impact system training efficiency. Therefore, without using parameters to denote anchors' sizes and aspect ratios, CornerNet defines the anchor box of potential targets by a pair of keypoints, i.e., top-left corner and bottom-right corner. Accordingly, the parameters used to define the same amount of anchors dramatically reduced from $O(w^2h^2)$ to $O(wh)$.

3) Other useful add-ons: Besides the two- and one-stage detectors discussed above, there are many other approaches designed for improving the detection accuracy. For instance, He et al. in [11] pointed out that, exploiting the enriched semantic information contained in the high-level feature is a key method to improve the detection accuracy. Hence, a sufficient network depth is of crucial importance. However, simply adding stacked layers to a network cannot directly lead to a higher detecting accuracy but increasing the training error, which in turn reduces the test error of the system. This phenomenon is called degradation. To cope with this degradation issue, following the hypothesis, adding identity mapping layers to copy features generated by shallow networks will not increase the training error, a deep residual learning network (ResNet) was proposed in [11]. In this work, unlike other related work directly designing the desired mapping function $\mathcal{H}(\mathcal{X})$, ResNet designed the mapping function as a residual function $\mathcal{F}(\mathcal{X}) := \mathcal{H}(\mathcal{X}) - \mathcal{X}$. The corresponding output of the ResNet layer is $\mathcal{F}(\mathcal{X})+\mathcal{X}$, where $\mathcal{X}$ is the feature learned by shallow networks. Accordingly, during the training process, by taking advantage of the known reference $\mathcal{X}$, optimization objective is converted to the minimizing $\mathcal{F}(\mathcal{X})$. By taking the identity mapping as an example, deriving solution of $\mathcal{F}(\mathcal{X}) = 0$ is much easier than optimizing the non-referenced function $\mathcal{H}(\mathcal{X})$. Correspondingly, ResNet provides an effective way to increase network depth without significantly increasing training complexity. Consequently, ResNet became one of the most widely adopted add-ons to increase the depth of a selected network.

B. The pedestrian detectors

As mentioned previously, pedestrian detection can be considered a specific application scenario of target detection. Accordingly, the existing pedestrian detectors are often designed based on a general target detection method by targeting the characteristics of on-road pedestrians. As mentioned by [27], [28], typically, the on-road pedestrians have the following characteristics: 1) Varies scales: Unlike other targets, such as vehicles that have relatively uniform outlines/sizes, the scales of pedestrians in pictures tend to have significant differences due to gender, age, and height. Also, the sizes of pedestrians relative to other participants in the transportation system are often smaller. 2) Hard negatives: Due to the visual similarities, in some cases, it is challenging to distinguish pedestrians from similar street backgrounds. 3) Occluded pedestrians in a crowd: Because pedestrians on the road will be concentrated on sidewalks on both sides of the road or pedestrian crosswalks on the intersection. The pedestrian has a large probability of being occluded by its surrounding participants in the transportation system. For example, in the well-known Caltech dataset [27], more than 70% of pedestrians are occluded. In the following part of this section, we go to discuss some representative pedestrian detectors designed by considering the characteristics of on-road pedestrians mentioned above.

To cope with the hard negative issue, in [29], the authors designed a pedestrian detector called task-assistant CNN (TA-CNN). In this work, without implementing pedestrian detection as a single binary classification task, the proposed TA-CNN detects pedestrian by learning high-level features from multiple semantic tasks. Accordingly, the pedestrian is detected by considering pedestrian attributes (e.g., hat, gender, backpack, etc.) and scene attributes (e.g. sky, building, road, etc.) learned from multiple data sources. Also, the structure projection vector (SPV) is adopted to reduce the gap between different datasets. As demonstrated by the authors, the proposed TA-CNN reduced the miss rate on the given test dataset (i.e., Caltech) to a certain extent (2∼5%). Similarly, by taking advantage of the additional features, Mao et al. in [30] designed a new network called HyperLearner to integrate the additional channel features into a CNN-based target detector to implement pedestrian detection. Briefly, by taking Faster RCNN as an example, an additional parallel sub-Conv-network is added besides the Conv-layers. Accordingly, an integrated feature map consisting of segmentation channel of pedestrian and the feature map extracted by the Conv-layer will be fed into the RPN to further generate corresponding proposals. Therefore, without adding references from other sources (like TA-CNN), the detection performance is also improved by around 2%.

Other than the approaches addressing the hard negative issue, there are many pedestrian detectors designed for recognizing pedestrian in a crowded environment. For instance, relying on the conventional HOG and DPM methods discussed previously, Ouyang et al. in [31] designed a method to recognize a single pedestrian from a group of people. Briefly, based on the HOG features derived from the input image, an additional two-pedestrian detector capture extra visual cues based on a mixture of DPM. The final detection results are derived by a specific SVM model, i.e., LatSVM-V2. The basic logic behind this method is to exploit the particular spatial patterns of walking together peoples revealed by [32].

Besides exploiting conventional boundary extraction method, in recent years, many occluded pedestrian detectors designed by taking the high-level feature learning ability of deep CNN. In [33], relying on the general structure of VGG16-based FRCN, the FC-layers is divided to two parallel branches, one (FC-14) for the visible part of pedestrian prediction and the other one (FC-11) for full-body prediction. Accordingly, based on the features of learned by Conv-layers, two types of Bbox can be simultaneously derived for positioning the visible part of the pedestrian and a potential full-body area. Consequently, the corresponding occlusion ratio of the potential pedestrians can also be calculated during the detecting process. By replacing the VGG16 backbone with the ResNet-50 deep network, the authors in [34] designed a detector for detecting pedestrians in a crowd. Technically, this work improves the accuracy of occluded pedestrian detection by reducing the false positives caused by crowd occlusion problem. To achieve this objective, the author designed a loss function, called Repulsion Loss, inspired by the phenomenon of magnets attract and repel, which is defined as,

$$L_{RepLoss} = L_{ATRX} + \alpha L_{Rep-GT} + \beta L_{Rep-Bbox}, \qquad (2)$$

where L_{ATRX} is the attraction part of the equation, which can be considered as a virtual attraction to make the predicted Bbox closer to the potential target. On the other hand, $\alpha L_{Rep-GT} + \beta L_{Rep-Bbox}$ denotes the repel part ensuring that the predicted Bboxes for closing ground truth boxes can maintain a reasonable distance. By this way, the false positives of the occluded pedestrian detection can be reduced to a certain extent.

Besides the pedestrian detectors designed based on the general framework of FRCN, there are many detectors designed based on the Faster RCNN. For example, Occlusion-Aware R-CNN (OR-CNN) introduced in [35] was designed based on the general framework of Faster RCNN. In this work, the authors first designed a new aggregation loss (AggLoss) function for supporting RPN to generate more accurate proposals that closer to the ground truth box. Then, a newly designed Part Occlusion-aware RoI (PORoI) Pooling Unit was adopted to replace the RoI pooling layer in Faster RCNN. More precisely, in addition to the feature $\mathcal{F}$ directly extract from the whole RoI, PORoI divides the input proposal into five parts based on basic structure of human body (i.e., head, left and right parts of human upper torso, and the upper and lower parts of the lower limbs of human) and then extracts features $(\mathcal{F}_1, \mathcal{F}_2, \cdots, \mathcal{F}_5)$ from these five parts respectively. Meanwhile, the corresponding visibility score for each part v_i is also predicted. The final feature with the corresponding visibility score is derived by taking the eltw sum, i.e., $\mathcal{F} \oplus (v_1 \cdot \mathcal{F}_1) \oplus \cdots \oplus (v_5 \cdot \mathcal{F}_5)$. Another pedestrian detector designed based on the Faster RCNN framework is FasterRCNN+ATT detector introduced in [36]. Technically, this approach can be considered as an add-on component, since it did not modify the mainstream of the original Faster RCNN but just adding a separated attention network to provide more accurate information about the visible parts of the potential pedestrian to the final feature maps. Simply put, by taking advantage of the strong response of visible parts of the pedestrian on the heatmaps, three different types of attention networks was designed, i.e., self attention (ATT-self), visible-box attention (ATT-vbb) and part attention (ATT-part). Then, relying on the channel-wise attention mechanism, the Conv features derived by the RoI pooling layer will be re-weighted and then fed into the classification network.

IV. Open Challenges for designing pedestrian detector for supporting autonomous driving

In the previous section, we discussed several representative pedestrian detectors designed by considering the characteristics of on-road pedestrians. However, for supporting the autonomous driving, in addition to considering the inherent characteristics of the pedestrians, the driving characteristics of the vehicle and the impact of its surrounding environment on pedestrian detection should also be considered. While, existing pedestrian detection methods often lack this consideration. Here, we list some open challenges for design designing pedestrian detector for supporting autonomous driving and provide some potential solutions.

A. Requirements for visual distance measurement/estimation

When driving autonomously, the most direct way to ensure the safety of pedestrians and vehicles is that, when pedestrians are detected, the vehicle has enough time to judge the risk of collision, and guarantees that, when emergency braking is required, there is still a sufficient safety distance between the vehicle and the pedestrian. Therefore, sufficient detection distance and real-time detection are vital requirements for a practical pedestrian detector.

TABLE I
Summary of the braking distance and the corresponding braking time

	40Km/h	50Km/h	60Km/h	70Km/h	80Km/h
Dist. (dry)	9m	14m	20m	27m	36m
Time (dry)	1.62s	2.02s	2.4s	2.78s	3.24s
Dist. (wet)	13m	20m	29m	40m	52m
Time (wet)	2.34s	2.88s	3.48s	4.11s	4.68s

Usually, pedestrian and vehicle collisions occur frequently in urban environments. Therefore, in Table I, we summarize the braking distance [37] and the corresponding braking time on wet and dry road without considering the response time at the speed of vehicles commonly seen in an urban environment. By taking 50$\sim$80Km/h speed as an example, we can clearly see that in order to avoid possible collisions, the pedestrian detector must be able to detect pedestrians 20$\sim$50 meters away. In other words, pedestrians in this range are the ones most in need of detection. However, on the basis of the detection of pedestrians, how to effectively judge the distance between pedestrians and the vehicle based on a captured image is still an open challenge to be solved.

To cope with this issue, based on the statistics of the scale of pedestrians in Caltech dataset, by taking the physical parameters of the camera into consideration, the authors in [27] pointed out that, for the images in this database, the distance between pedestrians and vehicle can be estimated based on the number of pixels in the vertical direction occupied by pedestrians. For example, 80 pixels and 20 pixels corresponding to a distance of 20-meters and 60-meter, respectively. However, due to the different parameters of the camera, this estimation method can only be used in this specific dataset. Meanwhile, due to differences in body types, the error of this estimation method will be large. For example, the number of pixels occupied by a child (e.g., one meter high) at the same distance may be only half that of an adult (e.g., 1.8-meter high).

Recently, some approaches have been proposed for implementing visual depth estimation based on the disparity prediction. For example, in [38], the authors showed that, by using the stereo image information, the image-based method could overcome the depth estimation problem to some extent and achieve a relatively closer performance like Lidar. Further, in [39], by taking advantage of the feature pyramid extracted by U-Net [40], according to the top-down direction, the proposed AnyNet derive the disparity map for the input image through the upsampling way in four-stage. Chen et al. in [6] also investigated the pedestrian distance by exploiting the visual disparity information extracted by HOG+Convolutional Channel Features (CCF) from the images captured by the stereo camera. In this work, the authors showed that, the magnitude of the visual disparity value is positively related to how many pixels the pedestrian occupies in the image. Therefore, similar to the problem in [27], The adverse effects on distance estimation results due to differences in body shape are still not well resolved.

In addition to the pedestrian distance estimation methods mentioned above, another potential solution for avoiding a potential collision is the pedestrian trajectory prediction. For example, in [41], based on the new designed deep NN, namely crowd interaction deep neural network (CIDNN), the motion information of a pedestrian (e.g., moving speed, acceleration, etc.) can be recognized from a crowd. Then, relying on the Long Short-Term Memory (LSTM) Recurrent neural network (RNN), the trajectory of the pedestrian can be predicted. In theory, based on such algorithms, an autonomous vehicle can make a corresponding driving strategy, such as deceleration, or emergency braking, by analyzing the probability that a pedestrian's trajectory will cross its own trajectory.

B. Impact of illuminating conditions in the driving environment of vehicles on pedestrian detection

In addition to the visual depth estimation issue mentioned above, the impact of illuminating conditions in the driving environment on the pedestrian is another critical issue that needs to be overcome, because vehicles often need to travel around the clock.

In [42], by considering the change of the pedestrian's space position in the continuous picture captured by the vehicle-mounted visible light camera caused by the relative movement of the pedestrian and the vehicle, the authors designed a night-time pedestrian detector based on Faster RCNN+VGG16 framework. More precisely, The author first adjusts the illumination and contrast of successive pictures to a similar level by a given normalization method, i.e., the pixel normalization method and histogram equalization (HE)+mean subtraction method. Then, the Conv-layer of Faster RCNN extracts the feature from the adjusted image. Instead of feeding the feature from single image directly to the classifier in conventional Faster RCNN, in this work, the authors adopted the weighted summation[2] to combine the features of N continuous image-frames into one integrated feature map and then fed the derived feature into the classifier. Accordingly, the spatial and temporal features of the pedestrian can be captured by the detector, which further improves the pedestrian accuracy in the night-time.

Although the success rate of pedestrian detection under low illumination conditions can be improved to some extent by extracting the spatio-temporal characteristics of pedestrians, the precondition of this method is that the illumination conditions for the target are sufficient to ensure that the target features are extracted from the background. This is why this method achieves high detection accuracy for pedestrians who can be illuminated by the vehicle's headlights on the pedestrian cross-walk, but can not effectively detect the pedestrians walking on the sidewalk that cannot be effectively illuminated by the lights.

Accordingly, the alternative solution is to combine the visible light camera with the thermal camera to realize the detection of pedestrians under nighttime conditions by extracting the target features in the thermal map. For example, in the existing studies [42], [43], the researchers investigated the effects of thermal maps on the accuracy of pedestrian detection when combined with visible light images.

C. Demand for datasets that can accurately reflect the pedestrian status under actual vehicle driving conditions

In addition to the specific requirements for pedestrian detection technology discussed above, there is still a lack of

[2] The corresponding weight for each feature map is given by a 1-D discrete Gaussian distribution according to the pedestrian's positions in those last feature maps.

datasets that can accurately reflect the pedestrian status under actual vehicle driving conditions.

Currently, the pedestrian images in most of well-known image-based objective detection datasets, e.g., Caltech [27], INRIA [9], etc., are not specifically taken from the perspective of the vehicle to accurately reflect pedestrian status. Moreover, for the pedestrian detection method designed based on deep neural network structure design (e.g., RCNN-family) or machine learning based classifier (e.g., SVM), it is necessary to grasp the characteristics of pedestrians as much as possible by learning the data in the data set. Therefore, when the pedestrian characteristics in the data set cannot truly reflect the characteristics of pedestrians in the perspective of the vehicle in motion, we cannot effectively verify the validity of the relevant pedestrian detection method through experimental methods. Consequently, when such methods are applied to a real driving environment, self-driving vehicles may not be able to make the correct driving strategy and cause serious traffic accidents because the pedestrian detector cannot distinguish pedestrians from the perspective of the vehicle.

Currently, the databases, in which, the images are mainly taken from the perspective of the vehicle, are listed as follows:

- KITTI [44]: This dataset published in 2012, which contains 15K day-time images. In detail, these images contain approximately 9K and 3K pedestrians and riders on dry road condition, respectively. Meanwhile, this dataset provides additional images captured by the stereo camera and the Velodyne Lidar system.
- KAIST [45]: This dataset published in 2015, which contains 95K day- and night-time images including around 86K pedestrians on dry road condition. Moreover, the images in this dataset are captured in the perspective of the vehicle by both visible-light camera and thermal camera.
- CityPersons [46]: This dataset published in 2017, which contains 5K day-time images captured by the visible light camera under dry road condition. These images contain around 31K and 3.5K pedestrians and riders, respectively.

Apparently, we can see that, although the pictures in these databases were taken from the perspective of the vehicle, since these pictures were basically acquired under daytime and dry road conditions, the data still could not adequately reflect the complex environment that the vehicle may face when driving under all weather conditions. Therefore, the demand for a more reasonable database is still not fully met. Recently, EuroCity Persons dataset [47] was published, which contains 40K day-time and 7K night-time images captured by visible light camera including 183K (day-time)/35K (night-time) and 18K (day-time)/1.5K (night-time) pedestrians and riders, respectively. Moreover, the images were captured under four-season and dry/wet road conditions. Since the dataset was just published in August 2019, currently, there are not existing pedestrian detectors adopting this dataset for testing.

V. Conclusion

In this paper, we comprehensively studied the existing computer vision-based pedestrian detection methods published in recent years, which are typically designed to take advantage of the powerful computational capacity of GPU and high-level feature learning ability of the deep convolutional neural network. We analyzed and summarized the characteristics of these methods, as well as the specific improvements they have made to the generic image-based target detector in order to meet the requirements of pedestrian detection. Further, by considering the practical application scenarios of autonomous driving techniques, we further discuss the open challenges of designing a practical pedestrian detection method for supporting autonomous deriving.

References

[1] SAE International, “Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles J3016,” [Online]. Available: https://www.sae.org/standards/content/j3016_201806/, 2018, accessed on: May, 2019.

[2] BMW, “Intelligent driving,” [Online]. Available: https://www.bmw.ca/en/topics/experience/connected-drive/bmw-connecteddrive-driver-assistance.html, 2019, accessed on: May, 2019.

[3] Mercedes Benz, “Mercedes safety,” [Online]. Available: https://www.mbusa.com/mercedes/benz/safety, 2019, accessed on: May, 2019.

[4] Audi MediaCenter, “Driver assistance systems,” [Online]. Available: https://www.audi-mediacenter.com/en/technology-lexicon-7180/driver-assistance-systems-7184, 2018, accessed on: May, 2019.

[5] Z. Tang and A. Boukerche, “An improved algorithm for road markings detection with svm and roi restriction: Comparison with a rule-based model,” in *Proc. ICC*, 2018, pp. 1–6.

[6] Z. Chen and X. Huang, “Pedestrian detection for autonomous vehicle using multi-spectral cameras,” *IEEE Trans. Intell. Veh.*, vol. 4, no. 2, pp. 211–219, 2019.

[7] A. Mammeri, D. Zhou, and A. Boukerche, “Animal-vehicle collision mitigation system for automated vehicles,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 9, pp. 1287–1299, 2016.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. NIPS*, 2012, pp. 1097–1105.

[9] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE/CVF CVPR*, vol. 1, June 2005, pp. 886–893.

[10] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *Proc. IEEE/CVF CVPR*, 2008, pp. 1–8.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE/CVF CVPR*, 2016, pp. 770–778.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE/CVF CVPR*, 2014, pp. 580–587.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” in *Proc. ECCV*, 2014, pp. 346–361.

[14] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep 2013.

[15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. ICLR*, 2015.

[16] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Proc. ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., 2014, pp. 818–833.

[17] R. Girshick, “Fast r-cnn,” in *Proc. IEEE ICCV*, 2015, pp. 1440–1448.

[18] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[19] T. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proc. IEEE/CVF CVPR*, July 2017, pp. 936–944.

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

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proc. IEEE/CVF CVPR*, June 2015, pp. 1–9.

[22] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in *Proc. IEEE/CVF CVPR*, July 2017, pp. 6517–6525.

[23] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” [Online]. Available: https://arxiv.org/abs/1804.02767, 2018.

[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *Proc. ECCV*, 2016, pp. 21–37.

[25] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollr, “Focal loss for dense object detection,” in *Proc. IEEE ICCV*, 2017, pp. 2999–3007.

[26] H. Law and J. Deng, “Cornernet: Detecting objects as paired keypoints,” in *Proc. ECCV*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., 2018, pp. 765–781.

[27] P. Dollar, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: An evaluation of the state of the art,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, 2012.

[28] Z. Zou, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” [Online]. Available: https://arxiv.org/abs/1905.05055, 2019.

[29] Y. Tian, P. Luo, X. Wang, and X. Tang, “Pedestrian detection aided by deep learning semantic tasks,” in *Proc. IEEE/CVF CVPR*, June 2015, pp. 5079–5087.

[30] J. Mao, T. Xiao, Y. Jiang, and Z. Cao, “What can help pedestrian detection?” in *Proc. IEEE/CVF CVPR*, July 2017, pp. 6034–6043.

[31] W. Ouyang, X. Zeng, and X. Wang, “Single-pedestrian detection aided by two-pedestrian detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1875–1889, 2015.

[32] M. Moussad, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, “The walking behaviour of pedestrian social groups and its impact on crowd dynamics,” *PLOS ONE*, vol. 5, no. 4, pp. 1–7, 2010.

[33] C. Zhou and J. Yuan, “Bi-box regression for pedestrian detection and occlusion estimation,” in *Proc. ECCV*, 2018, pp. 138–154.

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

[35] S. Zhang, L. Wen, X. Bian, Z. Lei, and tan Z. Li, “Occlusion-aware r-cnn: Detecting pedestrians in a crowd,” in *Proc. ECCV*, 2018, pp. 657–674.

[36] S. Zhang, J. Yang, and B. Schiele, “Occluded pedestrian detection through guided attention in cnns,” in *Proc. IEEE/CVF CVPR*, 2018, pp. 6995–7003.

[37] Queensland Government, “Stopping distances on wet and dry roads,” [Online]. Available: https://www.qld.gov.au/transport/safety/road-safety/driving-safely/stopping-distances/graph, 2016, access on: Aug., 2019.

[38] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, “Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving,” in *Proc. IEEE/CVF CVPR*, 2019, accepted.

[39] Y. Wang, Z. Lai, G. Huang, B. H. Wang, L. van der Maaten, M. Campbell, and K. Q. Weinberger, “Anytime stereo image depth estimation on mobile devices,” in *Proc. IEEE ICRV*, 2019, accepted.

[40] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Proc. MICCAI*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., 2015, pp. 234–241.

[41] Y. Xu, Z. Piao, and S. Gao, “Encoding crowd interaction with deep neural network for pedestrian trajectory prediction,” in *Proc. IEEE/CVF CVPR*, June 2018, pp. 5275–5284.

[42] J. H. Kim, G. Batchuluun, and K. R. Park, “Pedestrian detection based on faster r-cnn in nighttime by fusing deep convolutional features of successive images,” *Expert Syst. Appl.*, vol. 114, pp. 15 – 33, 2018.

[43] D. Xu, W. Ouyang, E. Ricci, X. Wang, and N. Sebe, “Learning cross-modal deep representations for robust pedestrian detection,” in *Proc. IEEE/CVF CVPR*, July 2017, pp. 4236–4244.

[44] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Proc. IEEE/CVF CVPR*, June 2012, pp. 3354–3361.

[45] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, “Multispectral pedestrian detection: Benchmark dataset and baseline,” in *Proc. IEEE/CVF CVPR*, 2015, pp. 1037–1045.

[46] S. Zhang, R. Benenson, and B. Schiele, “Citypersons: A diverse dataset for pedestrian detection,” in *Proc. IEEE/CVF CVPR*, July 2017, pp. 4457–4465.

[47] M. Braun, S. Krebs, F. Flohr, and D. M. Gavrila, “Eurocity persons: A novel benchmark for person detection in traffic scenes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1844–1861, 2019.