Performance Evaluation of Pedestrian Detectors for Autonomous Vehicles

Mingzhi Sha, Azzedine Boukerche
PARADISE Research Laboratory, EECS, PARADISE Research Laboratory, EECS, University of Ottawa, Canada
Emails: msha096@uottawa.ca, boukerch@eecs.uottawa.ca

***Abstract*—With the development of deep learning methods, adopting CNN-based detectors has become a trend to handle the detection task. The proposal of Intelligent Transportation Systems (ITS) has once again brought autonomous vehicles into the public eye. Pedestrian detection can ensure pedestrians' safety, and it is considered one of the most challenging problems that urgently need to be solved. We have noticed that researchers use various environments when publishing experimental results, leading to unfair comparisons of experimental results. Under different computing resources, the performance of the detector may be weakened or enhanced. In this paper, we will compare two representative detectors with the same computing power for a fair comparison study, aiming to find out how experimental settings affect the detector's accuracy.**

***Index Terms*—Intelligent transportation system, autonomous vehicle, pedestrian detection, convolutional neural network.**

I. Introduction

The possibility of widespread application of autonomous vehicles has been significantly increased with the extensive research on Intelligent Transport Systems-based communication and management protocols, such as [1]–[7]. Pedestrians are essential transportation participants, and their safety is non-negotiable. Pedestrian detection is one of the methods to protect the pedestrians' safety, which has a wide range of application requirements and can be applied to driver assistance systems and autonomous vehicles.

The development of deep learning and GPU computing power has enabled more researchers and industries to work in the field of computer vision and detection area. Many works have been proposed in recent years, such as [8]–[12].

It can be found that most of these proposed detectors (e.g., [13]–[17]) were trained and tested in unclear or different environmental settings: such as training with varying amounts of GPUs or some types of GPUs which are not in the same level of computing power. The various experimental conditions could have a considerable impact on the detector's accuracy and efficiency, which could obscure researchers' work's contribution. The GPUs are getting more powerful in recent years, making it unfair when researchers directly cite previous experimental results and compare detectors published in different years.

The relationship between the GPUs and detectors' performance is the elephant in the room. Using more GPUs with more powerful computing ability can significantly boost the detector's accuracy. However, this would introduce a high cost on GPUs, and consuming massive computing power may not be compatible with the autonomous vehicles' system requirements. The improvement of pedestrian detectors' performance should never rely solely on the use of greater computing power.

In this paper, we train two representative detectors by using the same environmental hardware on the CityPersons dataset [18] and the ETH dataset [19]. We compare these two chosen detectors under the same environmental and evaluation settings to make a fair comparison and explore how experimental settings and parameters could affect the performance.

II. Related Work

Pedestrian detection highly relied on traditional machine learning methods before Faster R-CNN achieved promising performance, such as [20] and [21]. Traditional machine learning methods apply the pre-defined hand-crafted feature descriptor to the entire image and perform pattern matching. Differently, CNN-based methods enable the detector to learn the feature of the target from the given dataset.

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 [22]–[24], 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. Differently, one-stage detectors predict the class and location simultaneously, without searching ROI proposals first, e.g., YOLO-v1 [25] and SSD [26].

The concept of 'anchor' was proposed in RPN [24], 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 points. The anchor mechanism is used to predict ROI proposals in two-stage

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

978-3-903176-32-4 © 2021 IFIP

detectors (e.g., Faster R-CNN [24] and Mask R-CNN [27]) and candidate bounding boxes in one-stage detectors (e.g., SSD [26] and YOLO-v2 [28]).

Instead of applying anchors into usage, YOLO-v1 [25] achieved an anchor-free detecting pipeline by dividing the images into grids to generate corresponding predictions. DeNet [29] avoids the use of the anchor mechanism by predicting the four keypoints of the target. Another anchor-free pedestrian detector TLL [30] was designed to predict top-bottom points and the topological line.

III. 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. We list some key characters of the methods we use in the Table I.

TABLE I
Detectors comparison.

Detectors	***Year***	***# of stages***	***Anchor***
Faster R-CNN	2015	2	Yes
CSP	2019	1	No

We adopt Faster R-CNN [24] (with FPN [31]) and CSP [17] in our comparative experiments. We demonstrate the model structure in Figure 1, and we list the environmental settings in Table III.

- **Faster R-CNN [24]** As one of the most representative two-stage detectors, Faster R-CNN is still widely used today. The concept of ‘anchors’ comes from Faster R-CNN. To better take advantage of the framework, we use FPN [31] to enrich the extracted features, which will be called Faster R-CNN + FPN in the rest of this paper. By proposing a pyramidal hierarchy network, FPN leverages the rich semantic features in high-level feature maps and fine localization information in high-resolution feature maps. We apply HRNet [32] and ResNet-50 [33] as the backbone respectively to find out the influence of backbones and understand the performance of this classic two-stage detector.
- **CSP [17]** By predicting the centerness of the pedestrian and the corresponding scales, the authors designed this one-stage anchor-free detector. Unlike the CornerNet [34], which took advantage of two corner keypoints, the success of CenterNet [35] indicated the importance of the center keypoint. The appearance of CSP emphasized the effectiveness of the object’s centerness.

It is found that more detectors are designed as one-stage detectors, and many anchor-free detectors have appeared, especially since 2019. These two selected detectors can cover a variety of different categories.

IV. Experiment

We perform the experiments on the CityPersons dataset [18] by training them with proper hyper-parameters and evaluate their accuracy on seven setups of the CityPersons validation set. We apply the detectors trained on the CityPersons dataset to the ETH dataset [19] to investigate their performance on unseen data.

A. Experimental settings

In the experiment, all the detection models are trained on a single Nvidia GeForce GTX 1080 Ti GPU, with 11GB GPU memory. We adopt the official work provided by authors of [17] and the MMDetection toolbox [36] 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 III. 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. 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.

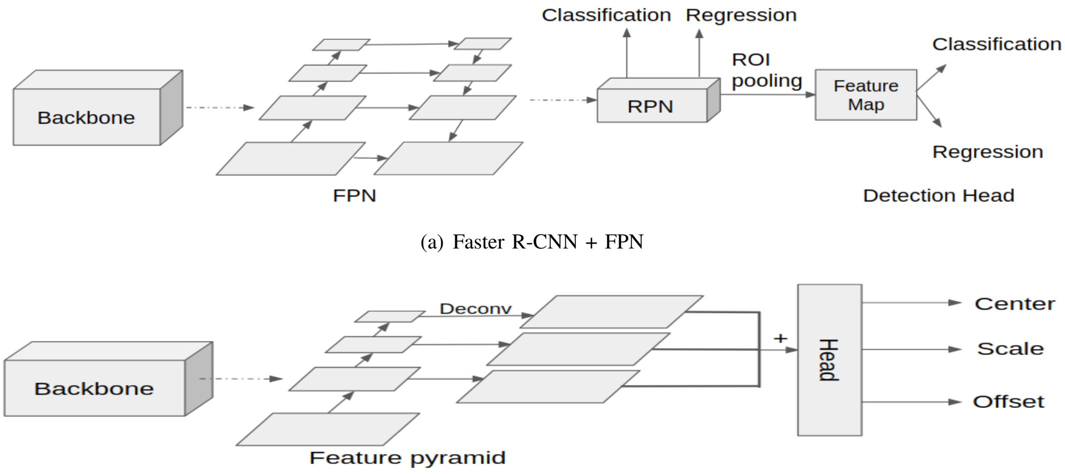
B. Evaluation

1) The CityPersons dataset: We test detectors on the CityPersons validation set with original image resolution 1024×2048, and the detailed configuration of each setup is listed in Table II. 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. The evaluation results on the CityPersons dataset are shown in Table IV, and we plot the bounding boxes on the same image from the CityPersons dataset in Figure 3 for visualization.

HRNet achieved promising accuracy performance in [32], indicating its powerful feature extraction capabilities. In our experiments, the resolution of training images of Faster R-CNN + FPN with ResNet-50 is larger than that of Faster R-CNN + FPN with HRNet. They achieve comparable performance, especially on Reasonable, Bare, and Partial setups. Our experimental results show that the image resolution will affect the accuracy of the detector.

We list the experimental results from the authors in Table V. The authors of [18] did not provide clear information about the GPU used in the Faster R-CNN experiment, which prevents us from making comparisons. We find by using four GPUs, CSP obtained a MR^{-2} of 11.0% on the Reasonable setup, which is 1.82 % better than the result obtained on one GPU. The performance of CSP on Small and Partial setups dramatically drops when it is trained with 1 GPU.

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 and support autonomous vehicles in real-world scenarios.



(a) Faster R-CNN + FPN

(b) CSP

Fig. 1. Model structure illustrations of the chosen detectors.

TABLE II
EVALUATION SETUPS OF THE CITYPERSONS DATASET [18]

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

TABLE III
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
CSP	ResNet-50	640×1280	2

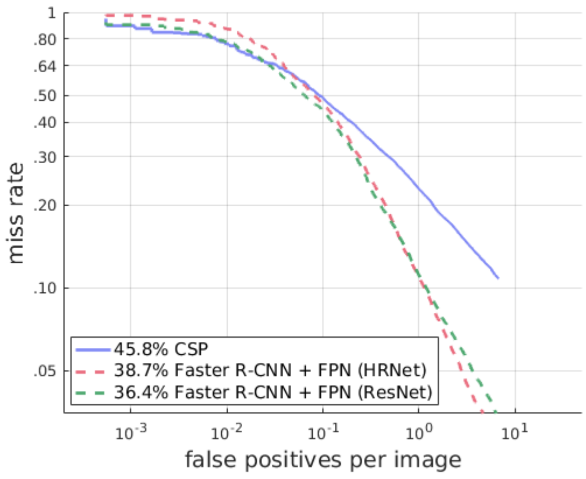


Fig. 2. Evaluation results on the ETH dataset [19]

2) The ETH dataset: The ETH dataset [19] has 1804 images in total, and the image resolution is 480×640, which is much smaller compared to 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 the Fig. 2. It can be observed that Faster R-CNN + FPN with ResNet-50 achieves the best performance, and Faster R-CNN + FPN with HRNet achieves a moderate performance. The performance of CSP drops significantly, compared with Faster R-CNN + FPN. 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.

V. CONCLUSION

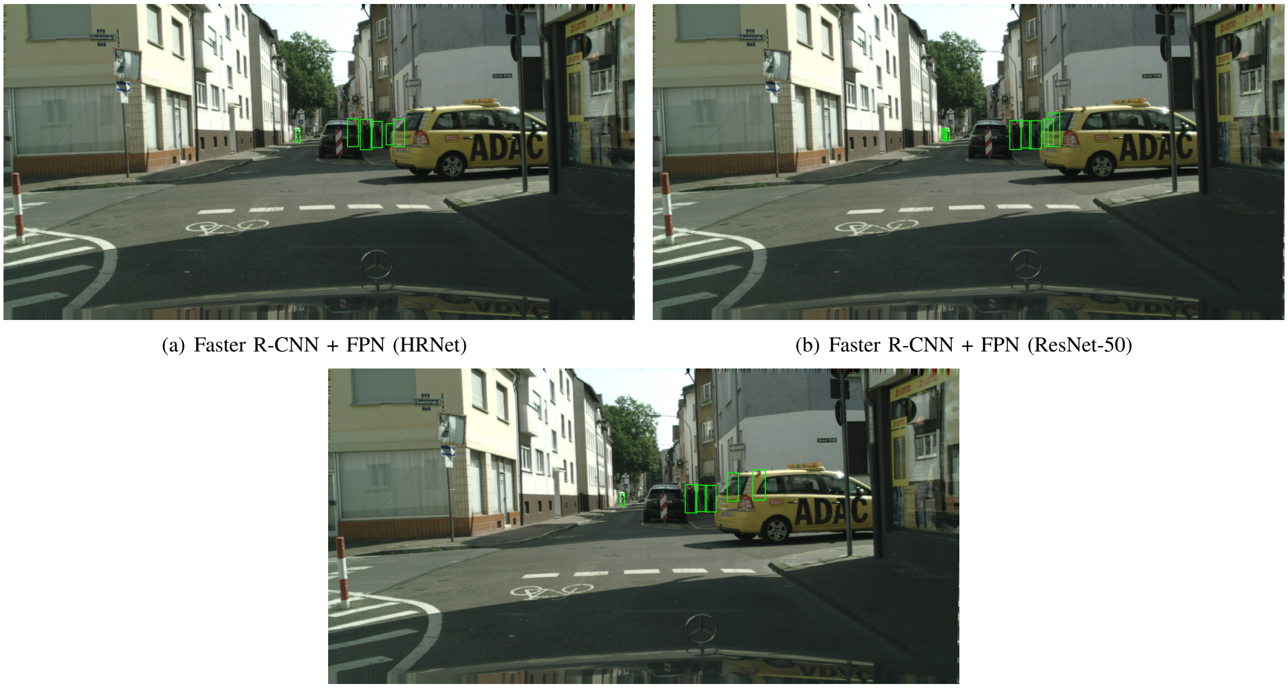
In this comparative study paper, our motivation was to compare two different detectors on the same environmental settings. We explained the reasons why we chose these two representative detectors in our study. From the evaluation results, we found the backbone and image resolution can affect the detector's accuracy. The trade-off of these hyper-parameter settings and network architectures is highly related to the model complexity, which can be limited by the GPUs, especially vehicular GPUs. Another open challenge in the pedestrian detection area is the generalization ability, as we

TABLE IV
EXPERIMENTAL RESULTS ON THE CITYPERSONS VALIDATION SET. THE RESULTS IN BOLDFACE INDICATE THE BEST ON THE CORRESPONDING SUBSETS.

	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
CSP	**12.82**	**8.27**	**12.75**	51.14	**19.23**	**4.78**	**7.27**

TABLE V
WE CITE THE EXPERIMENTAL RESULTS ON THE CITYPERSONS VALIDATION SET FROM ORIGINAL AUTHORS.

	GPU	*Reasonable*	*Bare*	*Partial*	*Heavy*	*Small*	*Medium*	*Large*
Faster R-CNN [18]	-	15.4	-	-	64.83	25.6	7.2	7.9
Faster R-CNN + Semantic [18]	-	14.8	-	-	-	22.6	6.7	8.0
Faster R-CNN + ATT-self [10]	-	20.93	-	-	58.33	-	-	-
CSP [17]	4 GPUs (1080Ti)	11.0	7.3	10.4	49.3	16.0	3.7	6.5



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

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

(c) CSP

Fig. 3. Visualization results on the CityPersons dataset.

demonstrated in the evaluation results on the ETH dataset. In future, we will work on improving the generalization ability of pedestrian detectors.

REFERENCES

[1] R. I. Meneguette and 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, and 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, and 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, and S. K. Das, “OBQR: orientation-based source qos routing in vanets,” in *Proc. ACM MSWiM*, 2020, pp. 199–206.

[5] N. Aljeri and A. Boukerche, “An adaptive traffic-flow based controller deployment scheme for software-defined vehicular networks,” in *Proc. ACM MSWiM*, 2020, p. 191–198.

[6] E. Kalogeiton and T. Braun, “On the impact of sdn for transmission power adaptation and fib population in ndn-vanets,” in *Proc. ACM MobiWac*, 2020, p. 57–66.

[7] A. Madhja, S. Nikoletseas, D. Tsolovos, and A. A. Voudouris, “Peer-to-peer energy-aware tree network formation,” in *Proc. ACM MobiWac*, 2018, p. 1–8.

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

[9] H. Huang and S. Lin, “Widet: Wi-fi based device-free passive person detection with deep convolutional neural networks,” in *Proc. ACM MSWiM*, 2018, pp. 53–60.

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

[11] P. Sun and A. Boukerche, “Challenges and potential solutions for designing a practical pedestrian detection framework for supporting autonomous driving,” in *Proc. ACM MobiWac*, 2020, p. 75–82.

[12] M. Sha and A. Boukerche, “Semantic fusion-based pedestrian detection for supporting autonomous vehicles,” in *Proc. IEEE ISCC*, 2020, pp. 1–6.

[13] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, “Occlusion-aware r-cnn: Detecting pedestrians in a crowd,” in *Proc. ECCV*, 2018, pp. 637–653.

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

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

[16] S.-K. L. Chengju Zhou, Meiqing Wu, “SSA-CNN: semantic self-attention CNN for pedestrian detection,” [Online]. Available:http://arxiv.org/abs/1902.09080, 2019, accessed on: Oct., 2020.

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

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

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

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

[21] C. P. Papageorgiou, M. Oren, and T. Poggio, “A general framework for object detection,” in *Proc. IEEE ICCV*, 1998, pp. 555–562.

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

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

[24] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Proc. NIPS*, 2015, pp. 91–99.

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

[26] 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.

[27] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proc. IEEE ICCV*, 2017, pp. 2961–2969.

[28] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” [Online]. Available: http://arxiv.org/abs/1612.08242, 2016, accessed on: Oct., 2020.

[29] L. Tychsen-Smith and L. Petersson, “Denet: Scalable real-time object detection with directed sparse sampling,” in *Proc. IEEE ICCV*, 2017, pp. 428–436.

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

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

[32] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” in *Proc. IEEE CVPR*, 2019, pp. 5693–5703.

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

[34] H. Law and J. Deng, “Cornernet: Detecting objects as paired keypoints,” in *Proc. ECCV*, 2018, pp. 734–750.

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

[36] K. Chen and et al., “Mmdetection: Open mmlab detection toolbox and benchmark,” [Online]. Available:https://arxiv.org/pdf/1906.07155.pdf, year = 2019, accessed on: Oct., 2020.