Flexible Neural Network for Fast and Accurate Road Scene Perception

Sabeeha Mehtab, Wei Qi Yan

Auckland University of Technology, Auckland, 1010 New Zealand

Abstract. Accurate object detection on the road is the most important requirement of autonomous vehicles. Extensive work has been accomplished for car, pedestrian, and cyclist detection; however, comparatively, very few efforts have been put into 2D object detection. In this article, a dynamic approach is investigated to design a perfect unified neural network that could achieve the best results based on our available hardware. The proposed architecture is based on CSPNet for feature extraction in an end-to-end way. The net extracts visual features by using backbone subnet, visual object detection is based on a feature pyramid network (FPN). In order to increase the net flexibility, an auto-anchor generating method is applied to the detection layer that makes the net suitable for any datasets. For fine-tuning the net, activation, optimization, and loss functions are considered along with multiple check points. The proposed net is trained and tested based on the benchmark KITTI dataset. Our extensive experiments show that the proposed model for visual object detection is superior to others, where other nets output very low accuracy for pedestrian and cyclist detection, our proposed model achieves 99.3% recall rate based on our dataset.

Keywords: Deep neural network, road scene perception, autonomous vehicles, self-driving car, YOLOv5

1 Introduction

In the field of autonomous vehicles, accurate road scene perception plays a vital role to avoid accidents. Despite of plentiful advancements, visual object detection is still a challenging task. It demands a great deal of efforts, especially for vulnerable road users like pedestrians and cyclists that occupy more than half of on-road death tolls as published by the World Health Organization (WHO) [1]. The situation becomes even worse in bustling streets or under extreme weather conditions where drivers' visibility is compromised due to truncation, occlusion, distances, and lighting conditions [2].

Traditional computer vision approaches were based on Histogram of Oriented Gradient (HOG) [3] for feature representation with the classifiers like SVM (i.e., Support Vector Machine). However, recent development in Graphics Processing Unit (GPU) has provided opportunity to successful solutions based on deep learning. The

remarkable success of AlexNet [4], which won the ImageNet Classification Challenge 2012, opened a door for very deep neural networks (DNNs). VGGNet [5] and GoogLeNet [6] obtained similar performance using deeper net architecture. Afterwards, Inception architecture was proposed as a net with multiple kernels [7] aiming at effective computations. ResNet [8] and DenseNet [9] emphasized on carrying forward the residual information to avoid extreme compression of ground information by using skip connections and direct connections between subsequent layers, respectively. Interestingly, these models were reused in other computer vision applications like object segmentation [7].

The popular approaches for visual object detection like Faster R-CNN [10] and Mask R-CNN [11] are based on Region Proposal Network (RPN) and Region of Interest (ROI) in the first stage and regression at the second stage for refining the detected object. On the other hand, the object detection approach in the end-to-end way has been adopted by YOLO (i.e., You Only Look Once) models [12], Single Shot MultiBox Detector (SSD) [13], and EfficientDet [14] networks which are based on a single regression network to detect visual objects in a speedy way.

In this paper, we aim to detect and classify cars, pedestrians, and cyclists in real time by using a unified DNN with high accuracy. In the proposed solution, we investigate a flexible neural network. The contributions of this paper are summarized as follows:

- The proposed method assigns different depth and width to the given baseline net dynamically to achieve the optimum outcomes based on the available hardware.
- The baseline architecture is designed by using CSPNet in an end-to-end way for feature extraction and object detection.
- An auto-anchor method is proposed for multiscale object detection based on *k*-means clustering to generalize the network.

A PyTorch-based framework is proposed to attain short detection time that allows automated vehicles to make decisions timely. The net architecture is inspired by YOLOv5 [15], based on a single regression net for visual object detection. The optimized model is fine-tuned by using optimizations and loss functions to achieve the desired accuracy. Finally, the performance is evaluated based on the benchmark KITTI dataset [16] that provides complicated and challenging conditions to evaluate the performance of the proposed neural network.

The remaining part of this paper is organized as follows. The related work is critically reviewed in Section 2. Furthermore, the proposed research method is outlined in Section 3. Next, our experimental results are demonstrated and discussed in Section 4, following the conclusion in Section 5.

2 Related Work

2.1 Vehicle Detection

An extensive literature survey is conducted for visual object detection in the field of autonomous vehicles. A myriad of DNNs have been proposed for vehicle detection. Faster R-CNN was applied on video frames to get the desired accuracy of vehicle detection in road scene perception [17], though the algorithm has a limitation in the speed of inference. Cao et al. [18] improved the basic structure of SSD by adding inception blocks and feature fusion layers in the original network to detect tiny objects accurately. The deep MANTA [19] was designed based on the principle of RPN to find ROIs that were passed into two convolutional layers and fully connected layers to get 2D bounding boxes and key parts of vehicles. MSVD_SPP method [20] modified YOLOv3 [21] by using five special pyramid pooling (SPP) blocks in visual feature extraction. Sang et al. [22] modified YOLOv2 by combining *k*-means++ clustering algorithm for generating best-fit anchor box. Wang et al. [23] exploited Faster R-CNN by including multishape receptive field and anchor optimizations.

On the other hand [24][25], visual features such as HOG [3] or Haar-like features [27] have been exploited to detect cars. A hierarchical HOG symmetrical feature was applied to various sides of vehicles, a modified HOG version was introduced to cover gradient information from different viewpoints [24]. Furthermore, a two-step object detection algorithm [25] was proposed based on the combined results of HOG and Harr-like features.

2.2 Pedestrian and Cyclist Detection

In the field of pedestrian detection, low-level image features have been employed exhaustively to produce ROIs based on different sensors [28]. These models exploited HOG features along with multiple methods, like decision tree or Local Binary Pattern (LBP) [29]. HOG descriptor with SVM (i.e., Support Vector Machine) classifier has been applied [3] to pedestrian detection with remarkable success. However, these hand-crafted methods remained susceptible to occlusion and other complex environments, not suitable for real-time scenarios. However, recent deep learning methods allow the network to produce high-level features of objects with real-time processing speed [30][31].

In human recognition, thermal images have been employed in many applications [32] for their heat-sensitive features. Pertaining to pedestrian and cyclist detection, thermal images resulted in better performance than RGB images in poor visibility conditions [26][33]. A fusion was conducted on thermal and camera RGB images with two parallel SSD detection streams [26]. Faster R-CNN was exploited to adaptively merge both modalities by a subnetwork of gated fusion [32]. A unified framework based on Fast R-CNN was employed for pedestrian and cyclist detection via multilevel feature fusion [34].

Car and pedestrian detections have contributed to road scene perception [35][36]. Song et al. [35] engaged in car and pedestrian detections by using SSD model with MobileNet backbone subnet for a faster detection ratio. Yang et al. [36] endeavored for car and pedestrian detection based on YOLOv2 by replacing *k*-means clustering algorithm of anchor generation with a prior knowledge of objects in the database.

A RetinaNet-based model was proposed [2] to detect cars, pedestrians, and cyclists in autonomous driving by using RGB and stereo images along with LiDAR point clouds. In [2], two models, namely, Stacked Fusion Double RetinaNet (SFD-Retina) and Gated Fusion Double RetinaNet (GFD-Retina) with multiple fusion styles were proposed. Liu et al [37] proffered a lightweight neural network to detect visual objects on the road by using limited computing resources while preserving the accuracy. Later, the previous work was refined [38] by using CentreNet-based anchor-free approach by bringing in Atrous Spatial Pyramid Pooling (ASPP) to extract visual features of multiscale objects with low computational costs.

Based on literature review, most 2D object detection in the autonomous vehicle focused on individual object with little research outputs on the unified framework. Consequently, multiple pipelines need to be run parallel, which lead to an increased number of operations and slow down the detection speed. On the other hand, the general outcomes of the object detection framework [2] result in low accuracy of pedestrian and cyclist detection compared to vehicle detection. Therefore, the proposed flexibleNet based on unified framework is significant which is able to achieve desirable precision and recall for pedestrian and cyclist detection and suits fast road scene perception.

3 Research Methods

In this section, we have firstly discussed the proposed DNN architecture deduced from the recent YOLOv5 network [15], followed by the auto-anchor generating method.

3.1 FlexiNet: Flexible Neural Network

The architecture and complexity of deep neural networks have shown powerful ability in feature extraction [4][5][6], however, it is only suitable for costly hardware components. Moreover, naively increasing the network depth also results in overfitting and vanishing gradient problems [14][40]. On the other hand, a wider network captures fine-grained features much precisely [39]. However, our empirical results as shown in Fig. 3 depict that going too wide also leads to decreased accuracy. Thus, we propose a FlexiNet model that allows to dynamically define a network structure by using the multiple depths and widths attributes of the baseline architecture to achieve the optimum accuracy for the existing database with available hardware resources.

In order to avoid losing residual spatial information with very deep networks, ResNet models were proposed with the skip connections [40], PANet was based on adaptive feature pooling [41], whereas DenseNet [9] and CSPNet [42] were proposed with cross-stage hierarchy to boost the flow of gradient information. Followed YOLOv4 [12], the proposed model exploits CSPNet [42] as the basic block that split the path of gradient

flow into two streams followed by concatenation and transition blocks to extract complex features of the given images as shown in Fig. 1. CSPNet has been proven to converge faster with no extra storage cost [42][12].

The proposed FlexiNet consists of multiple CSPNet blocks followed by convolutional blocks called partial transition blocks, which perform a hierarchical feature fusion mechanism [42]. The number of channels and the number of layers in FlexiNet are decided based on the depth and width. Every convolutional block comprises a Conv2D layer followed by a batch normalization and SiLU activation function [43].

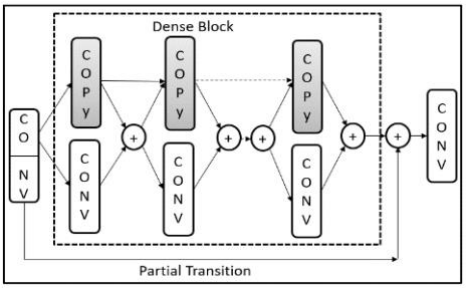


Fig. 1. CSPNet as a basic block in FlexiNet model with dynamic scaling

Against the constraint of fixed size image restriction in object detection, the proposed model makes use of a Spatial Pyramid Pooling layer (SPP) [44], which was successfully adopted in a number of end-to-end detection architectures. The SPP layer concatenates the feature maps produced from three intermediate convolutional layers yielding a fixed-length representation with the increased receptive field. Fig. 2 shows the FlexiNet baseline architecture, the final size of the net is evaluated concerning the parameters *depth_multiple* and *width_multiple*. Eq. (1) represents the formation of each block in flexible neural network based on the assigned depth_multiple and wedth_multiple parameters.

$$\begin{cases} Final\_layers\_in\_block = no\_of\_layers\_in\_block \times depth\_mutiple, \\ Final\_channels\_in\_block = no\_of\_channels\_in\_block \times width\_mutiple \end{cases} \quad (1)$$

In DNN, every convolutional layer provides feature extraction that results in losing fine-grained features. In order to deal with this problem, the proposed method extracts feature maps from three stages BB-s1, BB-s2, and BB-s3, in the backbone subnet.

Regarding object detection with feature extraction, the head module of the architecture is influenced by YOLOv3 [45] and YOLOv4 [12]. In the head section, object detection is fulfilled by using FPN [46] for different size objects by using multiscale anchors. As shown in Fig. 2, visual object detection is accomplished at three stages H-S1, H-S2, H-S3. However, multistage detection results in various outcomes of bounding boxes of the same object. Using non-max suppression, these extra boxes are removed by keeping the one with the highest confidence score.

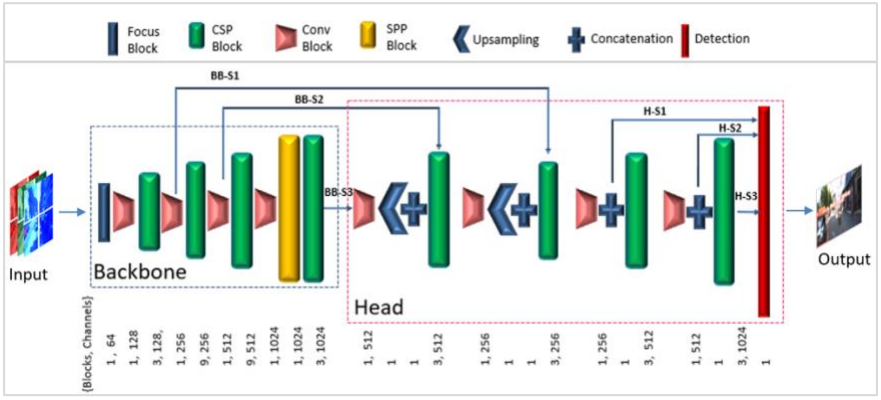


Fig. 2. The architecture of FlexiNet. The first module in the architecture is the backbone subnet by providing features extraction from three different stages. The head section of the network performs detection at three stages with multiscale objects.

3.2 Auto-Anchor

In order to obtain a high level of flexibility, an auto-anchor generating method is proposed by using *k*-means clustering algorithm. The proposed algorithm at first generates clusters based on IoU (i.e., intersection over union) by using the ground truth of bounding boxes (`GT_BBs`) at three scales. Secondly, the mean anchor sizes (mean_Anchor_size) are confirmed for each cluster. The pseudocode for the proposed algorithm is shown as follows:

```
Input: data GT_BBs,m
initialize Anchor_Size[3xm] with  base_values
no_change = False

repeat
  #generate 3xm clusters
  for i in Gt_BBs:
            associate Gt_BB[i] with a AnchorSize based on minimum IoU

  #claculate mean_AnchorSize[nxm] of each cluster
  For i in 3:
    For j in m:
      find mean_AnchorSize[ixj] based-on GT_BBs in cluster[i,j]
      if Anchor_size[i,j] == mean_AnchorSize[i,j]:
        no_change = True
      else:
        Anchor_size[i,j] = mean_AnchorSize[i,j]

untill no_change= False:
Output: Anchor_Size[3xm]
```

3.3 Workflow

Fig. 3 illustrates the workflow of 2D road scene perception by using the proposed flexiNet model. In order to improve the network performance, different fine-tuning strategies are followed based on gradient descent optimizers, loss functions, and different check-points.

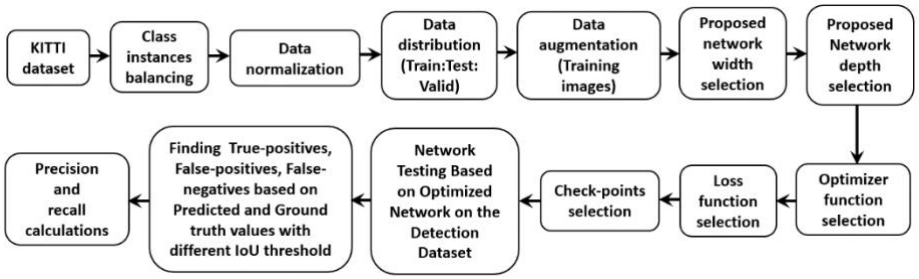


Fig. 3. The pipeline of the proposed model

4 Experimental Results and Evaluations

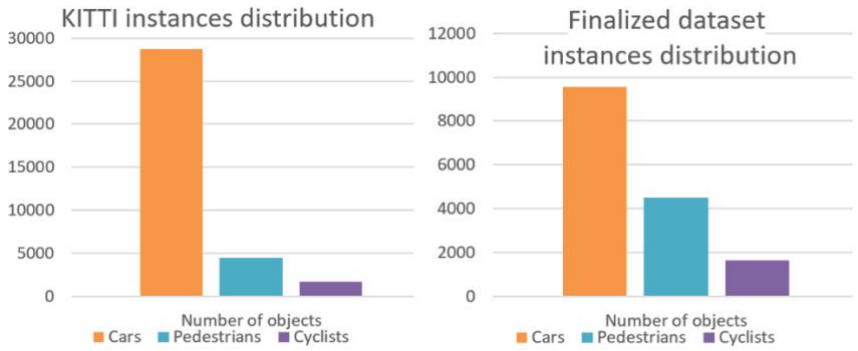
Regarding the model evaluations, we have primarily adopted a benchmark KITTI dataset [16]. The dataset particularly aims to push forward the development of computer vision and robotic algorithms for autonomous vehicles. In the KITTI dataset, there are 7481 labelled images with the average resolution 1350×350. Basic classes of interests are taken into consideration, including Car, Van, Pedestrian, and Cyclist. Fig. 4(a) shows the distribution of data instances in the KITTI dataset. Taken a balance between the class instances into account, 4,000 images were adopted for training, validation, and testing with the proportion 8:2:2. Fig. 4(b) depicts the instance distribution in final dataset. All images were scaled to 640×640 resolution that have been normalized. Motivated by the latest progress [12][15], we exploited CutMix [47] and image mosaic method as the augmentation based on the training dataset with a wider range of semantic variations. All experiments were run by using Tesla P100-PCIE-16GB GPU with 16 GB memory. The test results encapsulate precision, recall, mAP, bounding box, loss function and object loss function to demonstrate the capacity of the proposed model. The equations are shown as eq. (2)(3)(4).

$$Precision = tp/(tp + fp) \tag{2}$$

$$Recall = tp/(tp + fn) \tag{3}$$

$$mAP = 1/n \sum_{i=1}^{n} Precision_i \tag{4}$$

where *precision*, *recall*, and *mAP* are calculated by using true positive (*tp*), false positive (*fp*), and false negative (*fn*) based on the predicted results.

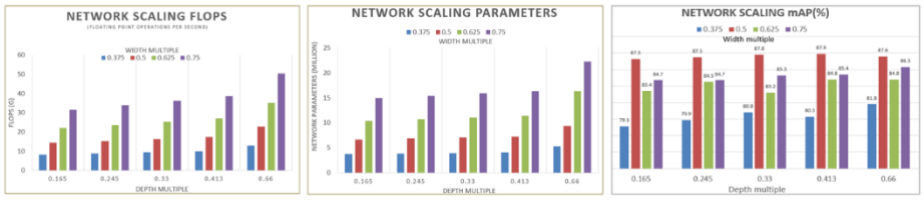


(a) (b)

Fig. 4. The sample distribution (a) The labelled KITTI dataset with the ratio of object numbers. (b) The sorted KITTI dataset in the experiments including all images containing pedestrians or cyclists with additional images

4.1 Network Optimization

In this section, the optimized network has been identified to achieve the best performance based on the existing hardware. Fig. 5(a), (b), and (c) show Floating Point Operations (FLOPs), involved network parameters, and final mAP@0.5IoU results, respectively, for a set of models based on FlexiNet baseline. Interestingly, there is a huge difference in FLOPs and network parameters as the network goes deeper and wider; however, the trends are not as the same with precisions.



(a) (b) (c)

Fig. 5. Net scaling results. (a) FLOPs executed with different networks (b) Parameters stored in different networks (c) mAP@0.5IoU obtained at different networks

As shown in Fig. 5(c), increasing the width of a net provides a significant improvement with the precision initially; however, it eventually results in sinking the performance. On the other hand, going deeper into the network improves the results at first, later leads to saturation, excessive computational and storage overheads. For the given dataset, FlexiNet attained 87.8% mAP @ 0.5 IoU with the scaling factors of *width* (0.5) and *depth* (3.3×10^{-1}), exploiting the minimum hardware with Adam optimizer [48] by using GIoU loss function [49].

Optimizer selection plays a vital role in the deep learning pipeline. In order to improve the accuracy of outputs, we investigated the SGD optimizer [50] with respect to Adam [48] based on our hyperparameters set having learning rate 1.0×10^{-2}, momentum 0.9, and weight decay 5.0×10^{-3}. Fig. 6 depicts that SGD optimizer with Adam method achieved 92.7% mAP and 87.4% recall based on the given training dataset. We see that SGD is superior to adaptive learning methods [51][52].

(a) (b) (c)

Fig. 6. Training and validation results based on Adam and SGD optimizer (a) mAP@0.5IoU curves obtained based on training dataset (b) obtained recall values based on training dataset and (c) objectness loss curves based on the validation dataset

The regression loss function for basic bounding boxes in the proposed model is GIoU, in order to further explore the model, we have tested recently published loss functions [49] based on the optimized net structure. The loss functions tested in this paper are shown as Eq. (5), (6), and (7).

$$\mathcal{L}_{GIoU} = \frac{|A\cap B|}{|A\cup B|} - \frac{|C\backslash(A\cup B)|}{|C|} \tag{5}$$

where C is the minimal closer area of bounding boxes A and B.

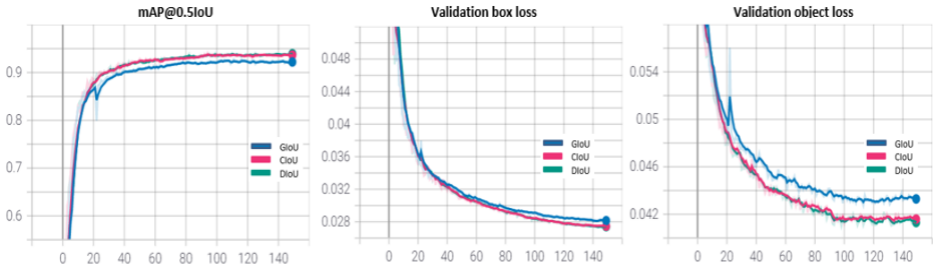
$$\mathcal{L}_{DIoU} = 1 - \frac{|A\cap B|}{|A\cup B|} + \frac{\rho^2(a,b)}{c^2} \tag{6}$$

where a and b are central points of ground truth and predicted boxes, $\rho(\cdot)$ is the Euclidean distance, c is the diagonal length of the smallest enclosing box covered by the two bounding boxes.

$$\mathcal{L}_{CIoU} = 1 - \frac{|A\cap B|}{|A\cup B|} + \frac{\rho^2(a,b)}{c^2} + \alpha v \tag{7}$$

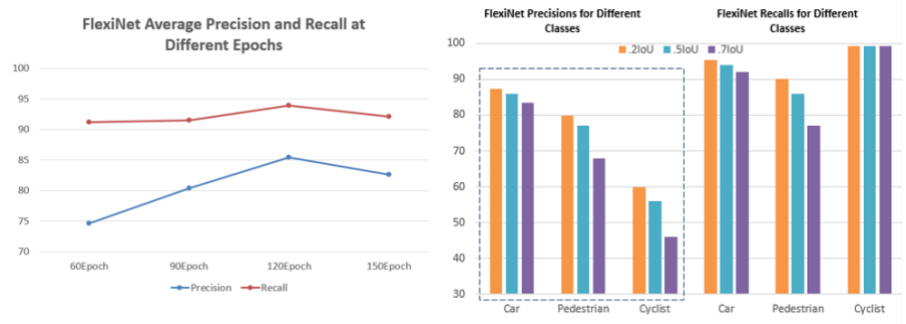
where α is a positive trade-off parameter, v measures the consistency of aspect ratio.

Fig. 7 shows the results obtained by using IoU loss functions. It is clear from these figures that DIoU and CIoU loss functions converge faster than the GIoU loss function, thereby give better accuracy. Fig. 7(b) shows bounding box losses by using different functions while Fig. 7(c) displays the final objectness loss based on the validation dataset. Using DIoU loss function [49] with SGD optimizer [50], FlexiNet achieves the best performance 94.2% mAP@0.5IoU based on the training dataset.



(a) (b) (c)

Fig. 7. Training and validation results based on GIoU, DIoU, and CIoU loss functions (a) mAP@0.5IoU curves based on training dataset (b) the loss curves for the bounding box, and (c) the loss curves based on the validation dataset



(a) (b)

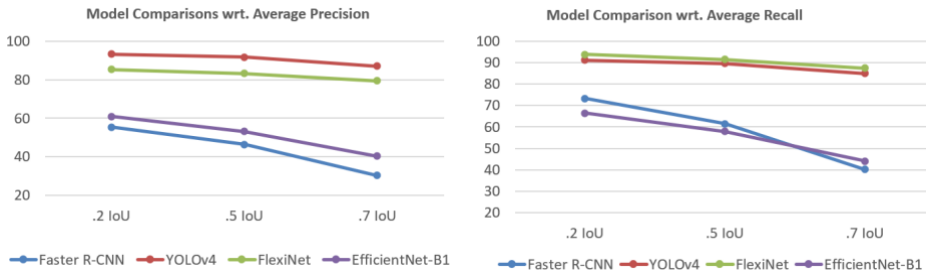
Fig. 8. FlexiNet results based on the dataset (a) Obtained precision and recall rates at 0.5IoU threshold check points (b) Precision and recall rates for the car, pedestrian and cyclist detection by exploiting the best check points.

Fig. 8 shows the FlexiNet results based on the given dataset. In order to check the overfitting of the model, we have tested the model and evaluate the performance at the check points. Fig. 8(a) indicates that 120 epochs model achieves the best performance and provides 85.4% average precision and 93.9% recall rate with 0.5IoU. Fig. 8(b) reveals the results of car, pedestrian, and cyclist detection for road scene perception. The results indicate that the proposed net is efficient in achieving high recall rates that directly indicate a low missing rate, whereas slight low precision also allows a margin of improvement.

4.2 Model Comparisons

Based on the detection results, FlexiNet performance is compared with popular detectors at present on the same platform with different IoU thresholds. Fig. 9 shows that FlexiNet and YOLOv4 [12] outperform Faster R-CNN [10] and EfficientDet-B1 [14]. Fig. 9(b) indicates that FlexiNet achieved the best recall rate and the lowest

missing rate over other models. YOLOv4 proves better in terms of precision that reveals lower false positives detection.



(a) (b)

Fig. 9. The comparison of FlexiNet model with popular methods for visual object detectors (a) Average precision with different IoU thresholds (b) Recall rates with different IoU thresholds

Table 1. shows the overall performance of FlexiNet and other detectors based on the KITTI dataset with complexity at 0.5 IoU threshold for car, pedestrian, and cyclist detection. Recall rate is the prime focus in autonomous driving because it contributes to false negative rates (missing objects), the most crucial parameter for avoiding accidents. The results obtained through FlexiNet show the best recall rate over all three classifiers, whereas YOLOv4 attained a better average precision. On the other hand, Faster R-CNN [10] and EfficientDet-B1 [14] unfold weak performance identifying cyclists, show the difficulty in differentiating cyclists from stand-alone cycles or pedestrians. Fig. 10 illustrates the detection results of cars, pedestrians, and cyclists presented on the road using detection images.

Table 1. The comparisons for FlexiNet model with other popular detectors based on the KITTI dataset with complexity at 0.5 IoU threshold

Model Name	Car (recall%)			Pedestrian (recall%)			Cyclist (recall%)			AP (%)	Saved model size (MB)	fps
	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard			
Faster R-CNN	96.1	76.3	68.7	80.6	66.7	43.6	12.4	9.1	6.8	46.3	**7**	5
EfficientDet-B1	96.4	72.6	68.8	80.0	77.2	46.2	17.5	16.3	5.9	53.1	27	15
YOLOv4	99.4	93.8	90.3	96.4	71.4	74.2	**99.9**	98.6	83.9	**91.8**	244	25
FlexiNet (Our)	**99.7**	**95.0**	**91.2**	**98.3**	**93.3**	**83.2**	99.3	**99.3**	**99.3**	83.2	54	**100**

Fig. 10. The detection results of cars, pedestrians, and cyclists by using the proposed FlexiNet based on the KITTI dataset

Based on the results achieved, one of the most important findings worth notifying is the importance of network balancing. Fig. 5 demonstrates that increasing the width and depth of DNN is not always the best option, in fact, the network needs to be fine-tuned with various depths and widths to optimize its structure. In FlexiNet, multiple factors have been incorporated to improve the performance like the selection of gradient descent optimizers and loss functions. Moreover, as shown in Fig. 8, early stopping is also a good choice if the results start degrading after certain epochs. Albeit FlexiNet remained unsuccessful in achieving the best precision compared to YOLOv4, it gives the lowest false negative rate, which takes account of autonomous vehicles along with four times higher speed. In addition, training the YOLOv4 model requires more time and memory as compared to FlexiNet.

5 Conclusion

In this article, we have proposed an end-to-end flexible net for object detection. The flexibleNet allows the architecture to be selected based on available hardware and dataset. CSPNet [42] is employed as a backbone network with SPP to obtain the complicated features of the input images. The deep net is supported by FPN to detect

multiscale visual objects. The proposed network is fine-tuned by using the gradient descent optimizers and loss functions. FlexiNet yielded the lowest rates of false-negatives in road scene perception based on KITTI dataset [16] with remarkable computing speed. Furthermore, the comparison with other nets shows our proposed model achieves desirable results. In future, we will work for decreasing the rates of false positives for visual object detection [53,54,55,56].

References

[1] World Health Organization (WHO), WHO 2018 Global status report on road safety, Prevention/road_safety_status/2018/en/, www.who.int/violence_injury_. (2018)

[2] Condat, R., Rogozan, A., Bensrhair, A.: GFD-retina: Gated fusion double RetinaNet for multimodal 2D road object detection. In: IEEE International Conference of Intelligent Transport Systems, pp. 1–6 (2020)

[3] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition, USA, pp. 886–893, (2005)

[4] Krizhevsky, A., Sutskever, I., Hinton.: ImageNet classification with deep convolutional neural networks. In: Advanced Neural Information Processing Systems, pp. 1097–1105 (2012)

[5] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2015)

[6] Szegedy, C., Lie.,W., Yangqing., J., Sermanet., P., Reed., S.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)

[7] Szegedy, C. , Vanhoucke, V., Shlens, J.: Rethinking the Inception architecture for computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)

[8] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[9] Huang, G., Liu, Z., Van, L., Maaten, D., Weinberger, K. Q.: Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2261–2269 (2017)

[10] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 28, pp. 91--99 (2015)

[11] Doll, P., Girshick, R.: Mask R-CNN. In: IEEE International Conference on Computer Vision, pp. 2961--2969 (2017)

[12] Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y. M.: YOLOv4: Optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934 (2020)

[13] Liu, W., Anguelov, D., et al.: SSD: Single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37 (2016)

[14] Tan, M., Pang, R., Le, Q. V.: EfficientDet : Scalable and efficient object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 10781--10790 (2020)

[15] Jocher, G. L., Stoken, A., Borovec, J., et. al.: YOLOv5 doi: http://doi.org/10.5281/zenodo.3983579.(2020)

[16] Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research, 32, pp. 1231--1237 (2013)

[17] Tourani, A., Soroori, S., Shahbahrami, A., Khazaee, S., Akoushideh, A.: A robust vehicle detection approach based on Faster R-CNN algorithm. In: IEEE International Conference on Pattern Recognition and Image Analysis, pp. 119–123, (2019)

[18] Jingwei, C., Chuanxue, S., Shixin, S., Silun, P., Da, W., Yulong, S., Feng, X.: Front vehicle detection algorithm for smart car based on improved SSD model. Sensors (Switzerland), 20(16), pp. 1–21, (2020)
[19] Chabot, F., Chaouch, M., Rabarisoa, J., Teuliere, C. and Chateau, T.: Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2040-2049 (2017)
[20] Kim, K.J., Kim, P.K., Chung, Y.S. and Choi, D.H.: Performance enhancement of YOLOv3 by adding prediction layers with spatial pyramid pooling for vehicle detection. In: IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 1–6 (2018)
[21] Redmon, J. and Farhadi, A.: YOLOv3: An incremental improvement. In: arXiv Prepr. arXiv1804.02767, 2018.
[22] Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q. and Cai, B.: An improved YOLOv2 for vehicle detection. Sensors, 18(12), 4272 (2018)
[23] Wang, Y., Liu, Z. and Deng, W.: Anchor generation optimization and region of interest assignment for vehicle detection. Sensors, 19(5), pp. 1089 (2019)
[24] Zhang, X., Gao, H., Xue, C., Zhao, J. and Liu, Y.: Real-time vehicle detection and tracking using improved histogram of gradient features and Kalman filters. In: International Journal of Advanced Robotic Systems, 15(1), 1–9 (2018)
[25] Wei, Y., Tian, Q., Guo, J., Huang, W. and Cao, J.: Multi-vehicle detection algorithm through combining Harr and HOG features. In: Mathematics and Computers in Simulation, pp.130-145 (2019)
[26] Hou, Y.L., Song, Y., Hao, X., Shen, Y., Qian, M., Chen, H.: Multispectral pedestrian detection based on deep convolutional neural networks. In: Infrared Physics and Technology, 94, pp. 69–77 (2018)
[27] Mita, T., Kaneko, T. and Hori, O.: Joint Haar-like features for face detection. In: International Conference on Computer Vision, 2, pp. 1619-1626 (2005)
[28] Ahmed, S., Huda, M.N., Rajbhandari, S., Saha, C., Elshaw, M. and Kanarachos, S.: Pedestrian and cyclist detection and intent estimation for autonomous vehicles: A survey. Applied Sciences, 9(11), 2335. (2019)
[29] Wang, X., Han, T.X. and Yan, S.: An HOG-LBP human detector with partial occlusion handling. In: International Conference on Computer vision, pp. 32-39 (2009)
[30] Hu, Q., Wang, P., Shen, C., van den Hengel, A. and Porikli, F.: Pushing the limits of deep CNNs for pedestrian. IEEE Transactions on Circuits and Systems for Video Technology, 28(6), 1358-1368 (2017)
[31] Shah, M., Kapdi, R.: Object detection using deep neural networks. In: International Conference on Intelligent Computing and Control Systems, pp. 787–790 (2018)
[32] Li, C., Song, D., Tong, R. and Tang, M.: Illumination-aware Faster R-CNN for robust multispectral pedestrian detection. Pattern Recognition, pp.161-171 (2019)
[33] Tumas, P., Jonkus, A. and Serackis, A.: Acceleration of HOG based pedestrian detection in FIR camera video stream. In: Open Conference of Electrical, Electronic and Information Sciences. pp. 1-4 (2018)
[34] Wang, K. and Zhou, W.: Pedestrian and cyclist detection based on deep neural network Fast R-CNN. In: International Journal of Advanced Robotic Systems, pp. 1–10 (2019)
[35] Song, H., Choi, I.K., Ko, M.S., Bae, J., Kwak, S. and Yoo, J.: Vulnerable pedestrian detection and tracking using deep learning. In: International Conference on Electronics, Information, and Communication, pp. 1-2 (2018)
[36] Yang, Z., Li, J. and Li, H.: Real-time pedestrian detection for autonomous driving. In: International Conference on Intelligent Autonomous Systems, pp. 9-13 (2018)
[37] Liu, Y., Cao, S., Lasang, P. and Shen, S.: Modular lightweight network for road object detection using a feature fusion approach. In: IEEE Transactions on Systems, Man, and Cybernetics: Systems, pp. 1–13, (2019)

[38] Li, G., Xie, H., Yan, W., Chang, Y., Qu, X.: Detection of road objects with small appearance in images for autonomous driving in various traffic situations using a deep learning based approach. In: IEEE Access, 8(1), 211164–211172 (2020)
[39] Zagoruyko, S. and Komodakis, N.: Wide residual networks, arXiv preprint arXiv:1605.07146. (2016)
[40] He, K., Zhang, X., Ren, S. and Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778 (2016)
[41] Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018)
[42] Wang, C.Y., Liao, H.Y.M., Wu, Y.H., Chen, P.Y., Hsieh, J.W. and Yeh, I.H.: CSPNet: A new backbone that can enhance learning capability of CNN. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1571–1580 (2020)
[43] Elfwing, S., Uchibe, E. and Doya, K.: Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Networks, 107, pp. 3–11 (2018)
[44] He, K., Zhang, X., Ren, S. and Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904-1916 (2015)
[45] Redmon, J., Farhadi, A.: YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
[46] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition., pp. 936–944 (2017)
[47] Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J. and Yoo, Y.: CutMix: Regularization strategy to train strong classifiers with localizable features. In: IEEE/CVF International Conference on Computer Vision, pp. 6023-6032 (2019)
[48] Kingma, D.P. and Ba, J.: Adam: A method for stochastic optimization. In: International Conference of Learning Representations, pp. 1–15 (2015)
[49] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R. and Ren, D.: Distance-IoU loss: Faster and better learning for bounding box regression. In: AAAI Conference on Artificial Intelligence, 34, No. 07, pp. 12993-13000 (2020)
[50] Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: COMPSTAT, Springer, pp. 177–186 (2010.)
[51] Choi, D., Shallue, C.J., Nado, Z., Lee, J., Maddison, C.J. and Dahl, G.E.: On empirical comparisons of optimizers for deep learning. arXiv preprint arXiv:1910.05446. (2019)
[52] Wilson, A.C., Roelofs, R., Stern, M., Srebro, N. and Recht, B.: The marginal value of adaptive gradient methods in machine learning. arXiv preprint arXiv:1705.08292. (2017)
[53] Mehtab, S., Yan, W.: FlexiNet: Fast and accurate vehicle detection for autonomous vehicles-2D vehicle detection using deep neural network. ACM ICCCV, pp. 43-49 (2021)
[54] Mehtab, S., Yan, W., Narayanan, A.: 3D vehicle detection using cheap LiDARs and RGB images. IEEE IVCNZ (2021)
[55] Yan, W.: Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics. Springer (2019)
[56] Yan, W.: Computational Method for Deep Learning: Theoretic, Practice and Applications. Springer (2019)