FlexiNet: Fast and Accurate Vehicle Detection for Autonomous Vehicles

Sabeeha Mehtab
Auckland University of Technology, Auckland
sabeeha.mehtab@aut.ac.nz

Farah Sarwar
Auckland University of Technology, Auckland
farah.sarwar@aut.ac.nz

Weiqi Yan
Auckland University of Technology, Auckland
weiqi.yan@aut.ac.nz

ABSTRACT

Autonomous vehicle has come to reach on the road; however accurate road perception in real-time is one of the crucial factors towards its success. The greatest challenge in this direction includes occlusion, truncation, lighting conditions, and complex backgrounds. In order to improve the accuracy and detection speed of vehicle detection, a dynamic scaling network is proposed that assists in constructing a balanced shape neural network to achieve optimum accuracy with minimal hardware. The net architecture is influenced by YOLOv5 and is composed of Cross-Stage Partial Network (CSPNet) as its backbone. In order to go even further, we have proposed an auto-anchor generating method that makes the network suitable for any datasets. Our neural network is fine-tuned by using activation, loss, and optimization functions so as to get the optimum results. Our experimental results demonstrate that the proposed net provides comparable performance of YOLOv4 and Faster R-CNN based on KITTI dataset as the benchmark.

CCS CONCEPTS

• **Computing methodologies**; • **Machine learning**; • **Machine learning approaches**; • **Neural networks**;

KEYWORDS

Vehicle detection, .Object detection, .Autonomous driving, .Self-driving car, .Deep neural network, .YOLOv4, .YOLOv5

ACM Reference Format:
Sabeeha Mehtab, Farah Sarwar, and Weiqi Yan. 2021. FlexiNet: Fast and Accurate Vehicle Detection for Autonomous Vehicles. In *2021 4th International Conference on Control and Computer Vision (ICCCV'21), August 13–15, 2021, Macau, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3484274.3484282

1 INTRODUCTION

Taken into consideration the increase in road accidents [1] and the involvement of human's energy in terms of time and exertion, autonomous vehicle (AV) has become a necessity in today's world. One of the most important requirements for an AV is to perceive the road scene accurately in real-time and detect vehicles precisely. However, it becomes challenging due to heavy traffic, complex backgrounds of roads, and extreme weather conditions. Vehicle detection using aerial images has gained tremendous attention with its applications in traffic monitoring, surveillance, and military applications [2]. The aerial view of vehicles shows less or no occlusion and shadows as compared to the ground-based images, which are needed for an AV. Thus, extensive work is needed for vehicle detection with ground-based images to accelerate AV success. The algorithms in computer vision are mostly based on a feature extractor to extract useful features like edges, corners, motif, contours, gradients. Histogram of Oriented Gradient (HOG) [3] that works as a feature extractor, has been used in plenty of detection models in combination with classifiers like Support Vector Machine (SVM) or Deformable Part-Based Model (DPM) for vehicle detection [4]. However, in the case of high occlusion and illumination changes, these handcrafted-feature-based algorithms exhibit difficulties in accurate vehicle detection [5]. With the advent of fast processing GPUs and big data storage devices, the detection models have adopted Deep Neural Network (DNN) as self-learning algorithms to showcase promising results. There are generally two types of DNN approaches that are used for object detection: 1) Region-based detection methods that firstly filter positive and negative regions, then are applied to object detection; 2) The end-to-end detection methods scan the input images only once and detect the objects by using single neural network. The end-to-end detectors are much faster, but export less accurate results compared to the region-based object detection approaches [6].

Our goal in this paper is to propose a DNN net that detects vehicles by using input images with high accuracy in real-time and conduct a comparative analysis with the existing state-of-the-art networks. In this paper, a PyTorch-based light framework is proposed that is influenced by YOLOv5 architecture [7]. The contributions of this paper are summarized as follows:

- Dynamic scaling neural network is proposed to generate the desired structure that achieves the best result for vehicle detection by using a single GPU.
- CSPNet-based [8] network is utilized that strengthens the network ability without increasing computational overhead.
- An auto-anchor generator is implemented that makes the network generalization for any datasets.
- In order to achieve the best outcome, the network is fine-tuned by using specified loss, activation, and optimization functions.
- The proposed neural network is trained and tested by using benchmark datasets like KITTI [9] and Waymo [10] that have complex, rich, and diverse images for vehicle detection.

ICCCV'21, August 13–15, 2021, Macau, China

ACM ISBN 978-1-4503-9047-7/21/08…$15.00
https://doi.org/10.1145/3484274.3484282

In this paper, we introduce the related work in Section 2, our methodology is depicted in Section 3, the results are shown in Section 4, our conclusion is drawn in Section 5.

2 RELATED WORK

A great deal of vehicle detection models in computer vision and DNN, have been proposed in recent years for vehicle detection. LiDAR, which provides 3D coordinates in the form of point clouds and multiple models of cameras (e.g., monocular, stereo, infrared) that acquires images with rich textures, has been chosen predominantly as imaging sensors [11].

Using RGB images, Zakaria et al. [1] proposed a HOG variant in combination with nonlinear SVM. The model is based on compass gradients to collect the features from multiple angles unlike horizontal and vertical angles only. Farag et al. experimented with multiple color schemes relying on low-cost hardware, using a HOG descriptor and SVM classifier with a sliding window over the selected region of interests (ROIs) [4]. Contextual information like the relative distance was applied, including the key features of vehicles like wheels and headlights for vehicle detection [12].

In the series of region-based detectors, Faster R-CNN [13] was a big leap over the existing approaches like R-CNN [14] and Fast R-CNN [15]. Faster R-CNN has been used in variant forms for vehicle detection [16–18]. Despite being given reasonable accuracy, Faster R-CNN could not satisfy the requirements for real-time vehicle detection and needs multiple GPUs in the training time.

The end-to-end networks are fast in the training time which has been exploited broadly. YOLOv2 was taken in combination with k-means++ clustering for selecting various anchor boxes [19], however, the results were not compared by using any benchmark datasets, high-cost hardware was utilized to conduct the experiments. Tiny YOLOv3 detection model with k-means clustering was proposed [5] and generated an accuracy 91.03% based on KITTI dataset. SSD structure was modified [20] by combining inception block and feature fusion layers in the original network, which obtained 92.18% mAP based on KITTI dataset. In order to adjust the limitations of region proposal-based and end-to-end approaches, HybridNet [21], which joined a single network with proposal-based detection, achieved up to 87.91% precision based on KITTI dataset with the detection rate at 22 video frames per second (fps).

Pertaining to the depth estimate of vehicles, LiDAR has been taken into account with digital cameras. LiDAR point-cloud-projected images and camera images have been fused [17, 18, 22] together before feeding into the detection network. Aggregate View Object Detection (AVOD) network [17] was also tested outdoors under the bad weather conditions, the results showed 85.44% mAP on moderate complexity KITTI dataset. On the other hand, Frustum PointNet [23] generates region proposals by utilizing 3D point clouds and images to increase accuracy. Hereinafter, the point clouds are split into multiple segments and got processed in the PointNet. The network generates features based on independent segments and thus limits the detection performance without showing any relationship regarding neighborhood [11].

The latest developments in the field of deep learning for object detection are YOLOv4 [6] and YOLOv5 [7]. YOLOv4 covers the previous inaccuracies whilst preserving the speed of object detection. To the best of our knowledge, there is no research work that has been published for vehicle detection by using YOLOv4 with ground-based images yet, so does YOLOv5 which has been released by Ultralytics on Github only.

3 METHODOLOGY

In this paper, a flexible network named FlexiNet is proposed by using DNN to detect vehicles by using ground-based images. As shown in Figure 1 (a), FlexiNet architecture is composed of two main modules: The backbone part and the visual object detection part. The network is drawn with a dynamic scaling approach based on the promising CSPNet framework [8]. At the detection layer, an auto-anchor generating method is proposed that makes this network generalization to be suitable for various datasets.

3.1 Backbone Network

The backbone network, which is responsible for feature extraction, plays a pivot role in accurate object detection. The performance of the backbone network is enhanced by increasing receptive field and network complexity [24]. Increasing the depth of neural network to a large extent seems promising to extract complex features of visual objects, though does not always lead to better accuracy, and may trigger vanishing gradient problem. In our experiments, it is found that ResNet-50 network generates better accuracy than ResNet-152 does, in spite of having a greater number of layers. On the other hand, increasing the width of channels aids in finding fine-grained features. However, a balance between the width and depth of the network is required to get successful results.

3.1.1 ***Dynamic Network Scaling.*** Dynamic scaling is a promising approach to find the minimum tradeoff between width and depth of the network for achieving optimum accuracy. Dynamic scaling has been employed for finalizing network structure, e.g., EfficientDet yields multiple variants based on various shapes of the network [24]. In the proposed model, the network shape is drawn at the run time by using variable depth and width and finalized based on results achieved.

In order to keep the image residual information intact with very deep neural networks, various networks have been proposed based on skip-connection approaches, like ResNet, ResNetXt, DenseNet and CSPNet [24, 25]; however, they differ in the network topology and the fusion approach for feature maps. Recently related model CSPNet [8] splits the path of gradient flow in two different streams. One branch towards the "Dense block" concatenates the information flows between convolutional layers, another branch carries the remaining part of the initial feature maps directly towards the partial transition block as shown in Fig. 1 (b). CSPNet has been proven to converge faster with no extra storage [8].

In FlexiNet, the underlying backbone architecture comprises of CSPNet blocks with alternate convolutional blocks that are responsible for changing the feature map sizes. The number of channels and the number of layers in each CSPNet blocks are determined by using the values of depth and width of the proposed network.

Traditional neural networks for object detection keep the constraint of size fixed images to deal with the layers that carry out classification [6]. However, regarding on-road vehicle detection,

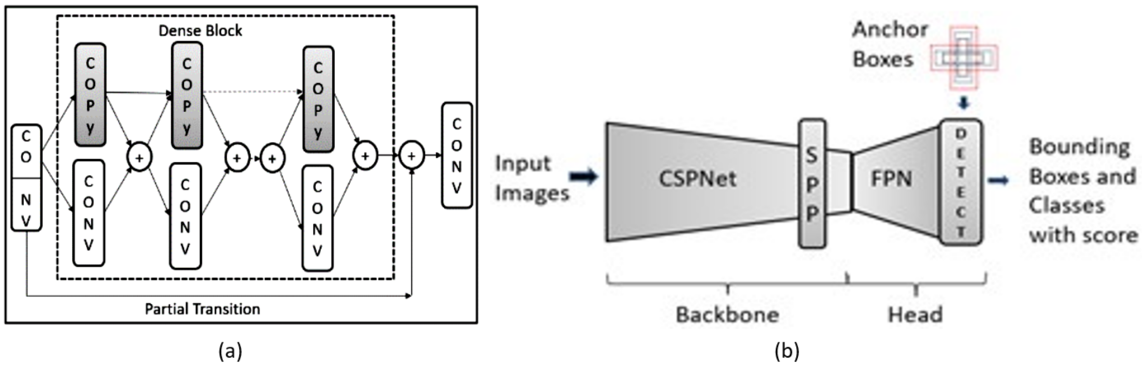

Figure 1: The Architecture (a) CSPNet works as a basic building block in FlexiNet with dynamic scaling. (b) The complete FlexiNet consists of two modules: 1) The backbone network is composed of multiple CSPNet blocks with SPP layer; 2) The detection part is based on FPN network to capture various scale objects using anchors.

it is difficult to crop the images so as to achieve the required aspect ratio and scale. In order to deal with this issue, the use of a Spatial Pyramid Pooling (SPP) [26] layer is proposed at the end stage of the backbone network. The SPP layer concatenates the feature extracted from three intermediate convolutional layers and generates a fixed-length representation before feeding them into the detection network.

3.2 Detection Network

Conventional DNN detection models generate the final predictions based on the feature maps received from the final convolutional layers, however, this stage may lose highly grained features. This problem is handled with success by using a multistage detection network. In our network, the features are extracted from three scales by using Feature Pyramid Network (FPN) with the anchors as discussed in YOLOv3 [27] and YOLOv4 [6].

FlexiNet takes use of three shapes based on vehicle pose and three scales of anchors to estimate bounding boxes of vehicles from multiple viewpoints. However, multistage detection leads to multiple bounding boxes of the same object. Extra bounding boxes are eliminated by using non-max suppression that retains only the bounding box with the highest confidence score.

3.2.1 Auto-anchor Generator. To make the network generalized, our model provides the flexibility to change anchor size based on the training set information. The proposed algorithm (Algorithm 1) firstly computes the average size of ground-truth bounding boxes (Avg_GT_BB) and aspect ratio at three different scales by using k-means clustering algorithm [28]. Taken into account of the initialization, the dimension of anchor boxes (AnchorSize) and aspect ratio are refined iteratively based on the difference from the average ground truth (GT) bounding box (BB) until no further change is needed. This comparison is fulfilled at three different scales [28]. This feature makes the algorithm suitable for different datasets without manually altering anchor boxes.

Algorithm 1 Auto-Anchor Generation Algorithm

```
Input: data GT_BBs, AnchorSize_i, size n × m
Compute Avg_GT_BB_i using k-means clustering algorithm
repeat
  Initialize no_Change = true
    for i = 1 to 3, do
        if  AnchorSize_i < Avg_GT_BB_i, then
              increase AnchorSize_i
              noChange = false
        elseif  AnchorSize_i > Avg_GT_BB_i, then
              decrease AnchorSize_i
              noChange = false
        endif
    endfor
until noChange = true
Output: AnchorSize_i
```

3.3 Activation Functions

With regard to efficient gradient flow, the activation function plays a key role in training dynamics and network performance. Rectified Linear Unit (ReLU) [29] has been a classic choice for dealing with the vanishing gradient problem effectively that also offers low computational cost. However, ReLU suffers from gradient information loss by collapsing the negative inputs to zero. Leaky ReLU [30], Flexible ReLU (FReLU) [31], Swish [32], Mish [33], and Hardswish [34] cover these problems in a more efficient way. These activation functions have been investigated individually to evaluate the methods based on the accuracy of FlexiNet.

- **FReLU**: The ReLU output is adjusted by using a rectified point to capture negative information and provide zero-like property.

$$f(x) = \begin{cases} x + b_l & if\ x > 0 \\ b_l & f\ x \leq 0 \end{cases} \quad (1)$$

where b_l is the layer-wise learnable parameter.

- **Swish**: It is unbounded above and bounded below like ReLU, whereas, smooth and non-monotonic. It is represented by equation 2)

$$s(x) = \frac{x}{1 + e^{-x}} \tag{2}$$

- **Mish:** This is a nonmonotonic, continuously differentiable activation function which is inspired by the self-gating property of Swish [33].

$$m(x) = x \cdot tanh\left(\ln\left(1 + e^{x}\right)\right) \tag{3}$$

- **Hardswish:** This is a piece-wise linear analog activation function [34] and specially designed for quantization network. Hardswish is represented by equation 4).

$$h(x) = \begin{cases} 0 & if\ x \leq -3, \\ x & if\ x \leq\ +3, \\ x \cdot \frac{x+3}{6} & otherwise \end{cases} \tag{4}$$

3.4 Optimization Functions

As a part of parameter adjustment and loss reduction, two optimization functions are investigated. Stochastic Gradient Descent (SGD) given in equation 5) is the best way for minimizing the loss that promotes the changes of network parameters on each training epoch [35]. On the other hand, adaptive momentum estimation (Adam) given in equation 6) is another optimizer that computes the adaptive learning rates for each parameter [35]. For network parameters θ, and learning rate η:

$$\theta =\ \theta - \eta \cdot \nabla_{\theta} J\left(\theta;\ x_i; y_i\right) \tag{5}$$

where, for SGD, x_i and y_i represent training example and label information of iteration i, and $\nabla_{\theta} j$ denotes the objective function.

$$\theta =\ \theta - \frac{\eta}{\sqrt{\hat{v}_t} + \in} \hat{m}_t \tag{6}$$

where, for Adam,$\hat{m}_t$, and $\hat{v}_t$ are the estimates of mean and variance. $\in$is the summation coefficient.

3.5 Loss Functions

Visual object detection in computer vision is being resolved through dealing with the bounding box regression and classification problems. Our model uses different loss functions for class probability and bounding box regression, respectively. The final loss of each iteration is the sum of both constituent losses that eventually tend to maximize the accuracy.

Binary Cross-Entropy with Logits Loss (BCEL) that is supported by PyTorch is utilized for calculating class probability and objectness score, which combines binary cross-entropy and sigmoid function in a single class and is thus much stable numerically.

Regarding bounding box regression, three Intersection over Union (IoU) losses have been considered to find the most suitable one, as IoU loss plays a prime role in the final non-maximum suppression (NMS) of bounding boxes. Loss functions under consideration are Generalized-IoU (GIoU) [36], Distance-IoU (DIoU) [37], and Complete-IoU (CIoU) [37] losses. Unlike basic IoU, the focus of these loss functions is not only on overlapping regions but also on other non-coincident regions, which better reflect the overlapping region between the predicted and ground truth bounding boxes. These loss functions are stated as follows with A and B representing ground truth and the predicted bounding boxes, respectively.

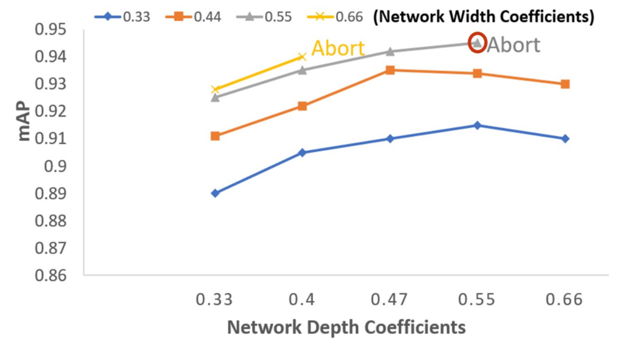

Figure 2: The performance of the proposed scaling network for a set of its widths and depths based on FlexiNet architecture. Each curve shows the width, each point represents a network with different depth. FlexiNet achieved the best mAP at .55 depth and .55 width based on the training dataset.

- **GIoU** loss is formulated in equation 7)

$$\mathcal{L}_{GIoU} = \frac{|A \cap B|}{|A \cup B|} - \frac{|C \backslash (A \cup B)|}{|C|} \tag{7}$$

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

- **DIoU** loss function is defined as

$$\mathcal{L}_{DIoU} = 1 - \frac{|A \cap B|}{|A \cup B|} + \frac{\rho^2(a, b)}{c^2} \tag{8}$$

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

- **CIoU** loss function is defined as

$$\mathcal{L}_{CIoU} = 1 - \frac{|A \cap B|}{|A \cup B|} + \frac{\rho^2(a, b)}{c^2} + \alpha v \tag{9}$$

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

4 RESULTS AND EVALUATIONS

Regarding experimental evaluations of the proposed network, we have adopted the benchmark KITTI dataset [9]. In this set, "Car" and "Van" are categorized as the classes, however, the main target of this research project is related to vehicle detection, so we have combined them in the same class as "Vehicle". As the relevancy, 2,000 random images are used for training and validation with a proportion 8:2, respectively.

The proposed network was trained based on a single TESLA X GPU with total_memory approximating to15GB, fixing a batch size of 16. All training and validation images have the resolution 640 × 640. Amid the training process, variation in functions like activation function, optimization function, and loss function is employed to observe the performance of the network. All these parameters are explained briefly. Followed the trend in object detection [13, 38], mean average precision (mAP) with 50% IoU metric is taken as the primary metric to validate the results.

Table 1: The influence of various activation functions on mAP

Activation Function	mAP@0.5IoU(%)	Recall(%)
FReLU	93.4	95.4
Swish	92.2	91.7
Mish	92.9	92.2
Hardswish	**94.5**	**95.4**

Figure 2 represents the results of the dynamic scaling network. Interestingly, the object detection results increase with deeper width. On the other hand, as the depth of the network is growing, initially, the network outcomes are improved at a fast pace but after going further deep, the performance starts degrading and eventually leads to abortion of the experiment because of the high GPU storage requirement. In Figure 2, FlexiNet attained the best performance at .55 depth and .55 width with assigned training dataset, so all experiments ahead were run on this network shape.

In the probe of activation functions on FlexiNet, we have found FReLU [31] and Hardswish [34] outperformed other functions as summarized in Table 1

We have also investigated the effects of multiple IoU loss functions for bounding box regression based on FlexiNet and observed that DIoU loss converges faster than CIoU and GIoU as shown in Figure 3(a) which gives 1% better mAP based on KITTI dataset. Pertaining to vehicle detection, 1% growth of precision is a remarkable achievement for avoiding road accidents. Figure 3(b) depicts the mAP curves of various IoU losses.

In the investigation of the best optimizer for our dataset, Adam has demonstrated faster convergence at the initial stages; however, SGD proves to be a better choice for giving the maximum accuracy whilst keeping a slow learning rate as shown in Fig. 3(c). The best results are obtained at learning rate 0.01, momentum 0.937, weight decay 0.0005 with SGD optimizer [35].

Table 2 shows the comparison of FlexiNet with popular networks based on the same platform. In Table 2, it is clear that SSD [39], YOLOv3 [27], and EfficientDet-B2 [24] remained unsuccessful to give promising results based on KITTI dataset due to its challenging conditions. Faster R-CNN achieves 82.9% mAP based on the same platform; however, its speed is much slower than YOLOv4 and FlexiNet, which is not suitable for real-time vehicle detection.

The respective loss curves are shown in Figure 4. With FlexiNet converging to 1.8% loss in 600 epochs, YOLOv4 achieved a comparable loss in 2,600 epochs, and Faster R-CNN shows continuous improvements in the loss but could not go beyond 12%. Between YOLOv4 and FlexiNet, there is not much variance in the final precision; however, there is a rich assortment of computational difficulties while executing YOLOv4. We have observed that FlexiNet generated 94.5% mAP and 95.4% recall after 4,50 epochs, whereas YOLOv4 gives comparable accuracy after 1,300 epochs. YOLOv4 demands much higher storage as compared to the proposed network. While executing YOLOv4, batch size could not be increased by more than 16 with image resolution 640×640 based on our 15GB GPU. On the other hand, FlexiNet runs efficiently with 64 batch size of the dataset.

Figure 5 shows the detection results of FlexiNet based on KITTI test images. To verify the performance of FlexiNet, we have also conducted FlexiNet training and detection based on Waymo dataset [10]. Despite night and rainy images of Waymo dataset, FlexiNet resulted in 97.2% mAP that is even better than the KITTI dataset-based performance. One of the reasons for this is the higher occlusion and truncation of KITTI over Waymo dataset. Figure 6 shows FlexiNet detection results based on Waymo dataset.

5 CONCLUSION

In this paper, we have proposed a flexible network, the network shape was adjusted to get the optimum results. FlexiNet has achieved 94.5% mAP@0.5IoU by using DIoU loss and Hardswish activation functions with SGD optimizer based on the benchmark KITTI dataset. To the best of our knowledge, it attains state-of-the-art performance for vehicle detection. The proposed auto-anchor generating method makes this network generalize to any datasets. In future, we would like to extend this work for 3D vehicle detection

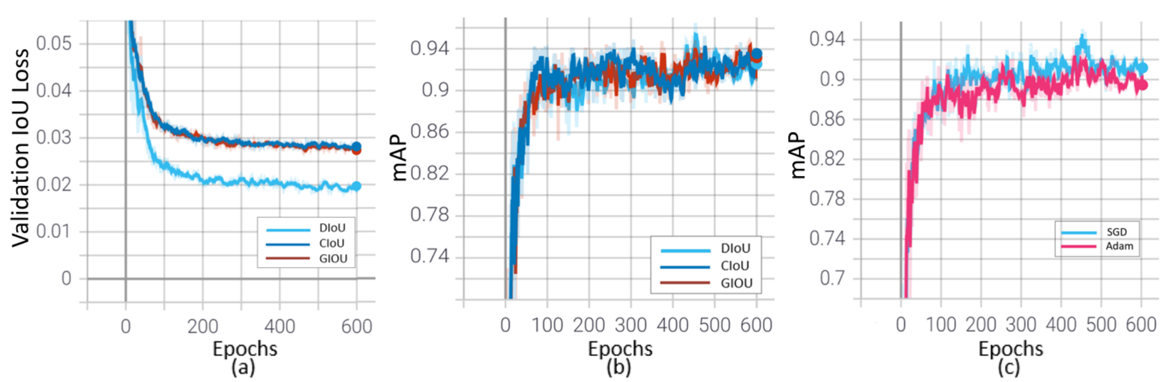

Figure 3: Our results (a) Validation loss curves with GIoU, CIoU, and DIoU functions, (b) Network performance with respect to different IoU losses, (c) Network performance with SGD and Adam optimizers

Table 2: Comparisons of the proposed FlexiNet with the state-of-the-art detection methods

Model	mAP@0.5IoU (%)	Recall (%)	FPS
EfficientDet-B2	31.9	32.1	8
Faster R-CNN	82.9	55.3	7
SSD	22.2	12.7	40
YOLOv3	70.3	22.9	35
YOLOv4	92.5	-	25
FlexiNet (our)	**94.5**	**95.4**	**72**

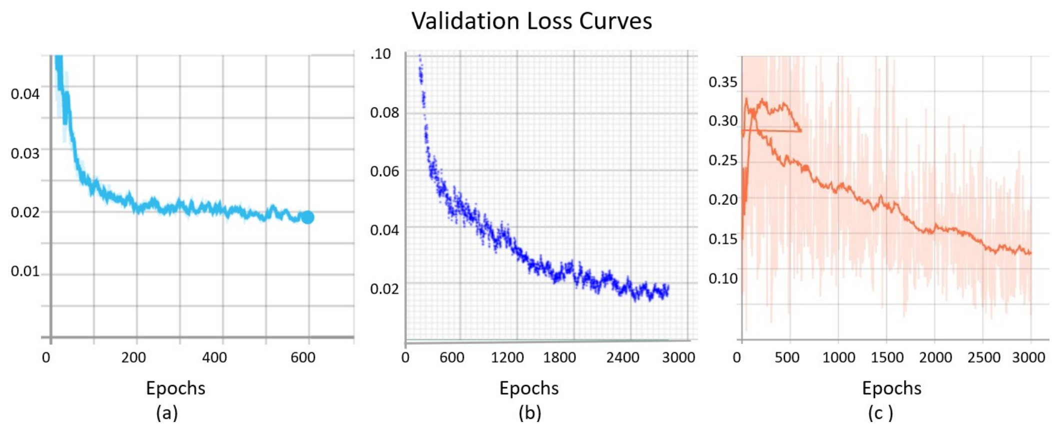

Figure 4: FlexiNet validation loss analysis with multiple networks, (a) FlexiNet converges to 1.8% loss at 600 epochs; (b) YOLOv4 converges to 1.8% loss at 2600 epochs; (c) Faster R-CNN converges to 12% loss at 3000 epochs.

Figure 5: Vehicle detection results based on test images of KITTI dataset by using FlexiNet

Figure 6: Vehicle detection results based on test images of Waymo dataset by using FlexiNet

that will be much relevant for getting an exact estimate of the shape and distance of the front lying vehicles [40–43].

REFERENCES

[1] Zakaria, Y., Ali, M.A., Abd El Munim, H.E., Yousef, A.H., Ghoneima, M., Hammad, S. 2018. A novel vehicle detection system. In Proceedings of International Conference on Computer Engineering and Systems. pp. 127–131.

[2] Mo, N., Yan, L. 2020. Improved Faster RCNN based on feature amplification and oversampling data augmentation for oriented vehicle detection in aerial images. Remote Sensing 12(16), 2558

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

[4] Farag, W. 2020. A lightweight vehicle detection and tracking technique for advanced driving assistance systems. Intelligent & Fuzzy Systems 39(3), 2693– 2710

[5] Wang, X., Wang, S., Cao, J., Wang, Y. 2020. Data-driven based Tiny-YOLOv3 method for front vehicle detection inducing SPP-net. IEEE Access 8, pp.110227–110236

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

[7] Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu,L., Laughing, tkianai, Hogan, A., lorenzomammana, yxNONG, AlexWang1900, Diaconu, L., Marc, wanghaoyang0106, ml5ah, Doug, Ingham, F., Frederik, Guilhen, Hatovix, Poznanski, Jake an Fang, J., Yu, L., changyu98, Wang, M., Gupta, N., Akhtar, O., PetrDvoracek, Rai, P.: ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements (Oct 2020). https://doi.org/10.5281/zenodo.4154370, https://doi.org/10.5281/zenodo.4154370

[8] Wang, C.Y., Liao, H.Y.M., Wu, Y.H., Chen, P.Y., Hsieh, J.W., Yeh, I.H. 2020. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 390–391

[9] Geiger, A., Lenz, P., Stiller, C., Urtasun, R. 2013. Vision meets robotics: The KITTI dataset. Robotics Research 32(11), pp. 1231–1237

[10] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo,J., Zhou, Y., Chai, Y., Caine, B., *et al.*2020. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2446–2454

[11] Arnold, E., Al-Jarrah, O.Y., Dianati, M., Fallah, S., Oxtoby, D., Mouzakitis, A. 2019. A survey on 3D Object detection methods for autonomous driving applications. IEEE Transactions on Intelligent Transportation Systems 20(10), pp. 3782–3795.

[12] Gählert, N., Wan, J.J., Weber, M., Zöllner, J.M., Franke, U., Denzler, J. 2019. Beyond bounding boxes: Using bounding shapes for real-time 3D vehicle detection from monocular RGB images. In Proceedings of IEEE Intelligent Vehicles Symposium (IV). pp. 675– 682.

[13] Ren, S., He, K., Girshick, R., Sun, J. 2016. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6), pp. 1137–1149

[14] Girshick, R., Donahue, J., Darrell, T., Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587

[15] Girshick, R. 2015. Fast R-CNN. In Proceedings of IEEE International Conference on Computer Vision. pp. 1440–1448

[16] Manana, M., Tu, C., Owolawi, P. A. 2018. Preprocessed Faster RCNN for vehicle detection. In Proceedings of IEEE International Conference on Intelligent and Innovative Computing Applications pp. 1-4.

[17] Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S. L. 2018. Joint 3d proposal generation and object detection from view aggregation. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1-8.

[18] Beltra´n, J., Guindel, C., Moreno, F.M., Cruzado, D., Garcia, F., De La Escalera, A. 2018. BirdNet: A 3D object detection framework from LiDAR information. In Proceedings of International Conference on Intelligent Transportation Systems. pp. 3517– 3523.

[19] Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q., Cai, B. 2018. An improved YOLOv2 for vehicle detection. Sensors 18(12)

[20] Cao, J., Song, C., Song, S., Peng, S., Wang, D., Shao, Y., Xiao, F. 2020. Front vehicle detection algorithm for smart car based on improved SSD model. Sensors 20(16)

[21] Dai, X. 2019. HybridNet: A fast vehicle detection system for autonomous driving. Signal Processing: Image Communication 70, pp. 79–88

[22] Yu, S.L., Westfechtel, T., Hamada, R., Ohno, K., Tadokoro, S. 2017. Vehicle detection and localization on bird's eye view elevation images using convolutional neural network. In Proceedings of IEEE International Symposium on Safety, Security and Rescue Robotics. pp. 102–109

[23] Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J. 2018. Frustum PointNets for 3D object detection from RGB-D data. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 918–927

[24] Tan, M., Pang, R., Le, Q. 2020. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781-10790.

[25] Rezaei, M., Azarmi, M.2020. DeepSOCIAL: Social distancing monitoring and infection risk assessment in COVID-19 pandemic. Applied Sciences 10(21)

[26] He, K., Zhang, X., Ren, S., Sun, J. 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(9), 1904–1916

[27] Redmon, J., Farhadi, A. 2018. YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767

[28] Bradley, P.S., Fayyad, U.M. 1998. Refining initial points for k-means clustering. In Proceedings of International Conference on Machine Learning. pp. 91–99.

[29] Nair, V., Hinton, G.E. 2010. Rectified linear units improve restricted Boltzmann machines. In Proceedings of International Conference on Machine Learning. pp. 807–814.

[30] Maas, A.L., Hannun, A.Y., Ng, A.Y. 2013. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of International Conference on Machine Learning. vol. 30, pp. 1–6.

[31] Qiu, S., Xu, X., Cai, B. 2018. FReLU: Flexible rectified linear units for improving convolutional neural networks. In Proceedings of International Conference on Pattern Recognition. pp. 1223–1228.

[32] Ramachandran, P., Zoph, B., Le, Q.V.2017. Searching for activation functions. arXiv preprint arXiv:1710.05941

[33] Misra, D. 2019. Mish: A self regularized non-monotonic neural activation function. arXiv preprint arXiv:1908.08681

[34] Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., *et al.* 2019. Searching for MobileNetV3. In Proceedings of IEEE/CVF International Conference on Computer Vision. pp. 1314–1324

[35] Ruder, S. 2016. An overview of gradient descent optimization algorithms. arXiv preprintarXiv:1609.04747

[36] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 658–666

[37] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D. 2020. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of AAAI Conference on Artificial Intelligence. vol. 34, pp. 12993–13000

[38] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. 2016. You only look once: Unified, real-time object detection. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788

[39] [39] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. 2016. SSD: Single shot multibox detector. In Proceedings of European Conference on Computer Vision. pp. 21–37. Springer

[40] [40] Yan, W. 2021. Computational Methods for Deep Learning Theoretic, Practice and Applications, Springer.

[41] [41] Yan, W. 2019. Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics, Springer.

[42] [42] Gu, Q., Yang, J., Kong, L., Yan, W., Klette, R. 2017. Embedded and real-time vehicle detection system for challenging on-road scenes. Optical Engineering 56 (6), 063102

[43] [43] Liu, X., Nguyen, M., Yan, W. 2019. Vehicle-related scene understanding using deep learning. Workshop AAPS 1 (1), 1-12