An Efficient Intersection Over Union Loss Function for 3D Object Detection

Sazan Mohammed[1,3], Mohd Zulhakimi Ab Razak[1], Abdul Hadi Abd Rahman[2]
[1]Institute of Microengineering and Nanoelectronics (IMEN), Universiti Kebangsaan Malaysia, Malaysia
[2]Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia
[3]Department of Automotive Technology, Erbil Technology College, Erbil Polytechnic University, Erbil, Iraq
Email: p103643@siswa.ukm.edu.my, zul.hakimi@ukm.edu.my, abdulhadi@ukm.edu.my

***Abstract*— In the area of computer vision, object detection using convolutional neural networks (CNNs) has become quite a popular procedure because of their effectiveness and simplicity. The loss function has a great influence on the average accuracy value of the CNN model's detector results. An improved 3D efficient intersection over union (EIoU) loss function is proposed to improve the localization accuracy. The diagonal distance between bounding boxes' corners and centers, with the dimensional change in boxes' geometry sides, are used for matching between the 3D predicted bounding box with the 3D ground truth bounding box. By taking the geometry sides change and diagonal distance between the 3D predicted and 3D ground truth boxes, a great influence on the localization accuracy is generated. For the network model, the localization accuracy is improved because of the strength of diagonal distance and geometry sides' adjustment. Utilizing the one-stage object detector 3D YOLO v4 and applying the 3D EIoU experimentally on the KITTI dataset, the findings demonstrate the effectiveness of the 3D EIoU in improving the accuracy of localization for the network model. Compared with 3D GIoU, the proposed EIoU enhances the average precision (AP) value by 0.24% and AP70 by 1.179% in the car class, AP by 0.578%, and AP55 by 6.022% in cyclist class, and AP by 0.1531% and AP30 by 2.548% in pedestrian class.**

Keywords—object detection, localization accuracy, 3D bounding box regression, 3D loss function, deep learning

I. INTRODUCTION

Convolutional neural networks (CNNs) are used mainly in object detection. When using CNN for regression or classification processes, the loss function estimates the degree value of inconsistency between the model's predicted and actual values. The main objective of model training is to minimize the loss function by using the optimization technique to calculate the model parameters. The loss function affects the working action of different object detectors, where the loss function determines the model's optimal value.

Bounding box regression and classification are two subsections of the loss function. For object's localization and identification, it is essential to calculate the bounding box regression loss. In recent years, deep CNNs showed high-quality performance in predicting the 2D bounding box of candidate objects. Despite their state-of-the-art performance, those detectors have inherent limitations for specifying the object's geometry, which misled the localization and classification process. For example, using a 2D bounding box in the autonomous driving field will increase the misdetection process [1]. Knowing two dimensions for cars on the road is insufficient to measure the distance between vehicles, therefore there is a significant need to know the 3D dimensions for objects increased in the autonomous driving field. In contrast to the 2D box, which only needs five variables, the regression of 3D bounding boxes needed seven variables, including the location points (x, y, z), size (w, l, h), and the orientation angle θ.

In the object detection process, bounded box regression is an important technique used to calculate the object localization performance. On the other hand, most old loss functions for bounding box regression have two specific drawbacks : (i) Both 3D IoU and L_n-norm loss functions, were inefficient to describe the objective of bounding box regression, which made the convergence between prediction and target boxes slow with inaccurate regression results [2]; (ii) the imbalance problem in bounding box regression ignored by most of the loss functions, where a large number of anchor boxes, that have small overlaps with the target boxes made most of the contribution to the optimization of bounding box regression [3].

We proposed an efficient 3D IoU loss (EIoU) to obtain a faster convergence speed and better regression results to solve the above issues. The 2D EIoU has a penalty term for two sides in previous work [4]. On the other hand, in this study, we add an additional side dimension to improve the bounding regression outcome results. Taking the three sides of the 3D box is an efficient way to consider the slight overlap of anchor boxes and get the correct results from the bounding regression process. The main contributions of this paper are:

(1) An efficient 3D Intersection Over Union (EIoU) loss equation is designed, which explicitly finds the discrepancies of three geometric elements in the bounding box regression, i.e., the overlap area, the center point distance, and the box sides' change.
(2) Evaluate the performance of 3D EIoU loss performance when it incorporates into a 3D YOLO v4 object detection model, achieving notable performance.

The paper is arranged as follows. The related work is explained in Section II. The comparison methodology of the loss functions' value in different situations is discussed in Section III. The experiment and exploration for object detection illustrated in Section IV, followed by conclusion.

II. RELATED WORK

The previous 3D object detecting techniques and 3D IoU loss function for bounding box regression are discussed in this section. With rapid development in many fields of automation, manufacturing, and so many others, 3D object detecting has become more and more important. An RGB

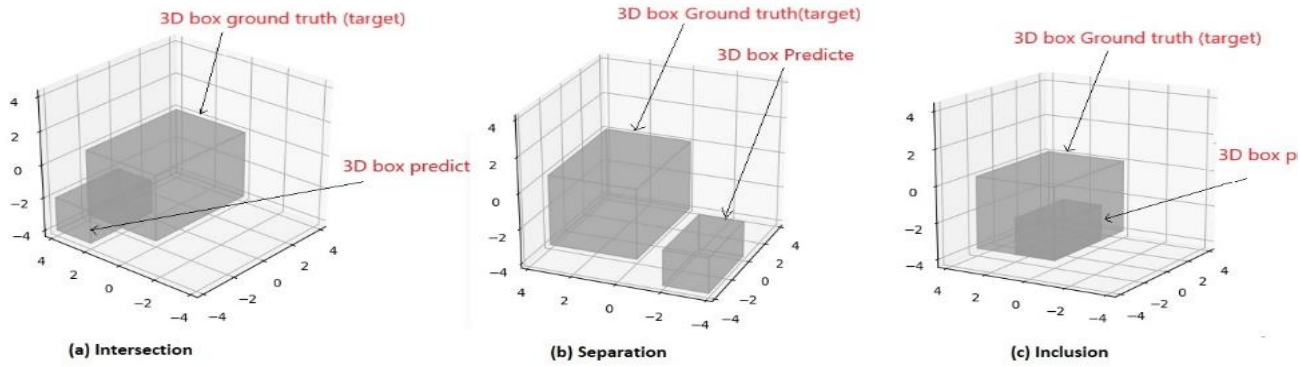

Fig.1 3D bounded boxes, ground truth and predicted in different situation

depth camera could be used, which give the user good information to obtain 3D geometry for an object. Additionally, LIDAR has better performance for getting 3D objects with accurate measurement. The generated points cloud is converted into 2D, whereas the disorganized point cloud is converted to regular grids. These object detection methods are called grid-based methods; any CNN can process the common data after being converted to grids. The work in [1][5]converted the point cloud data to a 2D bird's eye view map with a sequence of 3D bounding boxes for the 3D regression process. Like in [6][7], other methods stated that many layers for feature extraction effectively extract features from point cloud; these methods are named point-based. The 3D CNN was used for the final 3D prediction from the obtained regular feature map. Some methods like F-PointNet [8] and the work [9] used partially the point cloud data gathered by 2D ROI region. PointRCNN [10] and STD [11] used the point cloud data for detecting the 3D object directly. All the above methods used ℓ_n loss function, the IoU loss has been added to STD, significantly improving classification and notable regression improvement.

Many types of research related to the IoU-loss function are popular for 2D object detection, [2] stated the IoU loss function first for detection. IoU-Net [12] showed notable performance improvement by changing the classification score with the IoU value for ranking in NMS. The work [3] proposed a generalized type of IoU, to compensate for weaknesses of the IoU algorithm. Distance-IoU was proposed in [13] to perform the regression of bounding boxes and decrease the center distance between the two boxes. Complete IoU (CIoU) also considers the two-side geometry measurement with the midpoint distance between the two bounding boxes. The above techniques, significantly improved performance for 2D object detection, however they are rarely employed for 3D object detection. The work [14]is the single method that discusses the 3D IoU loss function, which solves the forward and backward problems for two rotating bounding boxes. This 3D IoU method did not solve the slow convergence speed and incorrect regression. Our work also focuses on the loss function model in 3D object detection perspectives. However, by adding three sides of the two bounding boxes to calculate the proposed efficient IoU (EIoU) function, the matching process will be faster and achieves favorable performance

III. METHOD

The previous study [4] defined a 2D EIoU loss function, however, in this work we propose an efficient version of 3D IoU loss, i.e., the 3D EIoU loss. This 3D EIoU loss function can handle three axis-aligned bounding boxes, where the EIoU loss is defined as follows,

$$L_{EIoU} = L_{IoU} + L_{dis} + L_{asp}\ , \tag{1}$$

$$= 1 - IoU + \frac{p^2(b, b^{gt})}{c^2} + \frac{p^2(w^p, w^{gt})}{C_w^2} + \frac{p^2(h^p, h^{gt})}{c_h^2} + \frac{p^2(l^p, l^{gt})}{c_l^2}. \tag{2}$$

where C_w, $C_h$and C_l values are the width, height, and length of the smallest enclosing box containing the two boxes. The above loss equation is divided into three parts: the IoU loss L_{IoU}, the distance loss L_{dis} and the aspect loss L_{asp}. The CIoU loss and DIoU loss characteristics can be observed in the above equation. The disparity between the width, height, and length of the target boxes and the anchor boxes is immediately reduce by the EIoU loss, resulting in a quicker convergence rate and improved localization.

It is crucial to look at how the loss values for the IoU, GIoU, and EIoU algorithms relate to one another. As illustrated in Fig.1, the predicted bounding box and the ground truth box both have three dimensions: width, length, and height. The two 3D bounding boxes have three geometry values; these sides are used for calculating the IoU, GIoU, and EIoU values to see the relationship among these losses in three different positions. Table I, contains the values for these losses in various positions between the bounding boxes: intersection, separation, and inclusion.

Table I. 3D loss functions, at three different places

Loss	Intersection	Separation	Inclusion
$\mathcal{L}_{IoU}$	0.924	1	0.83
$\mathcal{L}_{GIoU}$	1.11	1.563	0.83
$\mathcal{L}_{EIoU}$	1.913	1.835	1.54

As can be understood from table I, the value for IoU loss function is equal to 1 when the two bounding boxes are separated; this situation is shown in Fig.1(b). In other cases, the IoU loss function is bigger than 0.5, as shown in Fig.1(a) and (c). These two cases offer a good performance, and a matching process between the bounding boxes happen. The relationship between the GIoU and IoU losses is presented when the two boxes are in an inclusion situation (Fig.1 (c)), the GIoU loss degenerates to IoU loss. In Fig.1 (a) and Fig.1(b), GIoU loss value is bigger than IoU loss. The relationship between EIoU, GIoU and IoU loss functions shows that the EIoU loss values are bigger than GIoU and IoU loss functions. This indicates the better performance for the EloU loss function, where the change in three geometry sides for the two bounding boxes are considered during the matching process. This lead to a faster convergence speed. In this paper, 3D EIoU algorithm is designed to calculate the loss EIoU value as presented in Algorithm 1.

Algorithm 1: 3D EIoU loss Calculation

Input: Bounded Box of Ground truth
$\boldsymbol{B^{gt} = (x^{gt}, y^{gt}, z^{gt}, w^{gt}, l^{gt}, h^{gt}, x_c^{gt}, y_c^{gt}, z_c^{gt})}$
Input: Bounding Box of Prediction
$\boldsymbol{B^{p} = (x^{p}, y^{p}, z^{p}, w^{p}, l^{p}, h^{p}, x_c^{p}, y_c^{p}, x_c^{p})}$

Output: $\boldsymbol{\mathcal{L}_{EIoU}}$

1: If $\boldsymbol{(B^{gt} \neq 0) \cup (B^{p} \neq 0)}$ ***do***

2: $\boldsymbol{c_w = (x^{gt} - x^{p})}$

3: $\boldsymbol{c_l = (y^{gt} - y^{p})}$

4: $\boldsymbol{c_h = (z^{gt} - z^{p})}$

5: $\boldsymbol{c^2 = c_w^2 + c_l^2 + c_h^2}$

6: $\boldsymbol{p^2(w^p, w^{gt}) = (x_c^{gt} - x_c^{p})^2}$

7: $\boldsymbol{p^2(l^p, l^{gt}) = (y_c^{gt} - y_c^{gt})^2}$

8: $\boldsymbol{p^2(z^p, z^{gt}) = (z_c^{gt} - z_c^{gt})^2}$

9: $\boldsymbol{p^2(b^p, b^{gt}) = (x_c^{gt} - x_c^{p})^2 + (y_c^{gt} - y_c^{p})^2 + (z_c^{gt} - z_c^{p})^2}$

10: $\boldsymbol{\mathcal{L}_{EIoU} = 1 - IoU + \frac{p^2(b,b^{gt})}{c^2} + \frac{p^2(w^p,w^{gt})}{C_w^2} + \frac{p^2(h^p,h^{gt})}{C_h^2} + \frac{p^2(l^p,l^{gt})}{C_l^2}}$

11: else

12: $\boldsymbol{\mathcal{L}_{EIoU} = 0}$

IV. EXPERIMENTATION PROCESS AND RESULTS

All detections are trained, and all outcomes are reported, on the KITTI dataset [15] [16]. The proposed 3D EIoU algorithm evaluated by applying it to the 3D YOLO v4 model [17], including one-stage object detection instructions.

A. Experimental Setting

The experimental condition in this work is organized as follow: laptop with Asus Ryzen 9 CPU (4.6 GHz, 8 cores), 40 GB RAM, Ubuntu 20.04 64-bit operating system. NVIDIA GeForce RTX 3070 laptop GPU 8GB, Cuda V11.1.

B. Dataset and Training

The KITTI benchmark dataset [10] is used to examine our proposed loss function. This dataset holds 7481 training and 7518 testing files, including images, point cloud data, calibration, and label files. The image and point cloud data files contain cars, cyclists, and pedestrians. Using a resolution of 0.1 m per pixel, the point cloud projected in 2D space as a grid map. The grid map that resulted from the LIDAR space is 30.4 meters to the right, 30.4 to the left and 60.8 meters forward, with 0.1 resolution, the above range results in an input shape of 608x608 per channel. The 3D YOLO v4 model is used in this experiment, it is one of the most common neural network models. This model has three components: Backbone, Darknet, Neck, and Head. The Darknet training protocol is used with iteration set to 100K.

C. Analysis Process

The results are reported by measuring the AP value over three label classes for a certain IoU threshold value. These three class labels are car, cyclist, and pedestrian. The AP is used to measure experiment performance, where AP= (AP20+AP30+…+AP90)/12, which is typically the average of AP values across several different values of 12 IoU thresholds, i.e., IoU= {0.2,0.3 ...;0.9}. The values for the AP are reported and compared for EIoU and GIoU training processes. From Table II, $\mathcal{L}_{EIoU}$ gains of 84.65 % AP and 91.59% AP70 can be observed. Taking $\mathcal{L}_{GIoU}$ as the evaluation loss function, the AP value improved to its highest level by 0.2444% and 1.179% respectively. The $\mathcal{L}_{EIoU}$ achieves the highest value, 91.59% AP70, which is higher than the $\mathcal{L}_{GIoU}$ value of the 90.41% AP70, indicating that the good influence of three sides' change between the prediction and the ground truth's 3D bounded boxes on improving the detection accuracy for car object. For cyclist class, $\mathcal{L}_{EIoU}$ gains of 61.24 % AP and 83.57% AP55 . The AP value improved to its highest level by 0.578% and 6.022% correspondingly. The $\mathcal{L}_{EIoU}$ achieves the highest value, 83.57% AP55, which is higher than the $\mathcal{L}_{GIoU}$ value of the 77.55% AP55, indicating that the 3D EIoU loss algorithm can improve detection accuracy for cyclist objects.

For pedestrian class, $\mathcal{L}_{EIoU}$ gains of 41.83% AP and 81.77% AP30. The $\mathcal{L}_{EIoU}$ value increases the detection's precision by 0.1531% AP and 2.548% AP30. The $\mathcal{L}_{EIoU}$ algorithm achieves the highest level of performance improvement, with 3.133% AP70 and 2.821% AP65, slightly higher than $\mathcal{L}_{GIoU}$.

By examining the results in Table II, we discover that the $\mathcal{L}_{EIoU}$ gets the highest scores in car and cyclist classes. However, $\mathcal{L}_{EIoU}$ does not perform well in some places and $\mathcal{L}_{GIoU}$ obtains better performance by taking into account the smallest 3D box containing the ground truth and predicted 3D boxes. In some places it is easy to find that the average precision of $\mathcal{L}_{EIoU}$ is significantly higher than $\mathcal{L}_{GIoU}$, as shown in Fig.3 and Fig.4. The average precision of $\mathcal{L}_{EIoU}$ decrease more slowly with the higher value for the IoU threshold which demonstrates the excellent performance of the network.

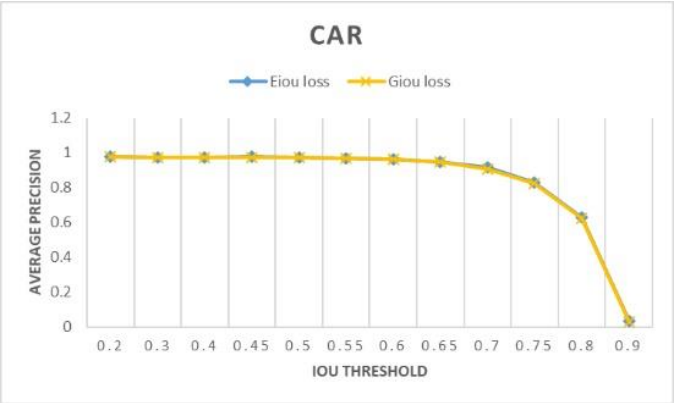

Fig.2 Performance against the IoU threshold for car class

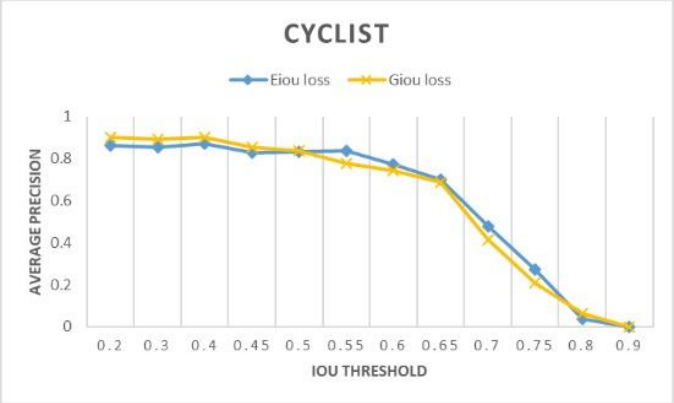

Fig.3 Performance against the IoU threshold for cyclist class

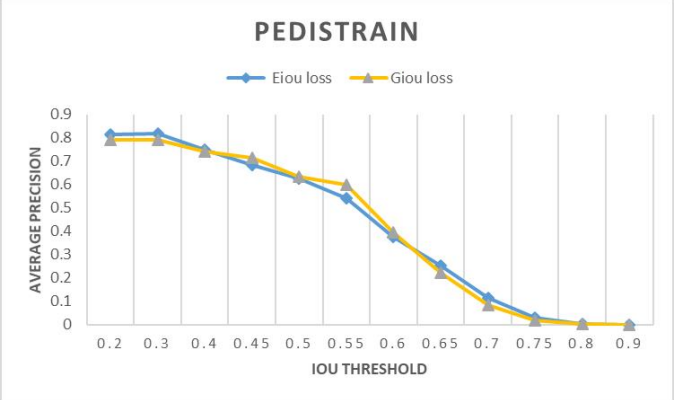

Fig.4 Performance against the IoU threshold for pedestrian class

D. Analysis The Detection Results

By taking the sample from the KITTI dataset, Fig.5, Fig.6, and Fig.7 show the detection outcomes. For RGB images, the 3D bounded boxes projected to their related images of the point cloud samples, 2D bounded boxes for point cloud data and 3D bounded boxes on the image. The image, and the associated point clouds in Fig.5, show typical car detection samples. The $\mathcal{L}_{EIoU}$ can achieve superior detecting outcomes, either it is distance or nearby vehicles, even though reachable points relating to a long-distance vehicle are little.

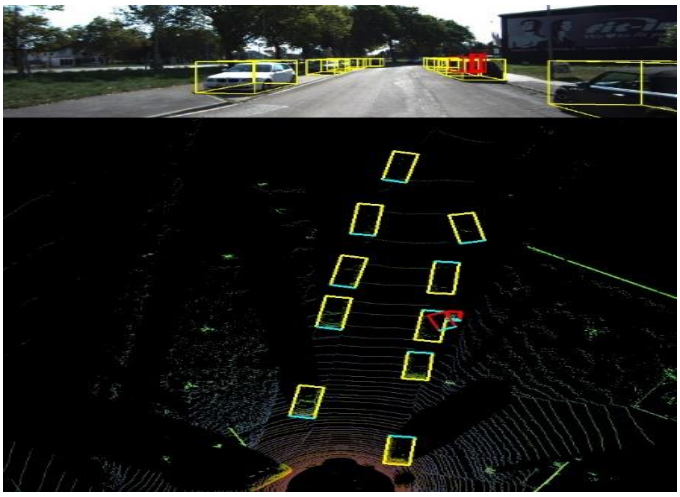

Fig.5 Vehicle object detection by using EIoU loss

Images, and the related point cloud's outcomes, in Fig.6 and Fig.7 showed, the detecting results of cyclists and pedestrians, respectively. There are comparatively a small number of cyclists and pedestrians in the training data set compared with vehicles. Additionally, the size value of cyclists and pedestrians was smaller; each object contain less points, which mistakenly confuse with other objects of similar size.

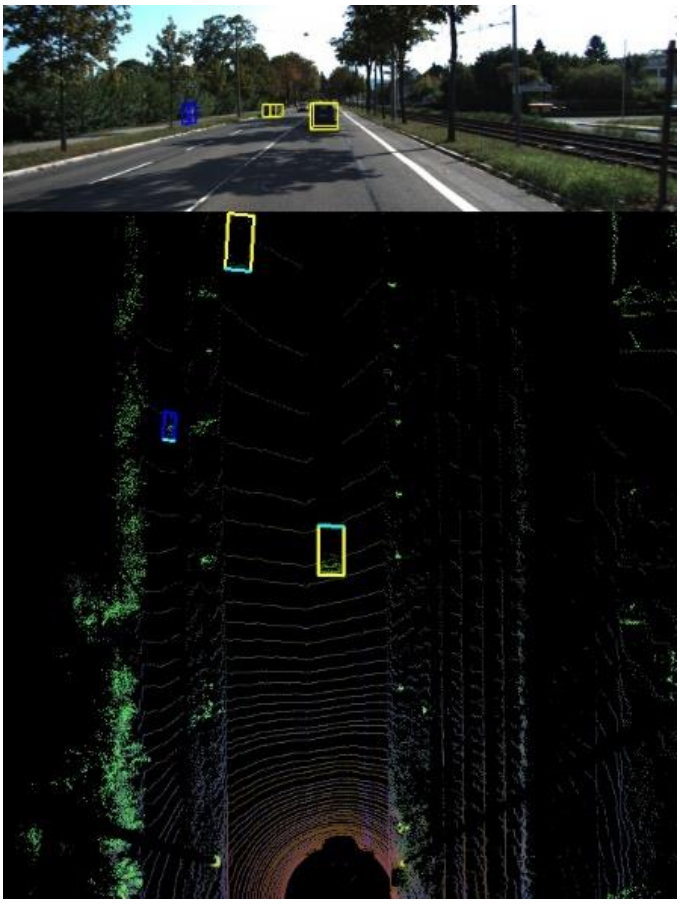

Fig.6 Cyclist with blue label, object detection using the EIoU loss

Table II. Comparison values of 3D YOLO v4 trained using $\mathcal{L}_{EIoU}$ and $\mathcal{L}_{GIoU}$. The outcomes are stated on the test set of KITTI for class car, cyclist and pedestrian.

Loss/ Evaluation	**AP20**	**AP30**	**AP40**	**AP45**	**AP50**	**AP55**	**AP60**	**AP65**	**AP70**	**AP75**	**AP80**	**AP90**	**AP**
Car class													
$\mathcal{L}_{EIoU}$	0.9798	0.9746	0.9738	0.976	0.9745	0.9679	0.96098	0.94451	0.91595	0.8285	0.6298	0.03177	0.8465
$\mathcal{L}_{GIoU}$	0.9760	0.9745	0.974	0.9743	0.9731	0.96948	0.95968	0.94596	0.90416	0.8247	0.6224	0.03044	0.8440
Relative change %	0.38	0.007	-0.02	0.166	0.139	-0.158	0.13	-0.145	1.179	0.382	0.74	0.133	0.2444
Cyclist class													
$\mathcal{L}_{EIoU}$	0.8621	0.854	0.8732	0.8277	0.8316	0.8357	0.7719	0.7013	0.4786	0.2733	0.0394	0000	0.6124
$\mathcal{L}_{GIoU}$	0.8997	0.8937	0.9010	0.8542	0.8384	0.7755	0.7429	0.6886	0.4158	0.2074	0.0620	0000	0.6066
Relative change %	-3.767	-3.985	-2.781	-2.643	-0.688	6.022	2.903	1.269	6.281	6.591	-2.254	0000	0.578
Pedestrian class													
$\mathcal{L}_{EIoU}$	0.8151	0.8177	0.7512	0.6859	0.6269	0.5417	0.3772	0.2524	0.1161	0.0306	0.0048	0000	0.4183
$\mathcal{L}_{GIoU}$	0.7918	0.7922	0.7424	0.7139	0.6339	0.5983	0.3961	0.2242	0.0847	0.01895	0.0048	0000	0.4168
Relative change %	2.333	2.548	0.886	-2.801	-0.707	-5.659	-1.89	2.821	3.133	1.165	0.008	0000	0.1531

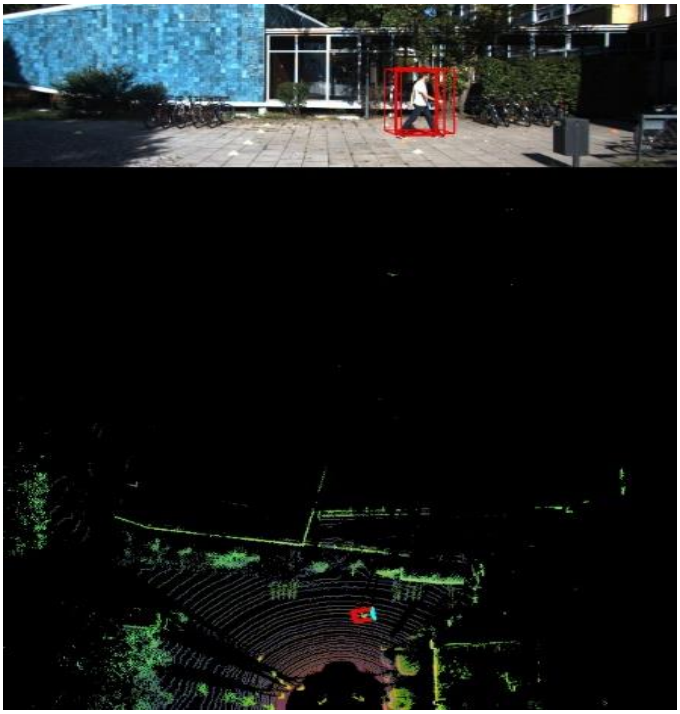

Fig.7 Pedestrian object detection using the EIoU loss

V. CONCLUSION

To accelerate the convergence process for matching the 3D ground truth bounded box with the 3D predicted box, three sides' terms in the $\mathcal{L}_{EIoU}$ of the bounded box regression are employed in this study. These sides term ratios more comprehensively consider the relationship between the 3D ground truth bounded box with the 3D predicted box. The loss function, which includes the sides' term, is called 3D $\mathcal{L}_{EIoU}$. Testing experiments on the KITTI dataset have showed the usefulness of 3D $\mathcal{L}_{EIoU}$ in enhancing the localization accuracy value of the one-stage target detector, the 3D YOLO V4 model. The EIoU loss function improved the position localization of the 3D predicted bounding box. The calculated outcomes of 3D $\mathcal{L}_{EIoU}$ function, show that our algorithm is essential in the 3D field. For future works, this loss function could be verified in other neural network models to examine the proposed function's usefulness better.

ACKNOWLEDGMENT

The research work is supported by research grants: FRGS/1/2020/STG07/UKM/02/3.

REFERENCES

[1] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-Janua, pp. 6526–6534. doi: 10.1109/CVPR.2017.691.

[2] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 516–520.

[3] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658–666. doi: 10.1109/CVPR.2019.00075.

[4] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," *arXiv Prepr. arXiv2101.08158*, 2021.

[5] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D Proposal Generation and Object Detection from View Aggregation," in *IEEE International Conference on Intelligent Robots and Systems*, 2018, pp. 5750–5757. doi: 10.1109/IROS.2018.8594049.

[6] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[7] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499. doi: 10.1109/CVPR.2018.00472.

[8] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 918–927. doi: 10.1109/CVPR.2018.00102.

[9] Z. Wang and K. Jia, "Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1742–1749. doi: 10.1109/IROS40897.2019.8968513.

[10] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 770–779. doi: 10.1109/CVPR.2019.00086.

[11] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: Sparse-to-Dense 3D Object Detector for Point Cloud," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1951–1960. doi: 10.1109/ICCV.2019.00204.

[12] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 784–799.

[13] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, no. 07, pp. 12993–13000.

[14] D. Zhou *et al.*, "IoU Loss for 2D/3D Object Detection," in *2019 International Conference on 3D Vision (3DV)*, 2019, pp. 85–94. doi: 10.1109/3DV.2019.00019.

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361. doi: 10.1109/CVPR.2012.6248074.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Rob. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[17] H.-M. G. Martin Simon, Stefan Milz, Karl Amende, "Complex-YOLO: Real-time 3D Object Detection on Point Clouds," 2018, doi: 10.48550/ARXIV.1803.06199.