Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.044967

ARTICLE

FIR-YOLACT: Fusion of ICIoU and Res2Net for YOLACT on Real-Time Vehicle Instance Segmentation

Wen Dong[1], Ziyan Liu[1,2,*], Mo Yang[1] and Ying Wu[1]

[1]College of Big Data and Information Engineering, Guizhou University, Guiyang, 550025, China

[2]The State Key Laboratory of Public Big Data, Guizhou University, Guiyang, 550025, China

*Corresponding Author: Ziyan Liu. Email: gzucomm@gmail.com

Received: 13 August 2023 Accepted: 03 November 2023 Published: 26 December 2023

ABSTRACT

Autonomous driving technology has made a lot of outstanding achievements with deep learning, and the vehicle detection and classification algorithm has become one of the critical technologies of autonomous driving systems. The vehicle instance segmentation can perform instance-level semantic parsing of vehicle information, which is more accurate and reliable than object detection. However, the existing instance segmentation algorithms still have the problems of poor mask prediction accuracy and low detection speed. Therefore, this paper proposes an advanced real-time instance segmentation model named FIR-YOLACT, which fuses the ICIoU (Improved Complete Intersection over Union) and Res2Net for the YOLACT algorithm. Specifically, the ICIoU function can effectively solve the degradation problem of the original CIoU loss function, and improve the training convergence speed and detection accuracy. The Res2Net module fused with the ECA (Efficient Channel Attention) Net is added to the model's backbone network, which improves the multi-scale detection capability and mask prediction accuracy. Furthermore, the Cluster NMS (Non-Maximum Suppression) algorithm is introduced in the model's bounding box regression to enhance the performance of detecting similarly occluded objects. The experimental results demonstrate the superiority of FIR-YOLACT to the based methods and the effectiveness of all components. The processing speed reaches 28 FPS, which meets the demands of real-time vehicle instance segmentation.

KEYWORDS

Instance segmentation; real-time vehicle detection; YOLACT; Res2Net; ICIoU

1 Introduction

The development of the global economy boosts the automotive industry, and the number of vehicles is increasing yearly. According to relevant data, cars will be over 1.446 billion globally by 2021. The growth of the automobile industry has brought many conveniences to people's lives. On the other hand, it has also caused many social problems, such as traffic congestion and safety. These issues have also become one of the hot points in society. Around 1.35 million people lost their lives in traffic accidents worldwide, according to the World Health Organization's 2018 Global Status Report on Road Safety. In the face of increasingly serious traffic problems, enhancing vehicle driving safety and reducing road accidents are challenges for the entire automotive industry and related researchers.

In recent years, with the development of computer vision technology, many countries have launched research on autonomous driving technology [1,2]. As the critical component of the transportation system, the improved autonomous driving technology will increase driving safety and reduce the risk of human-caused road traffic accidents. Therefore, autonomous driving has become a research focus for the global automotive industry [3–6]. Excellent environmental perception technology is essential for self-driving cars. Current environmental perception technology includes LIDAR-based and camera-based ones. The former is not popularized because of its high cost. On the contrary, due to the low cost, the latter is applied to process the image information in computer vision [7].

The rapid growth of computer hardware and deep learning technology stimulates image processing development. Convolutional neural networks (CNNs) have made excellent achievements in computer vision, and deep learning has become one of the popular scientific research directions. As a comprehensive subject, computer vision covers various popular research directions, such as object detection, pattern recognition, instance segmentation, and object tracking. In autonomous driving technology, determining the category and location of objects in front of the vehicle is a complex task for the computer. Thus, instance segmentation combining object detection and image segmentation can meet the requirements of autonomous driving.

As it is known, autonomous driving is challenging because the different scales of target objects, such as pedestrians and vehicles, may appear in front of the vehicle at the same time. So, it is essential to extract information at different scales of the detection algorithm. However, the drawback of the current instance segmentation networks is poor instance mask prediction because the backbone network mainly concentrates on global features and ignores local features. In addition, existing models' loss functions converge slowly, which prolongs the training time of segmentation models. Furthermore, the standard NMS method also has lower detection accuracy and efficiency.

This paper proposes a real-time instance segmentation algorithm, FIR-YOLACT, by fusing the ICIoU and Res2Net into YOLACT. The FIR-YOLACT can represent multi-scale features at a granular level and enhance each network layer's range of receptive fields, which is practical and robust in various scenarios. In addition, this paper adds the Cluster NMS algorithm to the models' bounding box regression to improve the performance of detecting similar occluded objects. The results demonstrate that FIR-YOLACT outperforms the basic model both quantitatively and qualitatively.

The main contributions of this paper are listed as follows:

- The paper proposes a fusion of ICIoU and Res2Net for the YOLACT algorithm named FIR-YOLACT. Compared with the original algorithm, the proposed algorithm achieves better performance and higher accuracy while meeting the requirements for real-time instance segmentation.
- The paper updates the network's loss function and Non-Maximum Suppression (NMS) algorithm to ICIoU and Cluster NMS, respectively, to improve the accuracy of prediction and detection of similarly obscured objects.
- The paper proposes a module named Res2nEt, which fuses the Res2Net with the ECA Net to represent the multi-scale features at a granular level and enhance each network layer's range of receptive fields.

This paper is organized as follows. In Section 2, this paper reviews related studies on vehicle detection, instance segmentation, and bounding box regression. Section 3 describes the algorithm's overall structure and each improved section's work. This paper indicates the experimental study on the training and evaluation of the model in Section 4. Section 5 summarizes the work and discusses future work.

2 Related Word

2.1 Vehicle Detection

As one of the hotspot research topics in Artificial Intelligence (AI), vehicle detection is an essential part of autonomous driving. Traditional vehicle detection techniques rely heavily on digital image processing, and the image information is digitized through image segmentation, image augmentation, and image transformation. However, with the development of transportation systems, today's traffic environment has become complex and changeable [8,9]. In today's traffic system, the traditional instance segmentation method is inadequate, so this paper focuses on vehicle instance segmentation based on deep learning. Compared with conventional methods, the algorithm relies on CNN to extract images' features, which can achieve higher detection accuracy and satisfy the real-time requirements for vehicle detection and segmentation. Also, the model is robust and can adjust to complex situations and variable environments.

2.2 Image Instance Segmentation

Images are an essential medium for humans to acquire knowledge. Today, in the age of copious data, images are used in various fields, such as medical, remote sensing images, and industrial fields [10,11]. With the advancement of computer vision algorithms and hardware performance, image instance segmentation has emerged as one of the computer vision technologies. Moreover, instance segmentation has achieved many outstanding achievements in computer vision research [12,13]. Instance segmentation is the fusion of object detection and semantic segmentation tasks [14]. Object detection needs to locate and recognize the target object, but it is inaccurate to represent objects with a detection box. Because the box usually contains much background information, accurate boundary information about the object cannot be obtained [15]. Furthermore, instance segmentation separates the target objects at pixel level and clusters them according to their instance classes, which can more accurately separate the target objects in the scene [16]. Therefore, instance segmentation can get more detailed image information and have a broader application scope.

Image instance segmentation methods are generally classified into two-stage methods and one-stage methods. The two-stage instance segmentation method consists of two steps: detection and segmentation. According to the sequential processing order, the two-stage instance segmentation method includes the top-down method based on detection and the bottom-up method based on segmentation. Mask R-CNN [17] is a classical two-stage detection framework extending from Fast R-CNN [18]. One-stage methods obtain the results directly by coalescing detection and segmentation into a single network. And the representatives include PolarMask [19], YOLACT [20] and SOLO [21]. The current instance segmentation methods have been summarized in Table 1.

Table 1: The overview of instance segmentation methods

Type	Algorithm	Highlight	Shortcoming
	MNC [22]	Cascade structure; Fast extrapolation	–
	Mask R-CNN	Adding mask branch to Faster R-CNN; Parallel target detection and segmentation tasks	Relying on target detection results

(Continued)

Table 1 (continued)

Type	Algorithm	Highlight	Shortcoming
Two-stage methods	MS R-CNN [23]	Modification of mask evaluation criteria; Rational evaluation of mask results	–
	PointRend [24]	Consider instance segmentation as a rendering problem in image processing; Refine the mask of Mask R-CNN	Complexity of results processing
	FCIS [25]	Improve the issue of not being able to output the target category	Very low accuracy values
	SOLO	Matrix NMS; Fast speed and high precision	Long training time
Single-stage methods	Polarmask	Polar coordinate modeling for mask	Edge information loss
	CenterMask [26]	Local and global masks; Balancing speed and accuracy	Unable to satisfy the demand for real-time instance segmentation
	YOLACT	Fusion of prototype graphs and detection boxes; Real-time instance segmentation	Lower precision than the two-stage methods

2.3 Loss Function for Bounding Box Regression

Bounding box regression is an essential task in instance segmentation, and the bounding box regression function evaluates the predicted box's detection performance.

Current bounding box regression loss functions include the following categories, such as Mean Square Error (MSE) to evaluate the degree of data variation; Smooth L1 loss for Faster R-CNN; The Intersection over Union (IoU) loss; GIoU [27] loss and DIoU [28] loss. The IoU, a standard function, represents the coverage between the predicted and ground truth boxes to evaluate the algorithms' accuracy. However, it is not suitable that the two boxes do not overlap. To solve this problem, Rezatofighi's team proposed the GIoU function, but it still faces the issues of slow convergence and inaccurate regression. By introducing the central point distance, Zheng presented DIoU loss with much faster convergence than GIoU loss. Based on DIoU, Zheng proposed CIoU [29], adding three important geometric metrics (overlap area, center point distance, and aspect ratio) to achieve faster convergence and better performance than GIoU and DIoU.

3 Methodology

This paper proposes an improved YOLACT algorithm of fusion ICIoU and Res2Net to achieve higher accuracy and speed on real-time vehicle instance segmentation tasks. As illustrated in Fig. 1, first, the paper updates the prediction head's loss function with ICIoU instead of the original one. Then, the original NMS algorithm is enhanced with Cluster NMS [29] to resolve the occlusion problem of similar objects. Third, the backbone network is strengthened using the Res2Net module combined with ECA (Res2nEt) to enhance the advanced feature capability and raise the mask AP score.

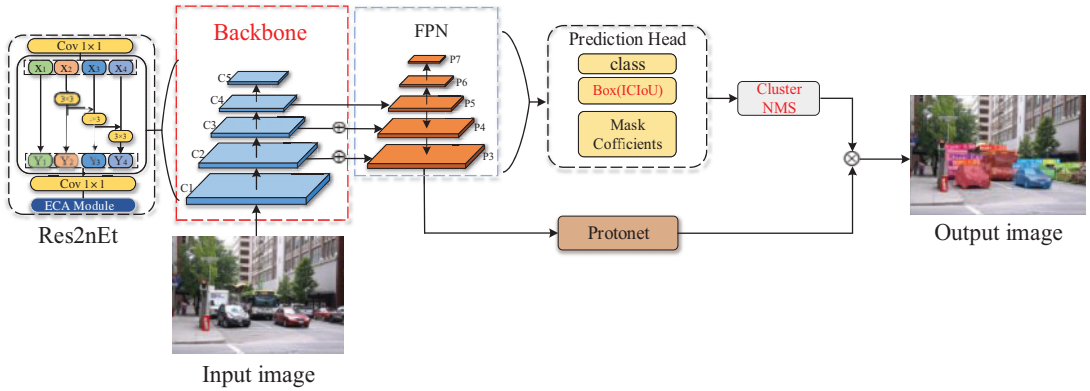


Figure 1: Overall framework

3.1 Backbone

The network structure of the YOLACT algorithm is depicted in Fig. 2. The backbone network is constructed based on ResNet101 and extracts the input image features to generate five feature maps. The three feature maps are used as the middle input layer of the feature pyramid, and the another five feature maps are generated by fusing the features at multiple scales. And then, they are sent into two parallel branching tasks. The first branch takes the feature map through the fully convolutional networks to generate prototype masks. The second branch not only predicts each prediction box's class confidence and position but also generates mask coefficients for each instance. Fast NMS selects the region proposal after bounding box regression to obtain the final instance prediction boxes. Then, the prototype masks and the corresponding mask coefficients are combined linearly to generate the instance mask.

YOLACT uses ResNet-101 as the backbone network. Furthermore, the Bottleneck is the basic unit of ResNet-101. This paper chooses the improved Res2Net module as the Bottleneck to enhance the multi-scale representation ability. The Res2Net constructs hierarchical residual-like connections within one single residual block to increase the range of receptive fields for each network layer [30]. The structure is shown in Fig. 3. The Res2Net replaces the 3×3 convolution with smaller groups of filters. The feature map is divided into four feature map subsets with the same spatial size, and then the output results of the four feature map subsets are gradually fused. Finally, the feature map is output by a 1×1 convolution. The Res2Net will improve the ability of the backbone to extract multi-scale information and the accuracy of the mask prediction.

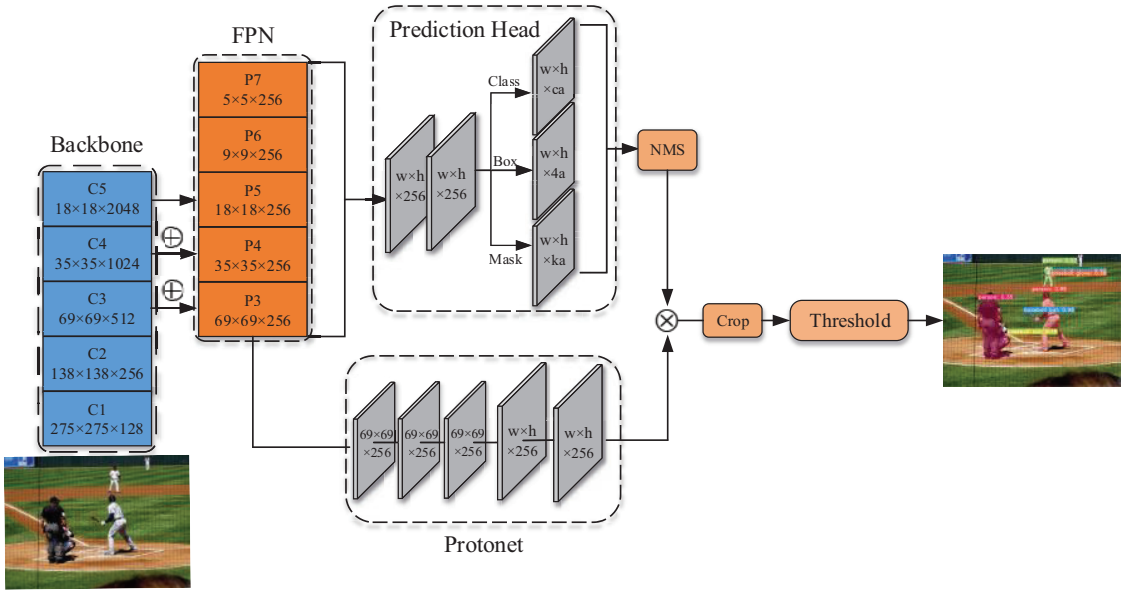
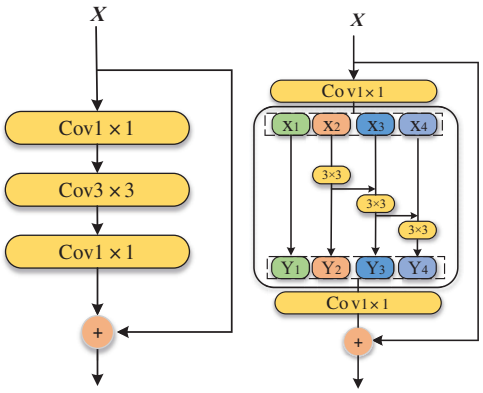


Figure 2: Architectures of YOLACT



(a) Bottleneck (b) Res2Net

Figure 3: Architectures of the Bottleneck and Res2Net

The improved Res2Net (Res2nEt) is incorporated with the ECA attention module (ECA Net) [31], and the introduction of the channel attention module is exceptionally beneficial to the performance of the convolutional neural network model. ECA Net is an improvement on SE Net [32], which can effectively balance the performance and complexity of the model by avoiding dimensionality reduction. Furthermore, it introduces appropriate cross-channel interaction to preserve performance while significantly decreasing model complexity. And Fig. 4 shows the ECA module's structure.

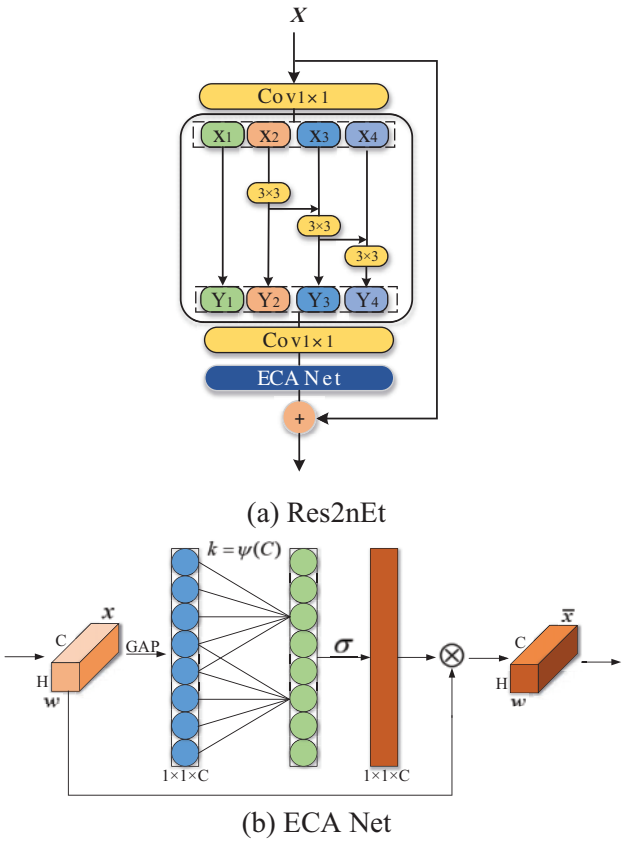


Figure 4: Architectures of the Res2nEt and ECA Net

The ECA Net procedure is as follows: the input feature layer is first processed by global averaging pooling, as given in the following equation:

$$y = \frac{1}{H \times W} \sum_{a}^{H} \sum_{b}^{W} x_i (a, b) \tag{1}$$

In Eq. (1): x_i represents the i-th feature map with input size $H \times W$, and y represents the global feature.

Then, the number of cross channels k is calculated adaptively using the channel dimension C. The adaptive function is formulated as follows:

$$k = \psi (C) = \left| \frac{\log_2 (C)}{\gamma} + \frac{b}{\gamma} \right|_{odd} \tag{2}$$

where $|t|_{odd}$ denotes the nearest odd number to t; C is the channel dimension; b and γ are both constants, $b = 1, \gamma = 2$.

Then, the channel weights are calculated by using a one-dimensional convolution with a convolution kernel size of k to obtain the interdependencies between channels. 1D convolution is formulated

as follows:

$$\omega = \sigma\left(C1D_k\left(\mathrm{y}\right)\right) \tag{3}$$

where ω is the channel weight; σ is the sigmoid function; $C1D$ is the one-dimensional convolution; y is the result of global average pooling, and k is the convolution kernel size. Finally, the original input features are dotted with channel weights to obtain features with channel attention so that the network can selectively enhance valuable features and suppress useless features.

3.2 ICIoU for Loss Function

The bounding box regression loss function measures the position difference between the prediction and ground truth boxes. The loss function of YOLACT is Smooth L1, and the loss is calculated for the length, width, and bias of the horizontal and vertical coordinates of the center point of the prediction box [33]. The original loss function cannot accurately measure the location of the prediction box because it lacks the calculation of the intersection over the union and the minimum bounding rectangle. The CIoU considers three geometric factors: the Intersection over Union, the center point distance, and the aspect ratio of the prediction box and the ground truth box. So the CIoU can measure the performance of the bounding box regression more accurately compared with the original loss function. The equation of CIoU is given below:

$$L_{CIoU} = 1 - IoU\left(b, b^{gt}\right) + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v \tag{4}$$

$$IoU\left(b, b^{gt}\right) = \frac{|b \cap b^{gt}|}{|b \cup b^{gt}|} \tag{5}$$

$$\alpha = \frac{v}{(1 - P_{IoU}) + v} \tag{6}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{7}$$

where ρ denotes the distance between the predicted box b and the geometric center of the ground truth box b^{gt}; c denotes the diagonal length of the minimum bounding rectangle of the predicted box and the ground truth box; w^{gt} and h^{gt} are the width and height of the ground truth box, while w and h are the width and height of the predicted box, respectively.

When $w^{gt}/h^{gt} \neq w/h$, $v > 0$, and $\alpha v > 0$, the penalty term αv plays an active role in the loss calculation. However, when $w^{gt}/h^{gt} = w/h$, then $v = 0$ and $\alpha v = 0$. Currently, L_{CIoU} degenerates to L_{DIoU}, and the convergence speed reduces, as Fig. 5 shows.

Inspired by the CIoU algorithm, Wang et al. proposed the ICIoU algorithm [34], which takes the ratio of the corresponding widths of the two bounding boxes of ground truth and prediction as the geometric factor. Moreover, the penalty function is calculated based on the ratio variance between each side of the ground truth box and the predicted box.

$$L_{ICIoU} = 1 - IoU\left(b, b^{gt}\right) + \frac{\rho^2\left(b^P, b^{gt}\right)}{c^2} + \alpha v^v \tag{8}$$

$$\alpha = \frac{v^v}{(1 - IoU) + v^v} \tag{9}$$

$$v^{v} = \frac{8}{\pi^2}\left[\left(\arctan\frac{w^{gt}}{w^{P}} - \frac{\pi}{4}\right)^2 + \left(\arctan\frac{h^{gt}}{h^{P}} - \frac{\pi}{4}\right)^2\right] \tag{10}$$

The method improves the comprehensiveness of the loss function calculation and effectively avoids the degradation of the CIoU algorithm to the DIoU algorithm when the aspect ratios of the actual and predicted boxes are equal. ICIoU also increases the localization accuracy and enhances the robustness of the loss function for calculating different box sizes.

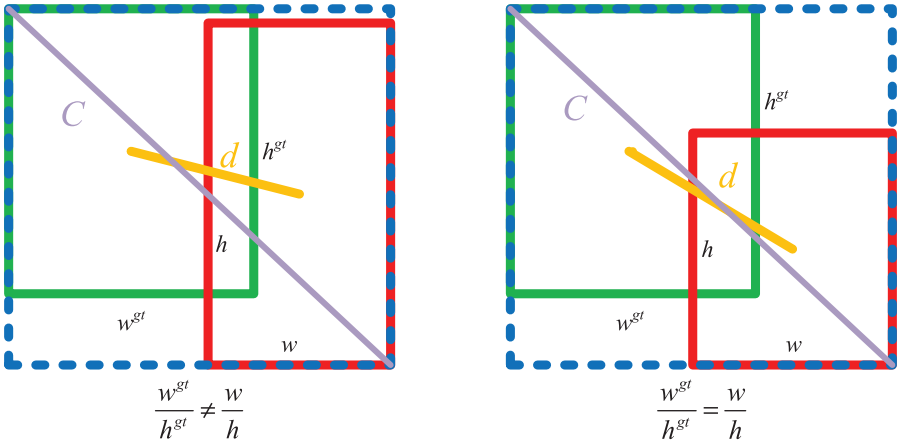


Figure 5: The degradation of CIoU

3.3 Cluster NMS

The YOLACT algorithm uses the Fast NMS algorithm to reduce the redundant boxes. Although the Fast NMS algorithm will significantly increase the processing speed, it quickly causes the region proposal of different instances with a high overlap rate to be mistakenly removed, making some adjacent similar objects easily regarded as one instance. To address this issue, this paper introduces Cluster NMS. Furthermore, it is defined as follows:

(1) Assuming that there are eight prediction boxes arranged in descending order according to confidence scores, $B = [B_1, \ldots, B_8]$; initialize the one-dimensional tensor $a^0 = [1, 1, 1, 1, 1, 1, 1, 1]$ and $t = 1$; compute the IoU matrix $A = \left[x_{ij}\right]_{N\times N}$ with $x_{ij} = IoU\left(B_i, B_j\right)$ and carry out upper triangulation:

$$A = \begin{bmatrix} 0 & 0.64 & 0.36 & 0.18 & 0.41 & 0.11 & 0.26 & 0.8 \\ & 0 & 0.88 & 0.42 & 0.21 & 0.87 & 0.23 & 0.16 \\ & & 0 & 0.51 & 0.46 & 0.36 & 0.42 & 0.24 \\ & & & 0 & 0.23 & 0.25 & 0.65 & 0.27 \\ & & & & 0 & 0.15 & 0.36 & 0.77 \\ & & & & & 0 & 0.42 & 0.29 \\ & & & & & & 0 & 0.87 \\ & & & & & & & 0 \end{bmatrix} \tag{11}$$

(2) The IoU matrix A is binarized according to (12), ε takes the value 0.5, and the processed matrix is:

$$a_{ij} = \begin{cases} 0, if & a_{ij} < \varepsilon \\ 1, if & a_{ij} \geq \varepsilon \end{cases} \tag{12}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ & & 0 & 1 & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 1 & 0 \\ & & & & 0 & 0 & 0 & 1 \\ & & & & & 0 & 0 & 0 \\ & & & & & & 0 & 1 \\ & & & & & & & 0 \end{bmatrix} \tag{13}$$

(3) Expand the initial one-dimensional tensor a^0 into the diagonal matrix P^l and left multiply A by P^l to obtain C^l:

$$P^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

$$C^1 = P^1 \times A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ & & 0 & 1 & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 1 & 0 \\ & & & & 0 & 0 & 0 & 1 \\ & & & & & 0 & 0 & 0 \\ & & & & & & 0 & 1 \\ & & & & & & & 0 \end{bmatrix} \tag{15}$$

(4) Taking the maximum value g by column and if $g > \varepsilon$, set $g = 0$, otherwise $g = 1$, the new one-dimensional tensor a^l consists of g; then a^l is expanded into the diagonal matrix P^2 and left multiply A by P^2 to obtain C^2, $t = 2$:

$$a^1 = [1, 0, 0, 0, 1, 0, 0, 0] \tag{16}$$

$$P^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{17}$$

$$C^2 = P^2 \times A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 1 \\ & & & & & 0 & 0 & 0 \\ & & & & & & 0 & 0 \\ & & & & & & & 0 \end{bmatrix} \tag{18}$$

(5) Repeat the above operation, skipping the intermediate calculation process here, until $t = 4$, when the matrix C^4 is obtained:

$$a^3 = [1, 0, 1, 0, 1, 1, 1, 0] \tag{19}$$

$$C^4 = P^4 \times A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 1 & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 1 \\ & & & & & 0 & 0 & 0 \\ & & & & & & 0 & 0 \\ & & & & & & & 0 \end{bmatrix} \tag{20}$$

(6) When $t = 5$, the maximum value is obtained for C^4 by column for $a^4 = [1, 0, 1, 0, 1, 1, 1, 0]$, $a^3 = a^4$. Then stop the iterative calculation, and predict boxes B_2, B_4, B_8 are suppressed, and B_1, B_3, B_5, B_6, B_7 reprsent the final output. The algorithm flow is shown in Table 2.

Table 2: The algorithm flow of Cluster-NMS

Cluster-NMS
Input: n prediction boxes $B = [B_1, \ldots, B_8]$; arranged in descending order according to confidence scores.
Output: $a^t = [a_i]_{1\times N}$, $a_i \in \{1, 0\}$, where 1 denotes reservation and 0 denotes suppression.
Step1. Initialize $a^0 = [1]_{1\times N}$, $t = 1$, $\varepsilon = 0.5$.
Step2. Compute the IoU matrix $A = \left[x_{ij}\right]_{N\times N}$ with $x_{ij} = IoU\left(B_i, B_j\right)$, carry out upper triangulation, $A = triu\,(A)$.
Step3. Expand the initial one-dimensional tensor a^0 into the diagonal matrix P^t, $P^t = diag\left(a^0\right)$, and left multiply A by P^t to obtain C^t, $C^t = P^t \times A$.
Step4. Taking the maximum value g by the column of C^t and if $g > \varepsilon$, set $g = 0$, otherwise $g = 1$, and the new one-dimensional tensor a^t consists of g, $a^t = [g_1, \ldots, g_n]$.
Step5. If $a^t \neq a^{t-1}$, $t = t + 1$, and jump back to step 2 to continue the calculation until the two vectors are equal.

If the Fast NMS algorithm is as the NMS algorithm, the obtained binarized one-dimensional tensor will be $a = [1, 0, 0, 0, 0, 1, 0, 0, 0]$, and the prediction boxes B_2, B_3, B_4, B_6, B_7, B_8 will be all suppressed, so the prediction results are inaccurate. However, the prediction results of the Cluster NMS algorithm are the same as the traditional NMS results, with the operation time shortened and the detection speed improved.

4 Experiment

4.1 Datasets

This paper's experiments are mainly based on the MS COCO, a large-scale dataset that composes 80 common object classes for object detection and instance segmentation tasks. The MS COCO consists of three data sets: the train set has about 115,000 images, the val set has 5,000 images, and the test-dev set has about 20,000 images.

The experiment selects 9880 vehicle images containing five categories of cars, trucks, buses, motorcycles, and bicycles with annotation information from the COCO 2017 as the training set. Similarly, 870 vehicle images are selected as the val set. The COCO 2017 test-dev is used as the test set, removing the images with annotation information to distinguish training set images and ensure the model's generalization capability. After training the method on the train set, the model's performance is evaluated on the val set. And the model is also compared with state-of-the-art models on the test-dev set.

4.2 Ablation Studies

The ablation experiments are conducted on the train and val set to validate the model's effectiveness. So, the experiments provide the impact of progressively adding the different components, including ICIoU loss function, Cluster NMS, and Res2nEt module, into the baseline. The experiments are accomplished on a computer with an AMD Ryzen 9 5900HX and Nvidia GTX3080 GPU (16G). The software is Python3.9-Conda-Pytorch 1.11. The total number of iterations is 300000, and the batch size is 8.

4.2.1 ICIoU Loss Function and Cluster NMS

In the ablation experiments, the YOLACT instance segmentation model is used to discuss the performance impact of ICIoU loss and Cluster NMS on the prediction box, respectively. The base and the modified models with the ICIoU loss functions are trained separately.

The ICIoU loss function and Cluster NMS perform better in the test results. The improved loss function ICIoU enables the prediction box to more accurately label the target object's location and improve the confidence score, while Cluster NMS increases the detection rate for similar occluded objects. Fig. 6 shows the test results of our model and YOLACT.

The results show that the ICIoU loss function performs better than the CIoU loss function, predicting the target object more accurately and improving the confidence score on the target. For example, the confidence score of the bus on the left of the first-row images is improved after adding ICIoU. In the second and third rows, the confidence scores of the targets on the right side are also improved compared to the left.

The NMS algorithm has good results in detecting single target objects but has shortcomings in the occlusion problem of similar objects. NMS removes the target object with a lower confidence score when the overlap of two objects is high, while Cluster NMS can improve the detection rate for similar occluded objects. As in the first row of pictures, the improved model can detect the target missed in the middle (the part circled by the red dotted line). The second row can detect the motorcycle obscured in the right, and the third row of pictures can detect the vehicle missing at the edge.

To demonstrate their effectiveness, the paper also designed a quantitative experimental analysis, and the evaluation function includes average accuracy AP, inference time Time, and FPS.

YOLACT our model

Figure 6: Comparison of our model (with ICIoU and Cluster NMS) and YOLACT (with CIoU and Fast NMS). Our results are shown in the right column

The curves of the box AP values with the number of iterations for the base model and the added ICIoU loss function model are shown in Fig. 7. The dotted line is the actual measured value, fitted separately for more straightforward observation, and the solid line in the figure is the fitted one.

As shown in Fig. 7, the improved model converges more rapidly than the original base model. Furthermore, the AP values are consistently better than the latter, indicating that the ICIoU loss function can improve the model's performance.

Table 3 compares the CIoU loss function with the ICIoU loss function in the box AP score under different NMS algorithms, and Table 4 shows the results of the mask AP score.

According to the experimental results, the average accuracy of the ICIoU loss function has improved compared with the CIoU loss function in the box AP score and the mask AP score under the same NMS conditions, demonstrating that ICIoU loss predicts the target box more accurately. Also, with the same loss function, the box AP score and the mask AP score of Cluster NMS are better than the fast NMS.

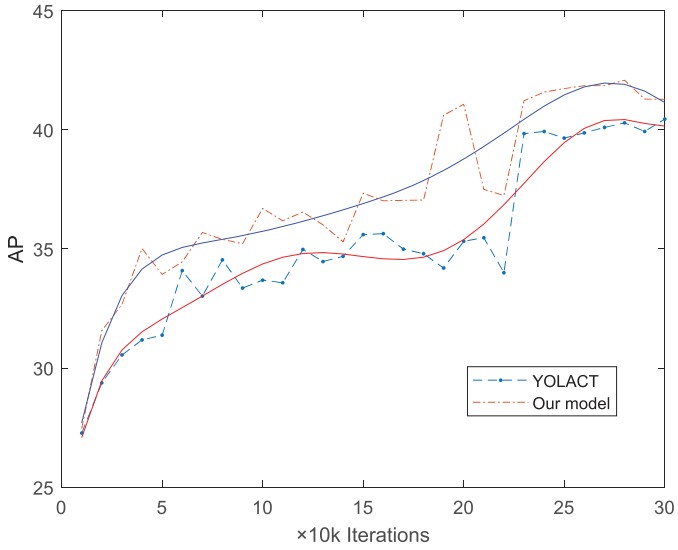


Figure 7: The average accuracy of our model and YOLACT

Table 3: The box AP score of comparison of CIoU loss and ICIoU loss on the val set

Method	Loss	NMS	FPS	Time (ms)	AP	AP_{50}	AP_{75}
YOLACT-550	L_{CIoU}	Fast NMS	36.05	27.74	40.86	63.78	43.19
		Cluster NMS	35.62	28.07	41.75	63.27	45.10
	L_{ICIoU}	Fast NMS	36.25	27.58	40.98	64.44	42.77
		Cluster NMS	35.85	27.89	**42.04**	**64.73**	**45.09**

Table 4: The mask AP score of comparison of CIoU loss and ICIoU loss on the val set

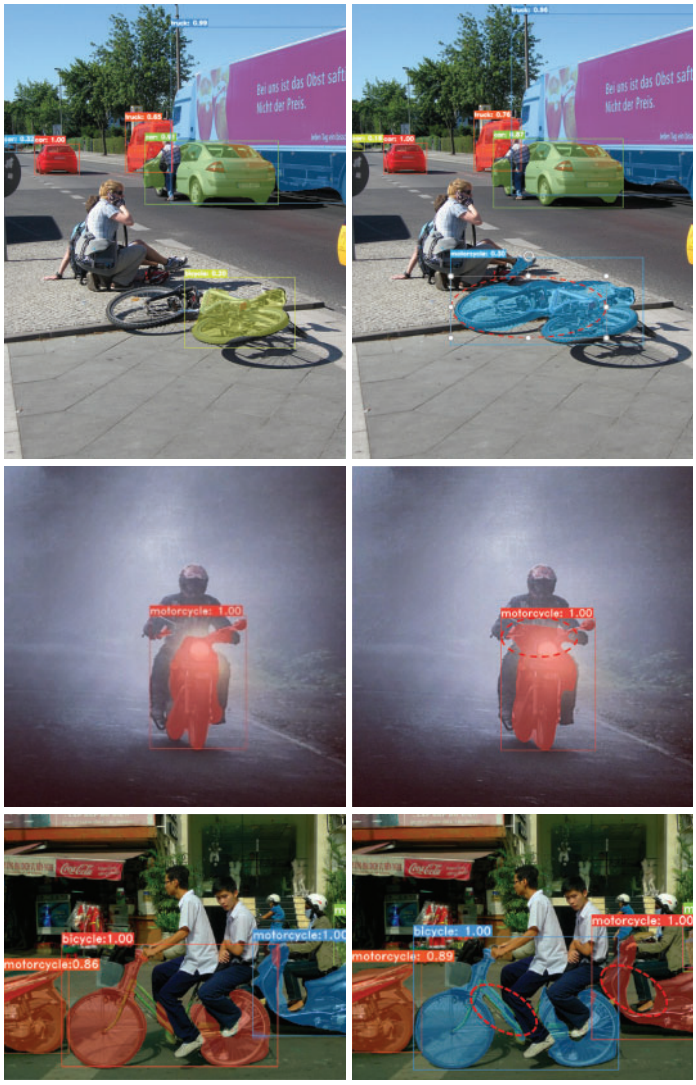
Method	Loss	NMS	FPS	Time (ms)	AP	AP_{50}	AP_{75}
YOLACT-550	L_{CIoU}	Fast NMS	36.05	27.74	35.40	57.45	36.29
		Cluster NMS	35.62	28.07	36.00	58.37	36.50
	L_{ICIoU}	Fast NMS	36.25	27.58	35.53	57.89	35.72
		Cluster NMS	35.85	27.89	**36.02**	**58.10**	**37.17**

4.2.2 Res2nEt Model

The improved Res2net module fused with the ECA module enhances the ability to extract global and local information and represent the multi-scale features at a granular level. Thus, it can improve mask prediction accuracy. Fig. 8 shows the results of comparing the improved models with the baseline.

Fig. 8 shows that the improved model with the Res2nEt has higher accuracy for mask prediction in the instance segmentation task. For example, the mask of the bicycle on the right of the first row is more complete than that on the left, and the model for mask prediction of the motorcycle in the second

row is also improved than the baseline. Similarly, the mask prediction of the bicycle in the third row is also more accurate and complete.



YOLACT our model

Figure 8: Comparison of our model with the original YOLACT. Our results are shown in the right column

To demonstrate the effectiveness of the Res2nEt, Table 5 shows the ablation experiments comparing the box AP values and mask AP values in adding the Res2net module and the ECA module, with the ICIoU as the loss function, on the val set.

Table 5 shows that when only Res2Net is available, the model improves in both box AP values and mask AP values. While the ECA module is added, the model performs best, which proves the effectiveness of the Res2Net module fused with the ECA.

Table 5: The box AP score of comparison of the CIoU loss and the ICIoU loss on the val set

Method	Loss	Res2Net	ECA	AP (box)	AP_{50}	AP_{75}	AP (mask)	AP_{50}	AP_{75}
YOLACT-550	L_{ICIoU}	×	×	42.04	64.73	45.09	36.02	58.10	37.17
		√	×	42.37	65.18	44.76	36.39	58.81	37.87
		√	√	**42.56**	64.18	45.22	**36.73**	**59.12**	37.51

Then, the ablation experiments are performed on the val set using the backbone ResNet101 to prove the effectiveness of integrating the different individual components, including Culser NMS, CIoU, ICIoU, and Res2nEt. The detailed results are shown in Table 6.

Table 6: Ablation study results on val set

Experiment	Models	Baseline	Culser NMS	CIoU	ICIoU	Res2nEt	FPS	Time	AP (box)	AP (mask)
1	YOLACT-550	√	×	×	×	×	35.85	27.90	40.45	35.80
2		√	√	×	×	×	35.72	28.00	40.71	36.18
3		√	√	√	×	×	35.62	28.07	41.75	36.00
4		√	√	×	√	×	35.85	27.89	42.04	36.02
5		√	√	×	√	√	**29.44**	**33.96**	**42.56**	**36.73**

Experiment 1 is the basic YOLACT; Experiment 2 changes the Fast NMS to the Culser NMS; Experiment 3 replaces the loss function with CIoU based on Experiment 2; And in Experiment 4, the ICIoU is applied as the loss function; Experiment 5 adds the Res2nEt module.

The results of the experiments show that all the components contribute towards improving the accuracy. It can be seen from Experiment 2 that the application of Culser NMS has improved the box AP score. Compared with Experiment 3, Experiment 4 has improved the box AP, so the ICIoU has played a more positive role in the model. The final results of Experiment 5 show that the mask AP score increased after integrating Res2nEt module into the baseline, proving the effectiveness of Res2nEt. After combining all improvements into the baseline, our model obtains a box AP score of 42.56% and a mask AP score of 36.73%.

4.3 Algorithm Comparison and Analysis

In this subsection, the experiments are conducted on the MS COCO train set to compare our method with some typical state-of-the-art methods on MS COCO test-dev in terms of accuracy (mask AP), speed (milliseconds and FPS) and model complexity (parameters P and FLOPs). The total number of iterations is 800000, and the batch size is 8. The results are demonstrated in Table 7.

Table 7 indicates that the MNC and FCIS have mask AP scores of 24.6% and 29.2%, respectively. Moreover, the typical Mask R-CNN, MS R-CNN, PoinInst, and SOLO have mask AP scores of 35.7%, 38.3%, 38.3%, and 37.8%, respectively. Although they outperform our model regarding mask AP, they require more than 75 milliseconds (ms) to detect an image during inference. Moreover, the QueryInst and SipMask algorithms only improve the accuracy and ignore processing speed. Therefore, they are unsuitable for real-time image processing for autonomous driving scenarios. Conversely, our model meets the demands of real-time image processing while outperforming the original YOLACT by 3.0% of the mask AP score, and our model is not much more complex than the original algorithm according to the FLPOs and P.

Table 7: Comparison of our method to other instance segmentation frameworks on MS COCO test-dev datasets

Method	Backbone	Time	FPS	P	FLOPs	AP	AP_{50}	AP_{75}
MNC	ResNet101	–	–	–	–	24.6	44.3	24.8
FCIS	ResNet101	–	–	–	–	29.2	49.5	–
Mask R-CNN	ResNet101	116	9	62.79	152.71	35.7	58.0	37.8
MS R-CNN	ResNet101	116	9	79.02	151.17	38.3	58.8	41.5
PointInst [35]	ResNet101	75	13	–	–	38.3	60.3	40.0
SOLO	ResNet101	86	12	54.91	442.13	37.8	59.5	40.4
QueryInst [36]	ResNet101	166	6	191.51	1065.17	42.8	65.6	46.7
SipMask [37]	ResNet101	65	15	–	–	38.1	60.2	40.8
YOLACT	ResNet101	30	33	53.78	85.41	29.8	48.5	31.2
Our model	ResNet101	35	28	54.46	88.83	**32.8**	**52.2**	**34.6**

In addition, Table 8 compares our model with other instance segmentation methods on the Pascal SBD test set. The experimental results show that our method performs best on the Pascal SBD dataset.

Table 8: Experimental results of different methods on the Pascal SBD test set

Method	mAP_{vol}	$mAP_{0.5}$	$mAP_{0.7}$
ESE-50 [38]	32.6	39.1	10.5
ESE-20 [38]	35.3	40.7	12.1
SDS [39]	41.4	49.7	25.3
Hypercolumns [40]	–	56.5	37.0
DeepSnake [41]	54.4	62.1	48.3
MNC	–	63.5	41.5
DIN [42]	55.4	52.0	44.8
YOLACT	–	72.3	56.2
Our model	–	**74.2**	**59.1**

Meanwhile, the improved model also enhances the original model's small-scale object detection capability, as shown in Fig. 9.

From Fig. 9, the improved model is superior to small-scale objects. For example, the car, the bicycle, and the smaller car are all detected correctly.

The proposed model is tested on the MS COCO test-dev datasets and compared with the original YOLACT algorithm on small, medium, and large targets. The results are listed in Tables 9 and 10.

The data from Tables 9 and 10 indicates that the improved algorithm increases by 4.2% in the box AP and 3.0% in the mask AP compared to the original YOLACT algorithm, while it achieves 1.8% and 1.1% improvement in the small object's box AP and mask AP, respectively.

YOLACT our model

Figure 9: Comparison of our model with the original YOLACT. Our results are shown in the right column

Table 9: The box AP score of comparison of our method and YOLACT on the MS COCO test-dev datasets

Box	AP	AP_{50}	AP_{75}	AP_s	AP_M	AP_L
YOLACT-550	32.3	53.0	34.3	14.9	33.8	45.6
Our model	**36.5**	**56.2**	**39.0**	**16.7**	**38.6**	**51.7**

Table 10: The mask AP score of comparison of our method and YOLACT on the MS COCO test-dev datasets

Mask	AP	AP_{50}	AP_{75}	AP_s	AP_M	AP_L
YOLACT-550	29.8	48.5	31.2	9.9	31.3	47.7
Our model	**32.8**	**52.2**	**34.6**	**11.0**	**34.6**	**52.4**

5 Conclusions and Future Work

This paper proposes the FIR-YOLACT vehicle instance segmentation algorithm to tackle the current problems, such as slow convergence and long training time. The proposed algorithm utilizes the Cluster NMS algorithm for bounding box regression to improve the accuracy of predicting and detecting similarly obscured objects. Additionally, the original loss function is replaced with ICIoU to prevent the degradation of the CIoU algorithm and strengthen the model's robustness. To extract richer image information and increase the mask accuracy scores, this paper makes the backbone network incorporate the Res2net module fused with ECA Net. The experimental results demonstrate that FIR-YOLACT performs significantly better than the original model, with a 4.2% and 3.0% increase in box AP and mask AP scores, respectively. Moreover, FIR-YOLACT achieves a processing speed of 28 FPS, indicating its excellent performance in balancing accuracy and processing speed.

However, the proposed method still has some shortages to be further improved. For example, the model does not consider the impact of complex weather scenarios and needs optimization for inter-frame timing information in the video dataset. In the future, we plan to deploy the model on a low-cost mobile platform and employ Tensor RT technology to improve the speed of the model detection and extend the proposed approach to video instance segmentation detection. We will explore practical methods to optimize the model to improve the system's real-time detection performance and find new application scenarios in the times of new energy smart vehicles.

Acknowledgement: Thanks are given for the computing support of the State Key Laboratory of Public Big Data, Guizhou University.

Funding Statement: This work is supported by the Natural Science Foundation of Guizhou Province (Grant Number: 20161054), Joint Natural Science Foundation of Guizhou Province (Grant Number: LH20177226), 2017 Special Project of New Academic Talent Training and Innovation Exploration of Guizhou University (Grant Number: 20175788), The National Natural Science Foundation of China under Grant No. 12205062.

Author Contributions: The authors confirm their contribution to the paper as follows: Wen Dong: Methodology: development or design of methodology; creation of models; software: programming, software development; designing computer programs; implementation of the computer code and supporting algorithms; testing of existing code components; writing-original draft: preparation, creation, and presentation of the published work, specifically writing the initial draft (including substantive translation); validation: verification, whether as a part of the activity or separate, of the overall replication/reproducibility of results/experiments and other research outputs. Ziyan Liu: Conceptualization: ideas; formulation or evolution of overarching research goals and aims; project administration: management and coordination responsibility for the research activity planning and execution; supervision: oversight and leadership responsibility for the research activity planning and execution, including mentorship external to the core team; writing-review & editing: preparation, creation and/or presentation of the published work by those from the original research group, specifically critical review, commentary or revision including pre-or post-publication stages; funding acquisition: acquisition of the financial support for the project leading to this publication. Mo Yang: Software; validation. Ying Wu: Software; validation. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Research data are not shared. Due to the participants of this study did not agree for their data to be shared publicly, so supporting data is not available.

Conflicts of Interest: The authors declare they have no conflicts of interest to report regarding the present study.

References

[1] K. Roszyk, M. R. Nowicki and P. Skrzypczyński, "Adopting the YOLOv4 architecture for low-latency multispectral pedestrian detection in autonomous driving," *Sensors*, vol. 22, no. 3, pp. 1082, 2022.

[2] Z. Chen, H. Guo, J. Yang, H. Jiao, Z. Feng *et al.,* "Fast vehicle detection algorithm in traffic scene based on improved SSD," *Measurement*, vol. 201, no. 1, pp. 111655, 2022.

[3] H. Gajjar, S. Sanyal and M. Shah, "A comprehensive study on lane detecting autonomous car using computer vision," *Expert Systems with Applications*, vol. 233, no. 1, pp. 120929, 2023.

[4] L. G. Zhou, G. Chen, L. Liu, R. N. Wang and A. Knoll, "Real-time semantic segmentation in traffic scene using cross stage partial-based encoder-decoder network," *Engineering Applications of Artificial Intelligence*, vol. 126, no. 1, pp. 106901, 2023.

[5] S. Liu, S. Huang, X. Xu, J. Lloret and K. Muhammad, "Efficient visual tracking based on fuzzy inference for intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 1, pp. 1–12, 2023.

[6] Y. Song, S. Hong, C. Hu, P. He, L. Tao *et al.,* "MEB-YOLO: An efficient vehicle detection method in complex traffic road scenes," *Computers, Materials & Continua*, vol. 75, no. 3, pp. 5761–5784, 2023.

[7] X. Chang, H. Pan, W. Sun and H. Gao, "A multi-phase camera-LiDAR fusion network for 3D semantic segmentation with weak supervision," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 3737–3746, 2023.

[8] C. Gao, G. Wang, W. Shi, Z. Wang and Y. Chen, "Autonomous driving security: State of the art and challenges," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7572–7595, 2021.

[9] S. Liu, S. Huang, S. Wang, K. Muhammad, P. Bellavista *et al.,* "Visual tracking in complex scenes: A location fusion mechanism based on the combination of multiple visual cognition flows," *Information Fusion*, vol. 96, no. 1, pp. 281–296, 2023.

[10] S. Wang, S. Huang, S. Liu and Y. Bi, "Not just select samples, but exploration: Genetic programming aided remote sensing target detection under deep learning," *Applied Soft Computing*, vol. 145, no. 1, pp. 110570, 2023.

[11] Z. Chen, J. Yang, L. Chen, Z. Feng and L. Jia, "Efficient railway track region segmentation algorithm based on lightweight neural network and cross-fusion decoder," *Automation in Construction*, vol. 155, no. 1, pp. 105069, 2023.

[12] T. Ngo, B. Hua and K. Nguyen, "ISBNet: A 3D point cloud instance segmentation network with instance-aware sampling and box-aware dynamic convolution," in *Proc. of 2023 IEEE/CVF Conf. on Computer Vision and Pattern Recognition(CVPR)*, Vancouver, Canada, pp. 13550–13559, 2023.

[13] W. Ye, W. Zhang, W. Lei, W. Zhang, X. Chen *et al.,* "Remote sensing image instance segmentation network with transformer and multi-scale feature representation," *Expert Systems with Applications*, vol. 234, no. 30, pp. 121007, 2023.

[14] J. Ma, S. Gu, Y. Deng and T. Ao, "Instance segmentation algorithm based on fine-grained feature perception and cross-path aggregation," *Knowledge-Based Systems*, vol. 276, no. 27, pp. 110754, 2023.

[15] N. Anoob, S. J. Ebey, P. Praveen, P. Prabudhan and P. Augustine, "A comparison on instance segmentation models," in *Proc. of 2021 Int. Conf. on Advances in Computing and Communications (ICACC)*, Kochi, Kakkanad, India, pp. 1–5, 2021.

[16] W. Gu, S. Bai and L. Kong, "Review on 2D instance segmentation based on deep neural networks," *Image and Vision Computing*, vol. 120, no. 1, pp. 104401, 2022.

[17] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," in *Proc. of 2017 IEEE Int. Conf. on Computer Vision (ICCV)*, Hawaii, USA, pp. 2961–2969, 2017.

[18] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
[19] E. Xie, P. Sun, X. Song, W. Wang, X. Liu *et al.,* "PolarMask: Single shot instance segmentation with polar representation," in *Proc. of 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, pp. 12193–12202, 2020.
[20] D. Bolya, C. Zhou, F. Xiao and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. of 2019 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Seoul, Korea, pp. 9157–9166, 2019.
[21] X. Wang, R. Zhang, C. Shen, T. Kong and L. Li, "SOLO: A simple framework for instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8587–8601, 2021.
[22] J. Dai, K. He and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. of 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, pp. 3150–3158, 2016.
[23] Z. Huang, L. Huang, Y. Gong, C. Huang and X. Wang, "Mask scoring R-CNN," in *Proc. of 2019 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Seoul, Korea, pp. 6409–6418, 2019.
[24] A. Kirillov, Y. Wu, K. He and R. Girshick, "PointRend: Image segmentation as rendering," in *Proc. of 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, pp. 9799–9808, 2020.
[25] Y. Li, H. Qi, J. Dai, X. Ji and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proc. of 2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Hawaii, USA, pp. 2359–2367, 2017.
[26] Y. Lee and J. Park, "CenterMask: Real-time anchor-free instance segmentation," in *Proc. of 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, pp. 13906–13915, 2020.
[27] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid *et al.,* "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. of 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, USA, pp. 658–666, 2019.
[28] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye *et al.,* "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. of the AAAI Conf. on Artificial Intelligence*, New York, USA, pp. 12993–13000, 2020.
[29] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye *et al.,* "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 8574–8586, 2022.
[30] S. H. Gao, M. M. Cheng, K. Zhao, X. Y. Zhang, M. H. Yang *et al.,* "Res2Net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2019.
[31] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo *et al.,* "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. of 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, New York, USA, pp. 11534–11542, 2020.
[32] J. Hu, L. Shen, S. Albanie, G. Sun and E. Wu, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020.
[33] Q. Wang, Y. Ma, K. Zhao and Y. Tian, "A comprehensive survey of loss functions in machine learning," *Annals of Data Science*, vol. 9, no. 1, pp. 187–212, 2022.
[34] X. Wang and J. Song, "ICIoU: Improved loss based on complete intersection over union for bounding box regression," *IEEE Access*, vol. 9, no. 1, pp. 105686–105695, 2021.
[35] L. Qi, Y. Wang, Y. Chen, Y. C. Chen, X. Zhang *et al.,* "PointINS: Point-based instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6377–6392, 2020.
[36] Y. Fang, S. Yang, X. Wang, Y. Li, C. Fang *et al.,* "Instances as queries," in *Proc. of 2021 IEEE/CVF Int. Conf. on Computer Vision (CVPR)*, pp. 6910–6919, 2021.

[37] J. Cao, R. M. Anwer, H. Cholakkal, F. S. Khan, Y. Pang *et al.,* "SipMask: Spatial information preservation for fast image and video instance segmentation," in *Proc. of European Conf. on Computer Vision (ECCV)*, Glasgow, UK, pp. 1–18, 2020.
[38] W. Xu, H. Wang, F. Qi and C. Lu, "Explicit shape encoding for real-time instance segmentation," in *Proc. of 2019 IEEE Int. Conf. on Computer Vision (ICCV)*, Long Beach, USA, pp. 5168–5177, 2019.
[39] B. Hariharan, P. Arbelez, R. Girshick and J. Malik, "Simultaneous detection and segmentation," in *Proc. of European Conf. on Computer Vision (ECCV)*, Zurich, Switzerland, pp. 297–312, 2014.
[40] B. Hariharan, P. Arbelez, R. Girshick and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. of 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, pp. 447–456, 2015.
[41] S. Peng, W. Jiang, H. Pi, X. Li and H. Bao, "Deep snake for real-time instance segmentation," in *Proc. of 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Seattle, USA, pp. 8533–8542, 2020.
[42] A. Arnab and P. H. Torr, "Pixelwise instance segmentation with a dynamically instantiated network," in *Proc. of 2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Hawaii, USA, pp. 441–450, 2017.