Road Object Detection Method Based on Improved SED-CRCNN

Xinying Chen, Shuo Lv, Wei Jiang, Ying Liu, and Mingjie Hu

Abstract—**Object detection has attracted significant attention in the field of autonomous driving. It assists autonomous driving systems in recognizing the surrounding roads, pedestrians, and other vehicles, thereby enhancing path planning and automatic navigation to improve the system's safety and reliability. In road condition images, the size of vehicle targets exhibits significant variation, while pedestrian targets tend to be smaller and more densely packed. Additionally, the complexity of road information further complicates the scene. These small targets, containing limited image information, pose a challenge for recognition and are susceptible to being overlooked during network training. Consequently, this study prioritizes the cascading target detection model as its research baseline. To achieve more accurate prediction boxes, multiple data preprocessing operations were employed. Moreover, to adapt to the varied shapes and sizes of vehicle targets, an attention mechanism was integrated into the backbone network. Subsequently, the study leveraged deformable convolution methods to enhance the global context correlation of feature maps and conducted vehicle target training and prediction across multiple-scale feature maps. Experiments were conducted in public settings. The available vehicle detection datasets include KITTI and SODA10M. A series of ablation and comparative experiments were completed. The results demonstrate that the algorithm proposed in this study significantly improves accuracy and mean precision in road target detection, exhibiting an overall effective detection performance. This method can provide substantial support to autonomous vehicle systems, thereby reducing the risk of traffic accidents.**

Index Terms—**Image processing, Deep learning, Object detection, Deformable convolution.**

I. Introduction

THE development of computers and the Internet has led to an increased focus on using these technologies to solve problems. One area of interest is the potential for computers to replace human workers, thereby reducing the need for human resources. This has led to the emergence of artificial intelligence technology, which is seen as an extension of human ability. The goal is to create intelligent machines that can respond similarly to human intelligence. These machines can continuously acquire knowledge and learn by simulating the information-processing processes of human thinking and consciousness. Artificial intelligence is currently being used in various fields, such as knowledge representation and automatic reasoning, machine learning and knowledge acquisition, natural language understanding, and computer vision.

Object detection is a popular branch in the field of computer vision, which is a technique used for the fast and accurate identification of specified targets. The principle is to use various image processing methods and algorithms to simulate the human visual system to determine the location and class of a given object. Object detection primarily uses feature extraction and classification algorithms to identify objects in an image. To identify objects in an image, feature extraction algorithms collect features from the image, such as edges, colors, textures, etc. Classification algorithms can identify objects in an image based on the extracted features. Object detection techniques are widely used and play an important role in fields such as intelligent surveillance, autonomous driving, virtual reality, and medical diagnosis.

Road object detection, as an important part of autonomous driving environment awareness, provides basic support for subsequent higher-level tasks such as decision planning and behavioral control of the vehicle. The main purpose is to detect element information such as vehicle and pedestrian targets and their position in the picture from still pictures or moving videos, and to classify vehicle types. It helps autonomous driving systems to recognize and understand their surroundings more accurately and quickly, and thus to better realize autonomous driving. As the foundation and core of intelligent mobility, vehicle detection is an important part of the operation of driverless vehicles and is of great relevance in higher-level vision tasks such as target tracking and event detection.

The traditional road target detection method mainly uses the artificial feature extraction methods. Feature extraction is to make a transformation or encoding of the data, which is mapped and transformed from a high-dimensional original feature space to a low-dimensional space. Good features are supposed to be undistorted and distinguishable. In 2005, Dala et al. proposed a target detection algorithm based on feature extraction (Histogram of oriented gradients, HOG) [1]. It extracts features by calculating the histogram of the gradient direction in the image to achieve target detection. For the moving target, Lowe proposed the Scale Invariant Feature Transform (SIFT) [2], whose idea is to use the gradient information near the key points of the image for feature representation.

Manuscript received January 28, 2024; revised March 9, 2025. This work has been supported by the Liaoning Provincial Department of Transportation Science and Technology Program (No.2023-360-17, No.2024-353-5, No.SZJT19), the Liaoning Provincial Science and Technology Department (No. 2022JH2/101300268), and the Fundamental Research Funds for the Provincial Universities of Liaoning (LJ212410150047, LJ212410150037).

X. Y. Chen is a professor at the School of Computer and Communication Engineering, Dalian Jiaotong University, Dalian, LiaoNing, 116021, China. (e-mail: chenxy1979@163.com).

S. Lv is a postgraduate student at the School of Computer and Communication Engineering, Dalian Jiaotong University, Dalian, LiaoNing, 116021, China. (e-mail: lvshuo4262506@163.com).

W. Jiang, PhD, is a teacher at the Key Laboratory of Advanced Design and Intelligent Computing of the Ministry of Education, Dalian University, Dalian, LiaoNing, 116622, China. (Corresponding author, e-mail: jiangwei@dlu.edu.cn).

Y. Liu is a postgraduate student at the School of Computer and Communication Engineering, Dalian Jiaotong University, Dalian, LiaoNing, 116021, China. (e-mail: 17686497985@163.com).

M. J. Hu is a postgraduate student at the School of Computer and Communication Engineering, Dalian Jiaotong University, Dalian, LiaoNing, 116021, China. (e-mail: 1021988931@qq.com).

However, in the traditional road object detection algorithm, since the relevant target feature information needs to be obtained manually, the superiority of the features extracted in the manual feature extraction stage will directly affect the detection performance of the whole algorithm. Traditional detection algorithms have many limitations, which also greatly limit their effectiveness in practical applications and directly affect the accuracy of the algorithms. At the same time, due to environmental changes, light intensity, object shape changes and other factors, artificial construction features lack of robustness and generalization. In addition, the high complexity of the sliding window method, along with a significant amount of redundant calculations, inevitably makes it tough to enhance the operational speed.

Deep learning methods rely mainly on the design of network structures and feature representation. The advantage of road target detection algorithms based on deep learning over traditional road target detection methods is that they do not require manual feature design. In addition, the convolutional neural network can automatically extract multi-level vehicle features from the shallow position information of the original image to the high-level semantic information, thus improving the accuracy and stability of detection. The object detection algorithm based on deep learning can be divided into a two-stage method based on the candidate frame and a one-stage method based on regression. The former requires the extraction of candidate boxes before classification, while the latter directly returns the detection results into the input image. The two-stage method has the advantage of being able to extract image detail features, so the algorithm has good detection accuracy.

However, the model runs slowly due to the fact that the algorithm is performed in two steps. Convolutional neural network (CNN) [3] is a typical two-stage target detection method. Region convolutional neural networks (R-CNNs) [4] were further proposed by continuous optimization of neural network algorithms. In addition, it is the first algorithm to apply deep learning to target detection. The Fast R-CNN algorithm was proposed by Girshick et al [5] in 2015, which incorporates the idea of SPPNet by designing the SPP layer of the network as a separate layer, i.e., the ROI Pooling layer, which further solves the problem of updating weights, improves the training performance and speeds up the training. Azam et al [6] used the Faster R-CNN method, a modified algorithm of Fast R-CNN, to detect the license plate number and body color of a vehicle and compared their accuracy from four perspectives of the vehicle.

In order to solve the slow detection problem of the two-stage method, in 2013, Sermanet [7] et al. proposed the well-known OverFeat algorithm, which can simultaneously perform classification, location and detection tasks. It was the first paper to propose a one-stage detection idea approach. In 2015, J. Redmon [8]et al. proposed the first one-stage detection method: the YOLO algorithm. This algorithm applies the neural network directly to the image, which greatly improves the detection speed, thus making real-time video detection a reality. However, it is ineffective in small target detection tasks, generating the problem of poor generalization ability caused by anomalous aspect ratios between similar objects.YOLOv2 [9], based on the YOLO algorithm, utilizes the Darknet-19 network structure, multi-scale feature fusion, and anchor frames to enhance detection performance and speed. However, it still faces challenges in detecting small targets effectively. Therefore, Redmon et al. proposed a new target detection method, namely the YOLOv3 [10] algorithm, which combines the advantages of YOLOv2 and residual networks, effectively addressing issues such as the challenge of detecting small targets. However, there are still problems, such as missing information resulting from the multilayer feature extraction process.

In December 2016, Liu et al [11] proposed the SSD algorithm, which utilizes VGG-16 as the backbone network and integrates the concept of multiscale feature map prediction to address the issue of subpar performance of the YOLO algorithm in small target detection. Therefore, the algorithm significantly improves the detection accuracy of the one-stage method.

Although the one-stage detection method is faster, its accuracy is lower, and its localization is less effective compared to the two-stage algorithm. The speed and accuracy of road target detection affect the driving safety of autonomous vehicles. The complexity of actual traffic conditions and the presence of light sources, obstacles, and other interference factors make vehicles face great challenges in detection. Accurate road target detection is the primary challenge that autonomous vehicles need to address. Therefore, this paper adopts a two-stage detection method to improve the detection accuracy of road targets.

To enhance the accuracy of road target detection, this paper proposes an advanced algorithm based on Cascade R-CNN to boost the performance of the road object detector. Firstly, data preprocessing and data augmentation operations are introduced to generate more accurate prediction frames to tackle the issue of challenging data collection and inadequate sample size, which leads to unsatisfactory training results. Secondly, by adding an attention mechanism to the backbone network, merging SENet and ResNet to enhance the algorithm's feature extraction and classification ability. This solves the problems of low precision, potential overfitting, and high memory consumption in the original network. Finally, by using deformable convolution to handle vehicle targets with different shapes and variable sizes, and by training and predicting vehicle targets on multi-scale feature maps, the problem of poor target detection and localization due to the inability to visually identify and fine-locate targets is solved. To demonstrate the validity of our experiments, a series of ablation experiments and comparison experiments are designed and validated on the KITTI dataset and the SODA10M public dataset, and the generalization capability of the algorithm is demonstrated in this paper.

II. Related work

Whether the unmanned vehicle can accurately detect road objects has become the research focus. The two-stage target detection method based on deep learning is relatively superior in both detection accuracy and localization precision, so this paper takes the two-stage target detection method as the primary research direction. The two-stage target detection algorithm first selects the suggestion frames for the input image and then conducts classification and position regression on the suggestion frames, which leads to the final detection outcomes. In this chapter, some classical two-stage

objective methods are further detailed. It also discusses the fundamentals and working principles to provide theoretical support for the research in this paper.

A. CNN

The development of convolutional neural networks can be traced back to the 1960s when Hubel [12] and others introduced the concept of receptive fields through their studies of cat visual cortex cells. Subsequently, Marvin Minsky [13], a master of artificial intelligence, identified the shortcomings of perceptual machines. Since it is unable to handle heterogeneous networks, its computational power is not sufficient to handle large neural networks. By the 1980s, Fukushima [14] proposed the concept of a neurocognitive machine based on the concept of perceptual field and the idea of convolution and pooling. It can be considered as the first implementation network of a convolutional neural network. In 1986, Hinton [15] proposed the BP backpropagation algorithm, which is an ingenious application of the chain rule mainly to find a set of weights with minimum error. On the base of that, Yann Lecun [16] identified handwritten character features and compared them with standard handwritten digit recognition, and the result was that convolutional neural networks significantly outperformed other techniques. Therefore the LeNet-5 algorithm was proposed and the prototype of a convolutional neural network was generated by applying the BP algorithm to the training of this neural network structure. Until 2012, in the ImageNet image recognition competition, Hinton's [17] group proposed a new Alexnet algorithm, introducing deep structure and the dropout method, which subverted the image recognition field and made CNN start to dominate gradually in the computer vision field.

CNN (Convolutional Neural Network) is a feed-forward neural network composed of numerous artificial neurons and structured in different connection patterns. CNN consists of one or more convolution layers, pooling layers and a fully connected layer on top, which is mainly used to extract local features of convolution objects. It is a mathematical or computational model used to mimic the structure and function of biological neural networks. It mainly solves the problem that the amount of image data to be processed, the original features during the digitization process, and the accuracy of image processing is not high.

CNN consists of three main parts: convolutional layer, pooling layer, and fully connected layer. The main role of the convolutional layer is to extract features. The pooling layer mainly serves to downsample without corrupting the recognition results. The main role of the fully-connected layer is to classify. Convolutional neural networks can be analogous to human brain thinking. For instance, if you observe the image of a car in the picture below, how does the human brain extract the information from the picture and identify it as a car? Firstly, it needs to be judged that the car has a regular shaped shell. Secondly, it was found by observation that it has a series of tires, tail lights, mirrors, license plates, and other accessories of a car. Finally, these are linked and combined with previous knowledge and experience to determine that it is a car. The principles of CNN are also different. The schematic diagram is shown in Fig.1. The convolutional layer is used to find the vehicle features. The pooling layer reduces training with fewer parameters while keeping the sample unchanged, ignoring some interference or useless information. Finally, the fully connected layer is used to classify to determine that this is a car.

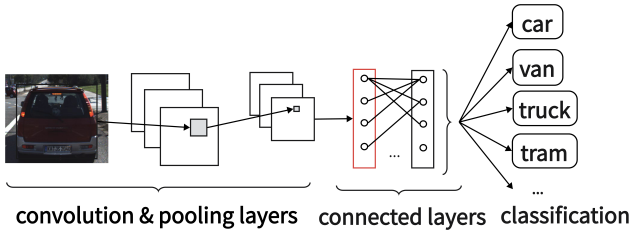

Fig. 1: Schematic diagram of CNN algorithm

B. R-CNN

The image classification task with CNN as the leading algorithm is one of the most fundamental tasks in computer vision. Image classification usually requires only the prediction of object classes. However, based on image classification, we need not only to classify the objects in the image but also to locate the position of the objects. In this way, another important task of computer vision is proposed: target detection. In the target detection task, a picture may contain multiple objects or even multiple objects of different categories in a single image. In this case, it is necessary to mark the position and size of each object and distinguish its category, and the exact location of the target is given, because the number and type of target in the image are variable. Obviously, the object detection task is more complex than the image classification task.

Girshick et al[18] first applied convolutional neural networks to the target detection task by combining candidate regions and CNNs in 2014. He proposed to use of deep convolutional networks as the backbone network for feature extraction. Combined with region selection methods to generate candidate regions and form an R-CNN architecture. This model becomes the basis of object detection algorithm based on deep learning, and has achieved great success in the field of object detection, laying the foundation for a series of subsequent detection algorithms. The R-CNN (Region-based Convolutional Neural Networks) algorithm consists of four main components, as illustrated in Figure 2. Firstly, the detected images are acquired through the network; secondly, the input images are extracted with approximately 2000 candidate regions using the Selective Search (SS) algorithm [19]. The candidate regions of various sizes are resized to a fixed size and sent to a CNN for feature extraction to acquire the features of the candidate regions. After obtaining the feature vectors, the class and location information of the target is obtained using Support Vector Machines (SVM) with multiple SVM classifiers and regressors. Finally, the bounding box is fine-tuned by discarding the regions with a high overlap ratio using non-maximum value suppression, and then the results are accurately detected using bounding box regression.

1) Selective search algorithm: The Selective Search algorithm first uses a segmentation tool that divides the image into one thousand to two thousand small regions. The similarity between the merged and adjacent regions is calculated

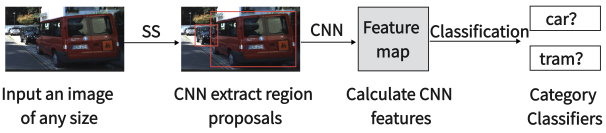

Fig. 2: Schematic diagram of R-CNN algorithm

by computing the similarity (e.g., color, texture, scale, etc.) between all neighboring regions and merging the regions with higher similarity together. Keep repeating the previous step until the whole image becomes one region.

2) SVM Classifier: When using SVM for classification, since SVM is limited to binary classification of data, and the majority of datasets are multi-category tasks. An SVM classifier is trained for each category (e.g., there are 20 categories in the Pascal VOC dataset, so 20 classifiers need to be used). In the SVM classification process, the Intersection over Union ratio (IOU) of the intersection of two bounding boxes is considered a negative case when IOU<0.3. When IOU>0.7, it is considered a positive case known as ground truth, indicating that the object is fully enclosed. All cases other than positive ones are discarded. Meanwhile, the SVM classifier also outputs a label that it predicts. The self-trained label is compared with the real label to calculate the training loss, and then the SVM continues to be trained. Since the number of candidate frames generated is far greater than the count of actual targets in the image, a large amount of candidate regions are overlapping, and therefore redundant candidate frames that need to be removed. The authors use the non-maximal suppression (NMS) method here to remove redundant bounding boxes so that some proposal regions with the highest scores in each category are obtained.

Since SVM is trained with small samples, it can be the case that there are far more negative samples than positive samples. For this scenario, the authors employ the hard negative mining method. The method initially utilizes all samples for training. After one round of training, the negative sample with the highest score, that is, the negative sample most likely to be misclassified, is added to the new sample training set for further training. The above steps are repeated until the stopping condition is reached such that the classifier performance no longer improves. It makes SVM suitable for small sample training without overfitting when the samples are unbalanced. Compared with traditional sliding window based target detection algorithms, R-CNN algorithm has improved its accuracy and achieved better results on major standard data sets. However, due to the cropping deformation of the proposed region, some feature information in the image will be lost, and the position information of the target will be distorted, so the detection accuracy will be indirectly affected. At the same time, each image requires convolution operations on approximately 2000 proposal regions, which increases the amount of computational redundancy and makes the target detection slow.

C. Cascade R-CNN

Typically, the object detection task is performed within a network where the image features are extracted using a deep neural network and subsequently identified based on feature utilization. The one-stage technique is less robust because it is more vulnerable to feature fluctuations, which leads to unsatisfactory regression findings. Cascade RCNN [20] provides a multi-detector model with a cascade structure to solve the lack of detection performance in the first stage and achieves substantial improvement, particularly in detection at high thresholds.

The Cascade R-CNN network improves upon the Faster R-CNN [21] network by including three major components: the ResNet feature extraction network [22], the Feature Pyramid Networks (FPN) [23], and the cascade detector. ResNet is utilized in this scenario to extract features and perform a multi-scale fusion of feature maps from deep to shallow layers. The fused feature maps are then sent to the RPN, which generates potential target areas.

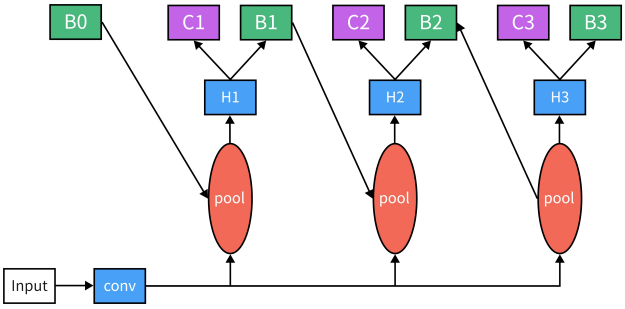

Fig. 3: Cascade R-CNN structure schematic

The Cascade R-CNN structure is shown schematically in Fig.3. Where $conv$ stands for the convolutional neural network and B_0 denotes the proposal region selected from the region recommendation network. The RoI Pooling layer utilizes the proposed regions and feature maps from the convolutional neural network to extract the characteristics of the region of interest. The fully connected layer H_i is then given the features. The classifier C_i and the border regression function B_i for precise localization, respectively, receive the characteristics generated by the fully connected layer.

Each bounding box in the general regression tasks has $b = (b_x, b_y, b_w, b_h)$. The process of regression is the progression of the proposed frame's coordinate data b towards the ideal potential frame g. The formula for this step is $f(x, b)$. This regression's distance may be represented as $\Delta = (\delta_x, \delta_y, \delta_w, \delta_h)$. The regression process is defined as Eq.(1).

$$f(x, b) = f_T \circ f_{T-1} \circ \cdots \circ f_1(x, b) \tag{1}$$

Where T represents the number of cascades, x denotes the input image, and b indicates the bounding box corresponding to the image corresponds. The coordinates are normalized by the same technique, as illustrated in Eq.(2), to eliminate the influence of the regression scale caused by the bounding box's size and location. as Eq.(1).

$$\begin{aligned} \delta_x &= (g_x - b_x)/b_w, \quad \delta_y = \left(g_y - b_y\right)/b_h \\ \delta_w &= \log\left(g_w/b_w\right), \quad \delta_h = \log\left(g_h/b_h\right) \end{aligned} \tag{2}$$

A cascade detector is a model composed of a series of models that process the output of one another in succession. Three cascaded detectors make up the cascade model in Cascade R-CNN. Objects that are easiest to recognize are found by the first detector. To make sure that the identified objective

is a real goal and not a false alarm, it often employs a high threshold value. Targets that are missed by the first detector must be found by the second detector. To produce results for detection that are more precise, the output of the first cascade head is further refined and filtered. In order to allow for the detection of more objectives, it often employs a lower threshold value. The third detector's role is to find targets that the second detector is unable to find. To ensure that the most challenging targets are found and the desired outcome is generated, it uses the lowest threshold value.

$$F \in \mathbb{R}^{C \times H \times W} \tag{3}$$

$$P_c^1 = \{p_c^1 \mid r\} \tag{4}$$

$$T_x^1, T_y^1, T_W^1, T_h^1 = \{t_x^1, t_y^1, t_W^1, t_h^1 \mid r\} \tag{5}$$

The original image and the suggested frame serve as detector inputs. Where r is the input proposal frame, $C, H, and\ W$ are the number of channels, height, and width, respectively, and F is the feature map extracted from the original image. Each proposal's category score and coordinate offset, or classification and regression, are the outputs. The relevant equation is shown in Eq.(3)-(5). Where P_c^1 is the category score of the predicted proposal, $T_x^1, T_y^1, T_w^1, and\ T_h^1$ are the offsets of the coordinates of the predicted proposal. The second and third cascade detectors are similar to the first detector and will not be described here. The overfitting and mismatching issues caused by merely increasing the IOU threshold in a convolutional neural network are resolved using this pattern of cascading detectors.

Cascade R-CNN improves the R-CNN detection network in Faster R-CNN to a cascade detection network that improves the IoU threshold of the candidate box layer by layer, which improves the sample quality, and then improves the detection accuracy compared with Faster R-CNN, but there are still vehicles missed detection, which cannot meet the demand for high quality detection accuracy of vehicle targets. Therefore, how to further improve the accuracy of vehicle recognition is a worthwhile research direction.

III. Improved SED-CRCNN method

Although deep learning-based road objective methods have gained a lot of interest from academics, the present algorithms have certain limitations. For example, it may result in overfitting, where accuracy may become insufficient after numerous upgrades, and the deployment of deeper networks may result in increased memory utilization, among other things. This research offers a deep learning cascade network (SED-CRCNN) based on an enhanced Cascade R-CNN to overcome these concerns. To get better detection results, the algorithm first executes several preprocessing actions on the data. Meanwhile, the backbone network ResNet is upgraded by introducing the SENet module, which allows the network to modify channel weights adaptively, thereby boosting model performance. Furthermore, the deformable convolution approach is used with Cascade RCNN to add an offset variable to the position of each sample point in the convolution kernel, thereby overcoming the limitations of classical convolution.

A. Data preprocessing

1) Data enhancement: By making minor adjustments to an existing dataset or generating artificially generated data from existing data, data augmentation techniques expand the quantity of data available. The technique is commonly utilized in deep learning, particularly for migration learning and small sample learning tasks. The main purpose of the method:

- Increasing training data: In object detection tasks, there is a lack of training data because it is difficult to gather the data or there aren't enough samples, which leads to subpar training outcomes. This is where using data augmentation can produce extra data and enhance the model's training effect.
- Enhance model generalization: Overfitting is a common problem in deep learning, and data augmentation can be used to better understand the distribution of the data and enhance model generalization.
- Improving data distribution: Data improvement can broaden the sample's variety and improve the distribution of the dataset's data, which are frequently distributed inequitably.

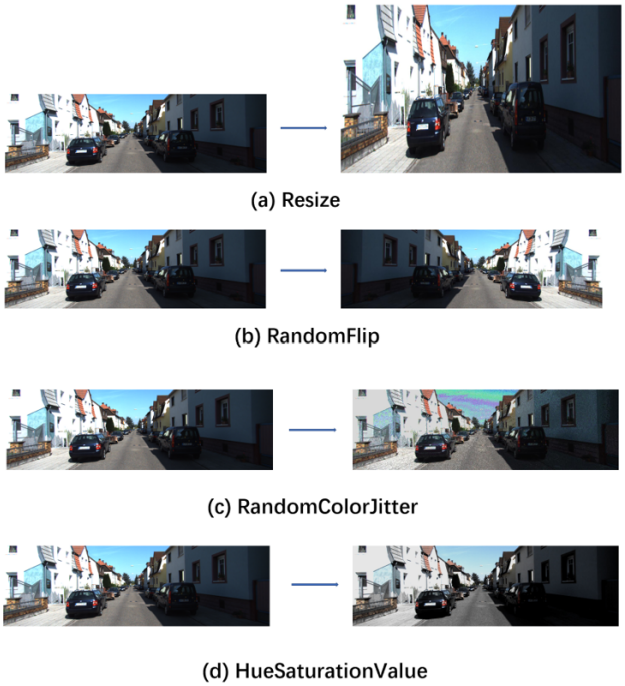

Fig. 4: Different data enhancement methods

In this study, the dataset is augmented by mirror flipping, modifying the hue, saturation, and brightness of the image, changing the color and brightness attributes of the image, and other gain acquisitions. A training set and validation set are created from the data-enhanced photos in a 7:3 ratio. The outcomes of utilizing various approaches for data augmentation to raise the caliber and diversity of the data are shown in Fig.4. The Resize procedure in (a) increases the image size from 1242×375 to 1333×800, bringing attention to minute target details and making the target item obvious. By flipping the image randomly up, down, and left to right, the RandomFlip operation in (b) enhances the training data.

By altering factors like the hue, saturation, and brightness of the image, RandomColorJitter and HueSaturationValue in (c) and (d) are employed to increase the generalizability of the model.

2) Create more accurate prediction boxes: Three aspect ratios, 1:1, 1:2, and 2:1, are frequently utilized to create anchor frames in the Cascade R-CNN model. However, due to the different attributes of images in different datasets, using the same aspect ratio leads to a large number of anchored frames that are formally mismatched, which reduces the accuracy of boundary regression. As a result, anchor frames with the appropriate aspect ratio dimensions were found through additional analysis of the experimental data collected in this article.

This study first examines the image data of all datasets to understand the dimensions of the images. Then, statistical analysis of the aspect ratio was performed by calculating the aspect ratio of each image. The data are finally represented graphically as seen in Fig.5. The statistical aspect ratio's size is plotted along the horizontal axis, and the number of visible images is plotted along the vertical axis. It can be seen that this dataset has a concentration of image aspect ratio sizes, so the results were modified accordingly. Without altering the network structure or the amount of parameters, this technique can increase the model's detection.

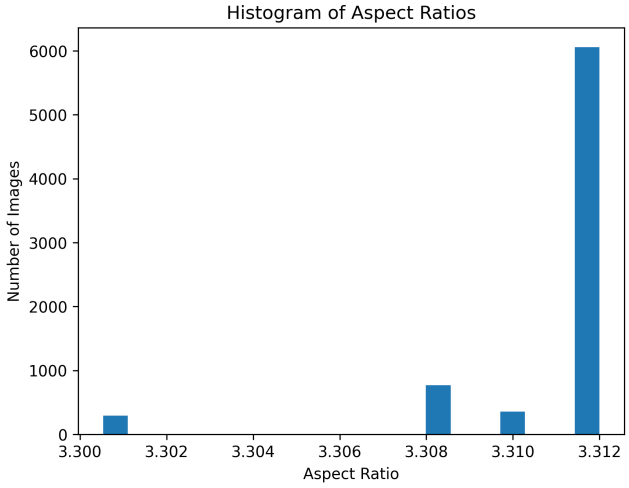

Fig. 5: Histogram of aspect ratios

B. SE-ResNet

Based on the concept described in the section before, a cascaded R-CNN detection module was proposed for object detection tasks, with the backbone module typically using the ResNet method. ResNet is a deep convolutional neural network model that introduces the residual connection to address the degradation issue with deep convolutional neural networks. Road object detection approaches that combine Cascade R-CNN and ResNet algorithms can detect people and vehicles more rapidly and precisely, but there are still some drawbacks. In the beginning, training on tiny datasets may lead to overfitting. Secondly, although cascaded detectors and residual networks have improved accuracy, it is still not very high. Finally, as the residual join necessitates the saving of additional intermediate feature maps, it could have a high memory footprint.

To address the drawbacks of combining the Cascade R-CNN and ResNet algorithms, we improve the backbone network of the Cascade R-CNN algorithm by introducing the SENet [24] method. SENet mainly consists of two modules, Squeeze and Excitation. Squeeze is a global average pooling layer, which extracts statistical information of each channel by feature map to perform global average pooling to extract statistical information about the channel. This generates a feature vector where each element represents the global information associated with that channel. The Excitation operation, on the other hand, is a neural network consisting of multiple fully connected layers. It generates a channel-specific importance weight vector by receiving the vector output from the Squeeze operation. This weight vector is scaled and normalized by an activation function and multiplied with the original feature map into elements. This will allow the SENet module to selectively enhance useful feature channels and suppress unnecessary ones.

The SENet method was combined with the ResNet method to obtain the SE-ResNet method for better detection results. SENet (Squeeze-and-Excitation Network) is a deep neural network model based on the development and research of ResNet, which further improves the performance of the model by introducing the attention mechanism. Its overall architecture embeds Squeeze and Excitation operations into a deep convolutional neural network, which enables the network to adaptively adjust the weights of the channels to improve the performance of the model. SENet uses the Squeeze-and-Excitation module to adaptively adjust the importance of each channel to better capture the correlation between different channels and improve the model's performance. The correlation between them improves the classification accuracy of the model. Combining SENet with ResNet can further enhance the feature extraction and classification capabilities of ResNet, thus further improving the classification accuracy.

To understand how the SENet functions in its totality, let's first suppose that there is a feature map X with the dimensions $H \times W \times C$. Where C denotes the number of channels, and H and W, respectively, denote height and breadth. With the Squeeze operation, the global average pooling of the ResNet-processed feature maps is accomplished to produce a feature tensor Z of size $1 \times 1 \times C$. As shown in Eq.(6).

$$Z = \text{GlobalAvgPool}(X) \tag{6}$$

The output vector F and a Bottleneck structure, which models the correlation between channels, are created by passing the input feature tensor Z through two tiers of completely linked layers. To create the vector S, the components in F are normalized using the Softmax activation function. To create a weight vector V (as illustrated in Eq.(7)–(9)), multiply each element in S by a weighting factor W.

$$F = \text{FC}(Z) \tag{7}$$

$$S = \text{Softmax}(F) \tag{8}$$

$$V = S \odot W \tag{9}$$

The final step is to multiply the initial feature map X by the weight vector V that is produced in order to create an updated feature map Y. Each channel's characteristics are

weighted according to the normalized weights. As shown in Eq.(10).

$$Y = V \odot X \tag{10}$$

A comparison of the ResNet and SE-ResNet structures is shown in Fig. 6. SE-ResNet automatically obtains the importance of each channel mainly by modeling the channel relationships. According to this degree of relevance, the network's performance is subsequently enhanced by improved beneficial qualities and suppressing those that are insignificant to the job at hand. By balancing and improving the feature dimension using a specially developed feature compression activation module, SE-ResNet enhances the model's capacity for learning new features. It enhances the network's categorization precision and enables a better grasp of the relationships between various channels.

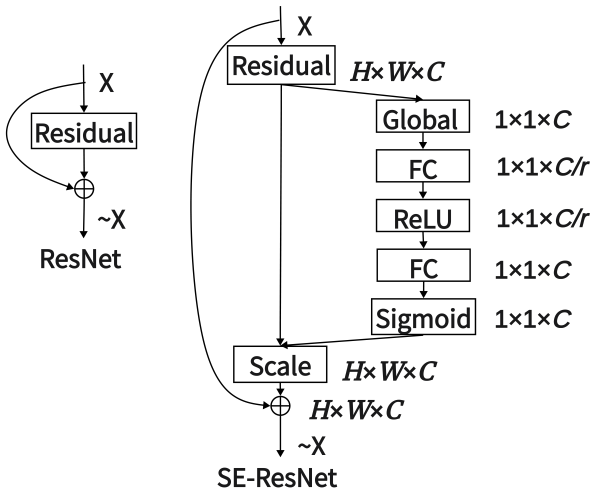

Fig. 6: Comparison of ResNet and SE-ResNet structures

C. Deformable convolution

The modeling of significant unknown changes is the natural limitation of neural networks. The CNN module's fixed geometric structure and the absence of internal methods to manage geometric alterations are the causes of this constraint [25]. As a result, the receptive field size is the same for all activation units in the same CNN layer. It renders it impossible to visually recognize things that need precise positioning and leads to inaccurate target localization. Due to the diversity of vehicles and pedestrians, as well as the wide range of geometries, determining the size and shape of each vehicle in tasks involving road object detection is more challenging. Since it is difficult to adjust the vehicle target recognition task by using the fixed size convolution kernel directly, deformable convolution is proposed to adapt to the changing target vehicle geometry. It can enhance spatial transformation invariance, collect more semantic information, and enhance feature extraction capabilities to better handle features of various sizes and forms.

The DCN (Deformable Convolutional Networks) algorithm is a crucial technique for resolving target identification object deformation issues. By using Deformable Convolution, the convolution kernel can change the position and size of its samples to fit the shape and size of various target objects, which solves the "object deformation" problem in target detection. As a result, detection accuracy is effectively improved, and the network is better able to adapt to the form and position of objects in a range of challenging settings.

A schematic representation of the sampling sites for both traditional and deformable convolution is shown in Fig.7. Figure (a) demonstrates that the traditional convolution only has one sample network with a set rectangular form. Figure (b) demonstrates how the convolution kernel adds an offset variable to the location of each sampling point so that the kernel may accomplish random sampling around the present position and overcome the constraints of conventional convolution. Eq.(11) demonstrates how to determine the offset

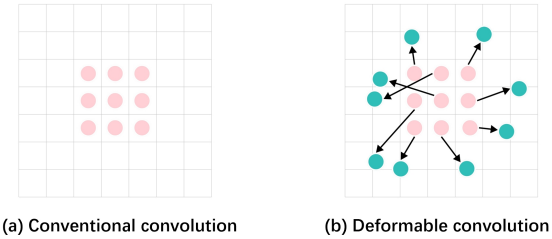

Fig. 7: Conventional and Deformable Convolution

$\Delta p_{i,j}$ for each feature point (i, j) of the input feature map.

$$\Delta p_{i,j} = \begin{bmatrix} \Delta p_{i,j}^{x} \\ \Delta p_{i,j}^{y} \end{bmatrix} = f_{\Delta}\left(X_{i,j}\right) \tag{11}$$

Each place on the feature map y should be denoted by $y(p_0)$, where R represents the nine positions of the sampled points of the convolution kernel concerning $x(p_0)$ in the input. The convolution result of standard convolution is a "weighted sum" operation between the convolution kernel and the input features, as shown in Eq.(12). The deformable convolution introduces a learning displacement Δp_n, the implementation process chooses the sampling position using bilinear interpolation, and the convolution process becomes Eq.(13). This allows for the extraction of features that more closely fit the shape of the target. Where p_0 is the feature sampling location, Δp_n denotes the offset, p_n is an enumeration of the convolution sampling location $\mathcal{R}$, w is the feature weight and x is the eigenvalue expression.

$$y(p_0) = \sum_{p_n \in \mathcal{R}} w(p_n) \cdot x(p_0 + p_n) \tag{12}$$

$$y(p_0) = \sum_{p_n \in \mathcal{R}} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \tag{13}$$

The structure of the cascaded R-CNN and deformable convolution combination used in the deformable convolution algorithm of this paper is shown in Fig.8. To create feature maps for the C3, C4, and C5 levels of the Feature Pyramid Network (FPN), the input images are routed through the backbone network. Where C3, C4, and C5 stand for the output feature maps of SE-ResNet network stages 3, 4, and 5, respectively. A cascade head network is given each feature map to process. A standard convolutional layer is first applied to each proposal region in order to extract features and create a feature map. The Deformable Convolution module uses the feature map as an input and performs a deformable

convolution operation. Following the deformable convolution procedure, ROI Align maps the ROI region and its contextual data into identically sized rectangular rectangles. After that, by enlarging the ROI region, the fusion is carried out via an additive process to gather contextual information.

D. Algorithm Overall Structure

1) Structure: As stated above, this study proposes an enhanced SED-CRCNN technique based on cascaded R-CNN, where the inputs are preprocessed, added to a ResNet backbone network, and deformed convolution operations are performed. The main flow chart of the algorithm is shown in Fig.9.

The following phases make up the algorithm's main process: The image dataset is initially built once the raw picture data has been pre-processed. Second, a certain ratio is used to partition the picture collection into training, validation, and test sets. The training and validation sets were run through the SE-ResNet component of the modified SED-CRCNN algorithm. The ResNet technique is enhanced with a SENet module to perform more precise feature extraction. After the SE-ResNet output, deformable convolution procedures were employed in layers 3 to 5 to enhance the model's perceptual field, producing three feature maps. Following that, Top-down Path and FPN procedures combine the three feature maps into a high-resolution feature pyramid. Then, using RPN, bounding boxes are extracted, those with boundaries are subjected to ROI Align procedures, and they are finally transformed into fixed-size feature maps. Subsequently, classification and regression operations are applied to each candidate bounding box to obtain the detection box. Finally, it is determined whether the accuracy of the trained model satisfies the requirements based on the outcomes of the training set, validation set, and test set following model training. If it is satisfied, the model is output, the target road traffic is detected, and the detection result is produced. If it is not satisfied, the model is sent back to the training section for training. The algorithm structure of SED-CRCNN is shown in Algorithm 1.

2) Loss function: The RPN network loss function, the ROI network loss function, and the cascade network loss function make up the three primary parts of the upgraded network's loss function.

(1) RPN network loss function: The loss function of the RPN network is a linear combination of the classification and regression components, as shown in Eq.(14). L_{cls} and L_{reg} denote the loss functions for the dichotomous classification task and the bounding box regression task, respectively.

$$L_{rpn} = L_{cls} + \lambda L_{reg} \tag{14}$$

The quantity and placement of bounding boxes are managed by the classification section, and it also determines whether a candidate box includes a target item. This section makes use of the CrossEntropy Loss function, which employs a sigmoid function to transform the network's output into a probability. The classification loss function is represented in Eq.(15), where N_{cls} is the total number of positive and negative samples in the training samples and y_i is a binary variable indicating whether the ith sample is the target. If $y_i = 1$, it falls within this classification; if $y_i = 0$, it does not. The probability that the ith sample will be the target is shown by the symbol p_i.

Algorithm 1 SED-CRCNN method

Input: Image data
Output: List of final detected objects
1: Feature Extraction using SEResNet
2: Region Proposal Network (RPN) for candidate bounding boxes
3: Region of Interest (ROI) Alignment
4: Classification and Bounding Box Regression Heads
5: Stage1:Classify and regress over candidate bounding boxes
6: Apply Non-Maximum Suppression (NMS) for Stage 1 detections
7: Stage2:Downsampling of Stage 1 detections
8: Feature Extraction using SEResNet
9: ROI Alignment
10: Classification and Bounding Box Regression Heads for Stage 2
11: Classify and regress over downsampled detections
12: Apply NMS for final detections
13: Repeat the above until the number of layers is complete
14: Output the final result

$$L_{cls} = -\frac{1}{N_{cls}} \sum_{i}^{n} [y_i \log p_i + (1 - y_i) \log(1 - p_i)] \tag{15}$$

The other part is the regression loss function, which measures the matching degree between the candidate box and the ground-truth box by calculating their intersection-over-union (IoU) ratio. Eq.(16) demonstrates the formula, where N_{reg} is the number of positive samples in the training sample and t_i and t_i^* stand for the predicted bounding box and the matching genuine bounding box, respectively, in terms of coordinate values. Here,a loss function called SmoothL1 is adopted, which has a lower outlier penalty and is therefore more suitable for object detection regression work. The model becomes more stable as a result of the smoothing, which also stops significant errors from having a negative impact on training. The equation is shown in Eq.(17), where x represents the difference between the predicted and true boxes.

$$L_{reg} = \frac{1}{N_{reg}} \sum_{i}^{n} y_i \text{smooth}_{L1}(t_i - t_i^*) \tag{16}$$

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2 & \text{if} \mid x \mid < 1 \\ \mid x \mid -0.5 & \text{otherwise} \end{cases} \tag{17}$$

(2) ROI network loss function: Similar to the RPN network, the ROI network's loss function is divided into two main parts: the objective classification loss function and the bbox regression loss function. CrossEntropy Loss function is utilized by the target classification loss function to determine the likelihood that a candidate frame contains a target item. A SmoothL1Loss function is utilized in the bbox regression loss function to determine how to modify bounding boxes to resemble the target item more closely.

(3) Cascade network loss function: The upgraded algorithm's cascade detector employs many sub-detectors to increase detection precision. The cascade detector's loss

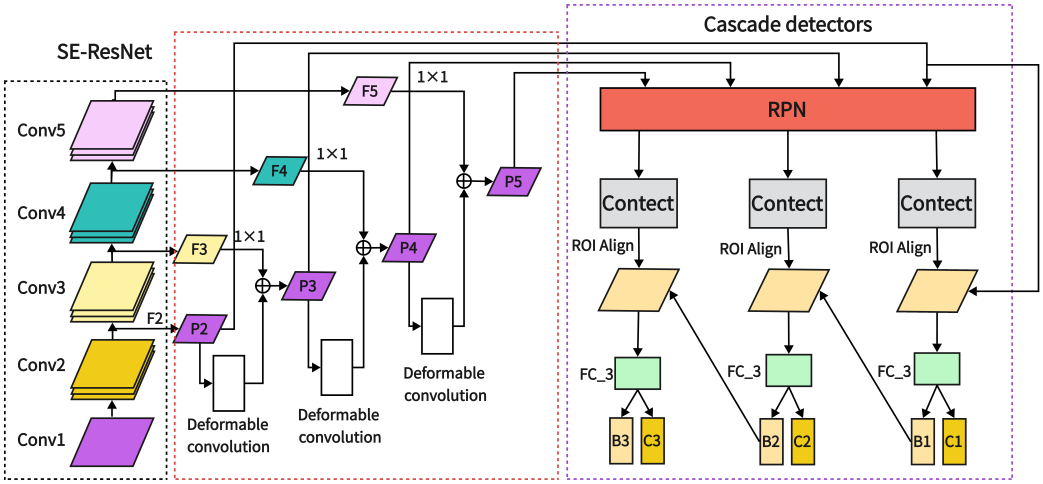

Fig. 8: Cascade R-CNN and deformable convolution combined schematic

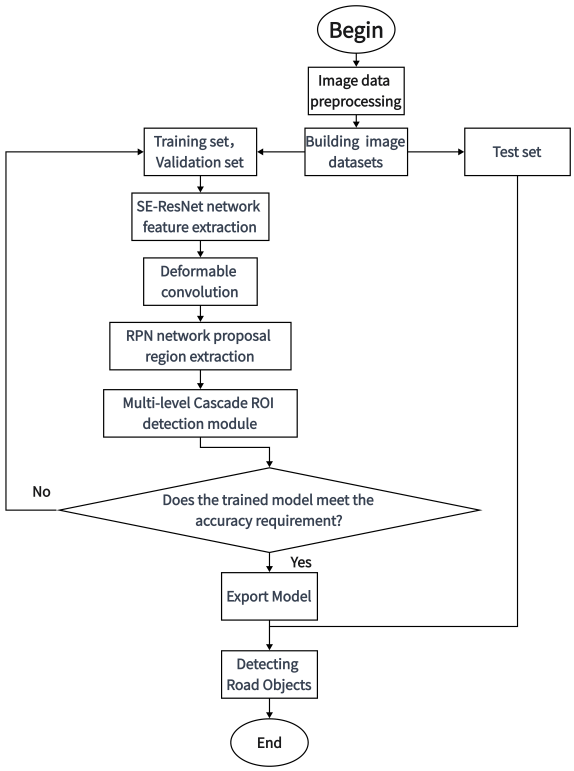

Fig. 9: Overall project flowchart

function is divided into many sub-detector loss functions, each of which includes a target classification loss function and a bbox regression loss function for figuring out how to modify bounding boxes to resemble the target object more closely.

IV. Experiment

This research proposes a Cascade R-CNN based cascade network (SED-CRCNN) model to enhance the accuracy of road target recognition, particularly based on small objects and targets with irregular sizes and shapes. In this section, thorough comparison and ablation experiments are carried out on the road target open datasets KITTI and SODA10M to verify the efficacy of the SED-CRCNN model suggested in this study. The paper will go into further detail on the experimental evaluation metrics, the experimental data set, the experimental environment and parameter settings, the comparative experimental outcomes, the experimental technique, and the analysis of the results in the next subsections.

A. Evaluation metrics

Evaluation metrics are an important basis for evaluating how well an object detection algorithm approach works. The object detection task requires not only the identification of the object class but also the prediction of its location, so the selection of suitable evaluation criteria is essential to measure the performance of the model.

The sample, which is frequently classified as a positive sample and a negative sample, is a crucial notion in object detection evaluation. Objects to be detected are represented by positive samples, whereas targets not to be detected are represented by negative samples. For example, car or pedestrian targets are positive samples for detecting roadway targets, while other elements are negative samples. The confusion matrix is shown in Table 1. Where TP (True Positive) is the probability of detecting a positive sample, FP (False Positive) is the probability of detecting an incorrect sample, FN (False Negative) is the probability of not detecting a correct sample and TN (True Negative) is the probability of detecting a negative sample.

TABLE I: Confusion matrix

Confusion matrix		Ground truth	
		Positive	Negative
Predicted value	Positive	TP	FP
	Negative	TP	FP

In this paper, AP (single class average precision), mAP values, and AP50 values will be used as metrics for the evaluation of vehicle target detection methods, and Recall-Precision curves will be plotted to evaluate the performance of the detection methods. The magnitude of the AP value is determined by the region enclosed by the RP curve; the greater the area, the higher the AP value and the more

accurate the detection. Eq.(18)-(21) illustrate this. Where c represents the total number of identified target classes and AP_i is the typical accuracy of the ith target class's detection.

$$AP = \frac{TP + TN}{TP + TN + FP + FN} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

$$mAP = \frac{\sum_{i=1}^{\varepsilon} AP_i}{c} \times 100\% \tag{21}$$

The steps to calculate the AP metric are:

(1) Compare the IoU values of each prediction box in the test set with all of the true boxes to determine which true box has the highest IoU value.

(2) Determine the Precision and Recall for each prediction box individually, ranking them in order of highest to lowest confidence scores, and note the appropriate Precision and confidence score thresholds.

(3) The AP50 was determined using the above-ranked Precision-Recall curve, taking into account only predictor frames with IoU values larger than or equal to 0.5.

B. Dataset

1) KITTI: The KITTI dataset [26] is a joint venture between the Karlsruhe Institute of Technology in Germany and the Toyota Institute of Technology USA for algorithm evaluation datasets in autonomous driving scenarios. It is a publicly available dataset for vehicle object detection and consists of nine categories Car, Van, Truck, Pedestrian, Person_sitting, Cyclist, Tram, Misc, and DontCare for studies such as deep learning model computation. DontCare indicates that certain areas are targeted, but for some reason, such as being too far away from LiDAR. It contains mainly real image data collected from urban, rural, and motorway scenes. With varied degrees of occlusion and truncation, each image may contain up to 15 automobiles and 30 pedestrians. The center point coordinates, detection frame coordinates, navigation angles, and occlusion truncation information make up the annotation information.

The KITTI dataset is divided as shown in Table 2 and images of parts of the dataset are shown in Fig.10. The road target detection algorithm used in this paper uses the KITTI dataset to train the model, and the results are only retained for seven categories related to road target detection, namely Car, Van, Truck, Pedestrian, Person_sitting, Cyclist and Tram, without going into the results of other categories.

TABLE II: KITTI dataset partitioning

Dataset	KITTI
Year built	2009
Number of images	7481
Image size	1242 × 375
Image type	7
Training set	5984
Test set	1497

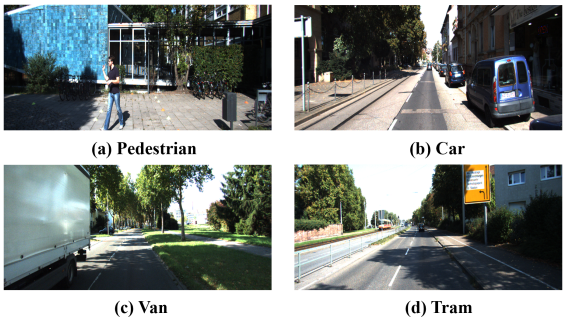

Fig. 10: Images in the KITTI dataset

2) SODA10M: SODA10M (Scenes, Objects, Depth and Annotations) [27] is a new generation of self-supervised 2D benchmark datasets published by Huawei Noah's Ark Lab in collaboration with Sun Yat-sen University. It includes 20,000 tagged photographs gathered from 32 cities and 10 million unlabeled images of various traffic scenarios. It is used to advance the study of autonomous driving researchers in relevant fields involving semi-supervised and self-supervised driving, as well as to jointly support the development of an ecosystem for autonomous driving. It is a high-quality driving scenario dataset that has attracted a lot of interest and recognition in the field of road object detection.

For the study, this report primarily employs 20,000 labeled photos. The SODA10M dataset includes the six primary types of pedestrian, cyclist, car, truck, tram, and tricycle, which are briefly summarized in Fig.11.

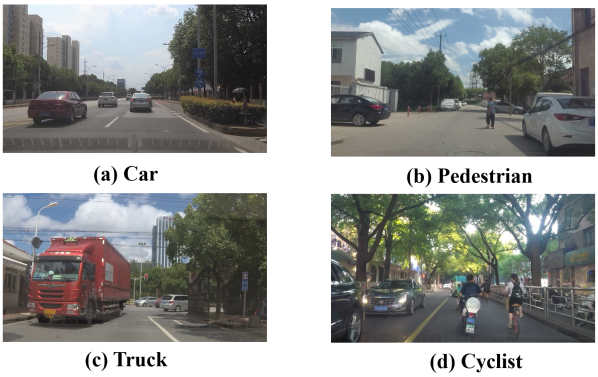

Fig. 11: Images in the SODA10M dataset

C. Experimental environment and parameter settings

1) Experimental environment: The experimental environment has an RTX 3090 (24GB) GPU and an Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz. The experiments are run on the Ubuntu 18.04.5 operating system with Python 3.9 as the coding language version. The deep learning framework uses Pytorch, which also uses OpenCV, Matplotlib, Numpy, and other toolkits for graphical analysis and processing, visualization, and presentation of the experimental data.

2) Training parameter: The momentum is set to 0.9 and the decay coefficient is set to 0.0001 during the training. The initial learning rate is set to 0.02, and the learning rate is adjusted to 0.002 and 0.0002 when the epoch is 9,12 respectively.

D. Experimental results

1) Ablation experiment: We will explore the effectiveness of each of the introduced modules through experimental results in this subsection. Using the KITTI dataset as an example, the corresponding modules are added to the previous improved algorithm separately. The experimental results of the AP values for each category are used to verify whether they are effective in a practical sense and whether they can achieve the theoretical results.

On the KITTI dataset, tests were performed using the original SE-ResNet and ResNet methods. The findings of the comparison are depicted in Fig.12 using the AP values for each category in the experiments that were recorded and analyzed. The AP values for the Car, Van, Truck, and Tram barely changed. Person_sitting, Cyclist, and Pedestrian all saw a discernible rise in AP values. This is because by incorporating SENet into ResNet, the network can better understand changes in the shape and size of targets. At the same time, the network was able to adjust the weights of the various functional channels to focus more on feature-related targets and fewer features. The introduction of the SENet post-attention mechanism can aid the network in better understanding the significance and relationship between various features and provide a more glaring improvement to the algorithm. Pedestrian, Person_sitting, and Cyclist are three types of target objects, and their forms and sizes vary greatly.

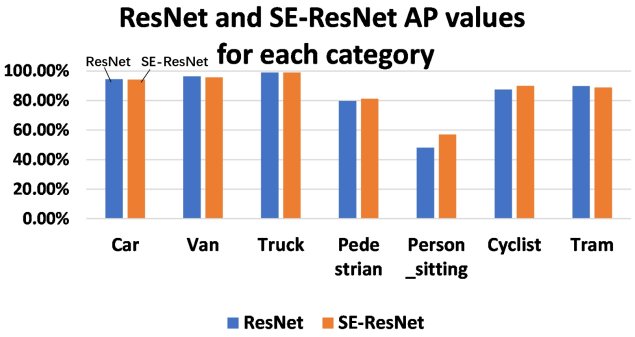

Fig. 12: ResNet and SE-ResNet AP values for each category

By comparing the original photos to the data-enhanced images, the trials were evaluated using the KITTI dataset. The findings of the comparison are depicted in Fig.13, using the AP values for each category in the experiment that were recorded and analyzed. All other classes have similarly better outcomes, with the exception of the Person_sitting class. Due to the limited training data in the KITTI dataset and the fact that the targets in these tasks can vary greatly in location, size, and shape. Therefore the use of data augmentation techniques allows the network to adapt to various target transformations more efficiently, improving the performance and robustness of the model.

On the KITTI dataset, the experiments compare the improved SED-CRCNN method with the cascaded R-CNN method. For comparison, the AP values for each category in the experiments are noted and assessed. As you can see from Figure 14, each class displays results to a similar degree. This is because deformable convolution methods can be used to handle objects with unusual shapes or sizes very

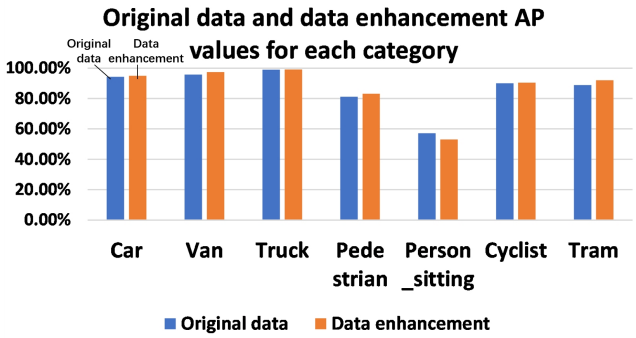

Fig. 13: Original data and data enhancement AP values for each category

well when performing object recognition tasks. To better accommodate various target forms and size changes, it may dynamically modify the convolution kernel's size and shape. The accuracy and robustness of detection can be increased by better capturing the features and detailed information of the target.

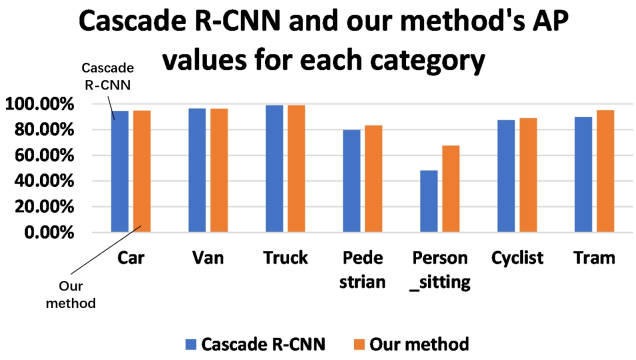

Fig. 14: Cascade R-CNN and our method's AP values for each category

The Cascade R-CNN with ResNet as the backbone network was used as the experimental benchmark for ablation experiments to verify the effectiveness of each module. The results of the relevant ablation experiments are shown in Table 3. With a 1.4% rise in its mAP value, the gains in the first and second rows of the chart demonstrate how successfully converting the ResNet to a SE-ResNet backbone network may boost detection performance. Secondly, the experimental outcomes in the third row of the table, which contains a new data augmentation module over the second row, have gone up somewhat, by 0.8 percentage points. The findings eventually achieved their maximal value once the DCN module was reintroduced, at which point the matching three modules were fully introduced. The approach described in this research is 3.7% better overall than the original Cascade R-CNN method, producing superior experimental results.

2) Experimental results on the KITTI dataset: With more training sessions and lower losses, the SED-CRCNN algorithm eventually converges on the KITTI dataset. The loss value stabilizes around 0.3 after 8000 iterations of the loss function curve for the KITTI dataset. The model has now reached its ideal condition. Fig.15 displays the curve for

TABLE III: Ablation experiment

Proposed Method				
Cascade R-CNN	+SE-ResNet	+Data enhancement	+DCN	mAP(%)
✓				79.6
✓	✓			81.0
✓	✓	✓		81.8
✓	✓	✓	✓	83.3

the loss function, where loss_rpn_bbox is the regression loss function of RPN, and loss_rpn_cls is the classification loss function of RPN. s0, s1 and s2 represent the loss functions of the three cascade detectors respectively. s0.loss_cls denotes the classification loss function of the first cascade detector, s0.loss_bbox denotes the regression loss function of the first cascade detector and so on.

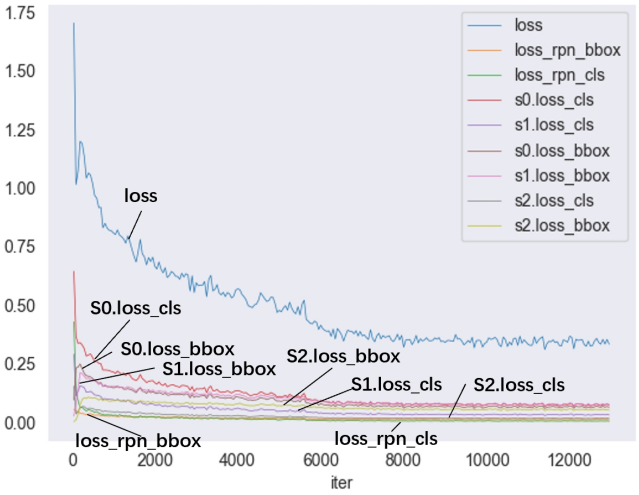

Fig. 15: Loss function curves

Fig.16 depicts the mAP value variation curve with epoch, where the vertical axis represents the mAP value for each round and the horizontal axis represents the exact number of epoch rounds. This figure makes it evident that the mAP value trend during the model's training process is progressively growing. As a consequence, the model is deemed to be well-trained, and the ideal model and outcomes are afterward established. To verify the effectiveness of the SED-CRCNN method proposed in this paper, the improved method was compared with other methods. The corresponding AP values for all categories, the total mAP values, and AP50 metrics are also listed, as shown in Table 4. In order to be more comparative, some representative methods are selected for comparison experiments in this paper. These methods include CornerNet, the classical algorithm for anchorless frames, the CRPN, which uses a cascade approach, and FCOS, the classical algorithm for one-stage detection. In addition, an enhanced Cascade R-CNN approach based on the very successful two-stage Faster R-CNN algorithm is included. All techniques were evaluated on the KITTI validation set after being trained on the KITTI training set.

A comparison of the experimental results shows that our algorithm outperforms other methods in terms of average accuracy performance when using an input image of 1333×600 size. Our study methodology offers a significant benefit in

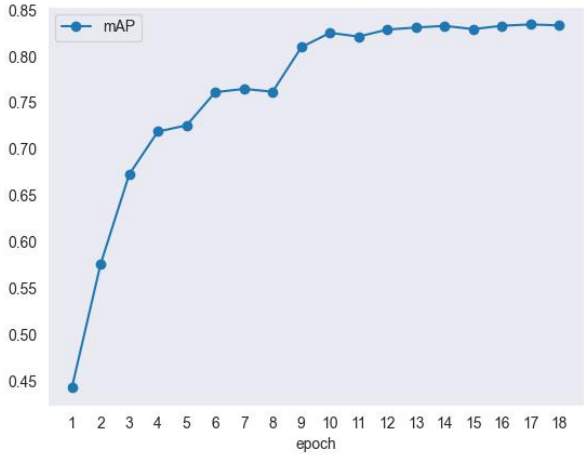

Fig. 16: mAP curve

terms of improved performance. It performs well across the board and obtains the best overall result of all the techniques evaluated, even though it cannot attain the greatest performance on every categorization.

Fig.17 displays as bar charts the detection results obtained using our approach, the Faster R-CNN method, and the Cascade R-CNN method that is based on the Faster R-CNN improvement. Its visualization displays the varying numbers of AP values for each category as they increase and decrease. Our algorithm, which plainly performs better than the other two approaches, is one of the grey bars.

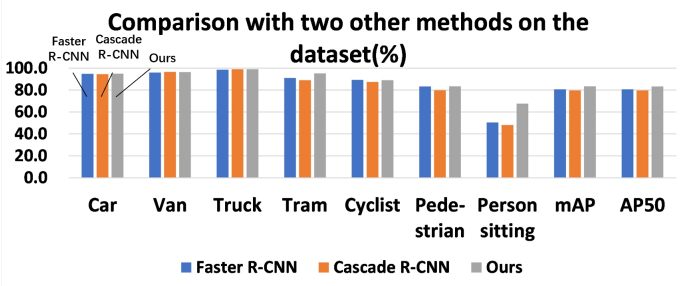

Fig. 17: Comparison with two other methods on the dataset(%)

Fig.18 shows the detection results of the Cascade R-CNN algorithm before improvement and our SED-CRCNN algorithm after improvement on the KITTI dataset. Where (a) shows the original image in the dataset, (b) is the result after Cascade R-CNN detection, and (c) is the result after our improved algorithm detection.

TABLE IV: Comparison with other methods on the KITTI dataset (%)

Method	Car	Van	Truck	Tram	Cyclist	Pedestrian	Person_sitting	mAP
CornerNet	82.9	91.6	84.2	91.7	82.9	75.7	48.9	74.35
CRPN	95.7	96.9	99.1	93.7	89.5	83.3	55.8	82.10
FCOS	94.8	96.2	98.6	91.4	89.1	82.4	47.7	80.30
Faster R-CNN	94.8	96.0	98.5	91.1	89.3	83.2	50.5	80.61
Cascade R-CNN	94.5	96.5	99.0	89.0	87.5	79.8	48.2	79.56
Ours	94.9	96.4	99.0	95.2	89.0	83.4	67.7	83.35

On the KITTI dataset, Fig.18 compares the detection performance of our SED-CRCNN method to that of the Cascade R-CNN algorithm before improvement. Where (a) represents the dataset's original image, (b) Cascade R-CNN detection results, and (c) results from our improved method detection.

3) Experimental results on the SODA10M dataset: This research plans experiments to migrate the dataset to the SODA10M dataset for validation in order to test whether the SED-CRCNN approach suggested in this paper has some generalization capacity. The Cascade R-CNN methodology and our enhanced method were tested on this dataset, and Fig.19 illustrates the associated AP values for all classes and the overall mAP values. It is evident that our algorithm performs better than the original algorithm in the majority of categories, and that our algorithm's mAP values are superior to those of the original algorithm.

We also contrasted the enhanced strategy with alternative approaches. Table 5 lists the total mAP values as well as the associated AP values for each category. Some example methodologies are chosen for comparison experiments in this paper to make them more comparable. These include the two-stage classical algorithm Grid-RCNN, the one-stage detection classical algorithm FCOS, the anchor-free method FSAF which also uses FPN, and the baseline model Cascade R-CNN method. Each method was tested on the SODA10M validation set after training on the SODA10M training set.

A comparison of the trial results reveals that, even though our algorithm cannot get the best results on every classification, it produces the best results overall and does well in all categories. In terms of average accuracy performance, it performs better than alternative methods. This demonstrates that our study methodology performs better.

On the SODA10M dataset, Fig.20 displays the detection results for the pre-changed Cascade R-CNN algorithm and our modified SED-CRCNN technique. Where (a) represents the original image from the SODA10M dataset, (b) represents the Cascade R-CNN detection result, and (c) represents the detection result from our enhanced technique.

For each category of target detection in this comparison, Cascade R-CNN did not produce good results, and small targets were not detected well. In contrast, as shown in portion (c) of Fig.20, when our improved method detected, it also discovered several targets that the pre-improvement algorithm was unable to identify as well as somewhat decreased the duplicated frames, leading to superior detection results.

V. Conclusion

Road object detection is a crucial step in autonomous driving environment perception and is crucial to the functioning of driverless cars. It provides the foundation for following higher-level activities like decision planning and behavioral control of the car. The ability of autonomous driving systems to identify and comprehend their surroundings more quickly and accurately can be improved by improving the accuracy of road object detection algorithms.

This paper investigates the Cascade R-CNN-based road target detection algorithm and improves and optimizes the algorithm to effectively improve its detection capability. The main work of this paper is as follows: To address these issues, this paper focuses on cascaded object detection models as the baseline for research to address the overfitting and mismatching problems in convolutional neural networks. Firstly, a study on data enhancement and generation of more accurate prediction boxes was used to pre-process the data for the problem of difficult data collection or insufficient sample size, resulting in unsatisfactory training results. The backbone network is then given an attention mechanism to alleviate the original network's low accuracy, which could lead to overfitting and high memory utilization. The attention mechanism SENet algorithm and the backbone network ResNet are coupled to create the SE-ResNet technique, which enhances the algorithm's feature extraction and classification capabilities while also enhancing network performance. To solve the problem that the original algorithm is unable to visually identify finely localized items, resulting in poor target identification and localization, deformable convolution was finally added to the method. By adaptively adapting its sampling position and size to different shapes and sizes of target objects, the network can better adapt to different target object shapes and poses, and effectively improve detection accuracy.

Finally, lots of ablation comparison experiments were conducted on KITTI and SODA10M public datasets to analyze the evaluation indexes and the overall detection effect of each category, confirm the effectiveness of each module, and prove that the algorithm has a certain generalization ability. The detection performance of our method has been established by comparison experiments with the upgraded prior algorithm and comparisons with other cutting-edge algorithms, and it has some importance to unmanned research.

References

[1] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection," in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886-893, 2005.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, pp. 91-110, 2004.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2323, 1998.

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

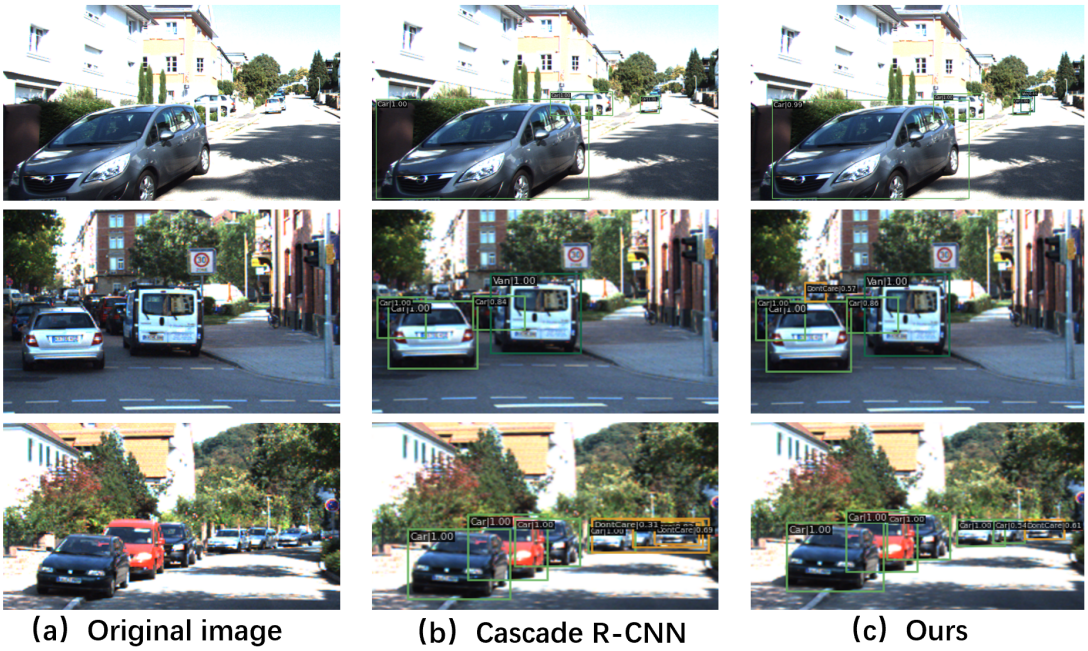

Fig. 18: Detection results for KITTI

TABLE V: Comparison with other methods on the SODA10M dataset (%)

Methods	Pedestrian	Cyclist	Car	Truck	Tram	Tricycle	mAP
Grid-RCNN	26.2	36.2	59.6	46.6	36.6	11.1	36.1
FCOS	23.5	41.0	63.5	48.1	38.1	7.9	37.0
FSAF	27.0	44.7	67.0	48.7	35.5	0.6	37.3
Cascade R-CNN	30.3	42.1	59.6	43.4	42.1	15.2	38.9
Ours	29.7	42.7	61.1	44.8	43.5	13.7	39.3

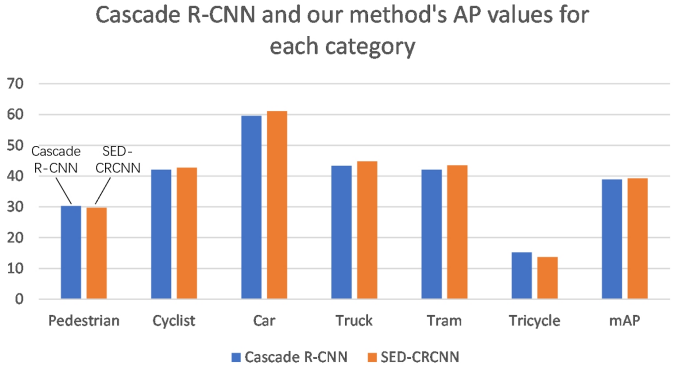

Fig. 19: Cascade R-CNN and our method's AP values for each category

[5] R. Girshick, "Fast r-cnn," in Proceedings of the IEEE International Conference on Computer Vision, pp. 1440-1448, 2015.
[6] S. Azam, A. Rafique, and M. Jeon, "Vehicle pose detection using region based convolutional neural network," in 2016 International Conference on Control, Automation and Information Sciences (ICCAIS), pp. 194-198, 2016.
[7] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, and R. Fergus, "Overfeat: integrated recognition, localization and detection using convolutional networks," arXiv: 1312.6229, 2013.
[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788, 2016.
[9] J. Redmon, A. Farhadi, "Yolo9000: better, faster, stronger," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271, 2017.
[10] J. Redmon, A. Farhadi, "Yolov3: an incremental improvement," arXiv:1804.02767, 2018.
[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg, "Ssd: single shot multibox detector," in Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, pp.21-37, 2016.
[12] D. Hubel, T. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," the Journal of Physiology, 1959.
[13] M. Marvin, A. Seymour, "Perceptrons (expanded edition)," MIT Press, Cambridge, Mass, 1988.
[14] K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," Biological Cybernetics, pp. 193-202, 1980.
[15] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," Nature, pp. 533-536, 1986.
[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffer, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, pp. 2278-2324, 1998.
[17] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in Communications of the ACM, pp. 84-90, 2017.
[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587, 2014.
[19] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling indeep convolutional networks for visual recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1904-1916, 2015.
[20] Z. Cai, N. Vasconcelos, "Cascade r-cnn: delving into high quality object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6154-6162, 2015.
[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," Advances in Neural Information Processing Systems, 2015.
[22] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.
[23] T. Lin, P. Dollár, R. Girshick, and K. He, "Feature pyramid networks for object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117-2125, 2017.
[24] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks." in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132-7141, 2018.

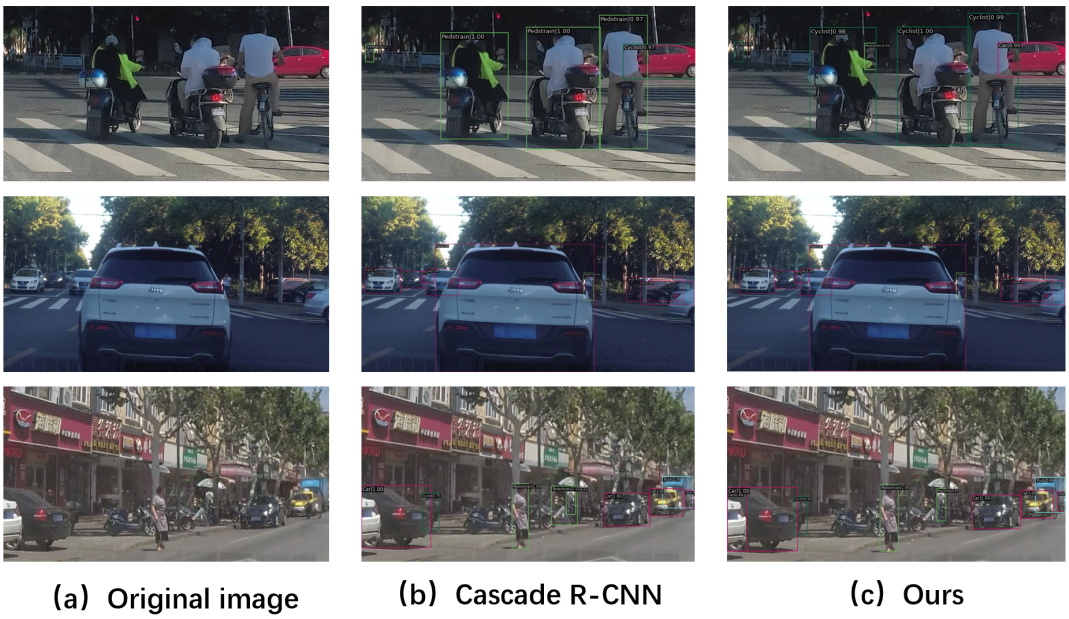

Fig. 20: Detection results for SODA10M

[25] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in Proceedings of the IEEE International Conference on Computer Vision, pp. 764-773, 2017.
[26] 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, pp. 3354-3361, 2012.
[27] J. Han, X. Liang, H. Xu, K. Cheng, L. Hong, J. Mao, and C. Xu, "SODA10M: a large-scale 2D self/Semi-supervised object detection dataset for autonomous driving," arXiv:2106.11118, 2021.