Computing and Informatics, Vol. 42, 2023, 834–860, doi: 10.31577/cai_2023_4_834

UDP-YOLO: HIGH EFFICIENCY AND REAL-TIME PERFORMANCE OF AUTONOMOUS DRIVING TECHNOLOGY

Yonghao Liu, Hongwei Ding*, Zhijun Yang, Qianxue Xu

School of Information
YunNan University, Kunming, 650500, China
e-mail: {lyh19990202, dhw1964}@163.com

Guangen Ding

Yunnan Province Highway Networking Charge Management Co.
Kunming, 650000, China

Peng Hu

Research and Development Department, Youbei Technology Co.
Kunming, 650000, China

Abstract. In recent years, autonomous driving technology has gradually appeared in our field of vision. It senses the surrounding environment by using radar, laser, ultrasound, GPS, computer vision and other technologies, and then identifies obstacles and various signboards, and plans a suitable path to control the driving of vehicles. However, some problems occur when this technology is applied in foggy environment, such as the low probability of recognizing objects, or the fact that some objects cannot be recognized because the fog's fuzzy degree makes the planned path wrong. In view of this defect, and considering that automatic driving technology needs to respond quickly to objects when driving, this paper extends the prior defogging algorithm of dark channel, and proposes UDP-YOLO network to apply it to automatic driving technology. This paper is mainly divided into two parts:

* Corresponding author

1. Image processing: firstly, the data set is discriminated whether there is fog or not, then the fogged data set is defogged by defogging algorithm, and finally, the defogged data set is subjected to adaptive brightness enhancement; 2. Target detection: UDP-YOLO network proposed in this paper is used to detect the defogged data set. Through the observation results, it is found that the performance of the model proposed in this paper has been greatly improved while balancing the speed.

Keywords: Automatic driving technology, computer vision, object detection, image processing

1 INTRODUCTION

The target detection task is to find out the objects that people are interested in images or videos, and simultaneously detect their positions and sizes. As one of the basic problems of computer vision, target detection forms the basis of many other vision tasks, such as instance segmentation [1], image annotation [2], and target tracking [3]. From the perspective of application of detection, pedestrian detection [4], face detection [5], text detection [6], traffic light detection [7], and remote sensing target detection [8] are collectively referred to as the five major applications of target detection.

At present, the target detection algorithms are conducted in two phases: Inputting an image and generating candidate region suggestions, after classifying candidate areas and correcting coordinates, and finally detecting them. This kind of algorithm is a two-stage algorithm based on generating regional suggestions. Typical representative algorithms include R-CNN [9], Fast-RCNN [10], Faster-RCNN [11], MASK-RCNN [12], etc. The other one is the single-stage algorithm, which carries out regression analysis of neural network by directly inputting pictures, and then detecting them. This kind of algorithm regards target detection as a regression problem and does not need to generate regional suggestions. Typical representative algorithms are YOLO Series [13, 14, 15, 16, 17], SSD [18], etc. Although the accuracy of the single-stage algorithm is slightly lower than that of the former, it is favored by researchers because of its powerful real-time detection speed in such an area of pursuing real-time. Despite the above two types of algorithms have good performance in their respective fields, there are some problems in cross-domain detection. In addition, under the limited mobile devices, YOLO series algorithms cannot meet the requirements of real-time detection, so YOLOv4-tiny [19], a simplified version of YOLOv4, was born, considering both detection performance and real-time detection. YOLOv4-tiny reduces the network model and parameters based on YOLOv4, and is suitable for deployment in mobile devices with limited computing power. However, although the above algorithm has good performance when applied to other data sets, there will be some problems when applied to some specific scenes. For example, when we apply the above algorithm to the direct detection

of data sets in foggy scenes, there will be either errors in the detection category or a decline in the detection performance.

In view of this defect, this paper proposes an improved dark channel prior defogging algorithm and UDP-YOLO model, which mainly performs some image processing on the data set in foggy environment first, and then detects the defogged data set by our UDP-YOLO model. When detecting objects in fog environment, it is generally implemented by the principle of defogging firstly and then detecting. The defogging of images can be divided into two types: one is defogging by traditional learning methods, such as image enhancement or image restoration. The defogging algorithms of image enhancement include histogram equalization [20], homomorphic filtering [21], wavelet transform [22] and Retinex [23]. The defogging algorithm of image restoration includes dark channel prior the defogging algorithm. The other is to defog the image by deep learning, and the representative algorithms are DehazeNet [24], AOD-Net [25] and GCANet [22]. Although the algorithm based on deep learning is better than the algorithm based on traditional learning in image defogging, it is not suitable for unmanned driving technology because it takes a long time. Therefore, among the above-mentioned algorithms, the dark channel prior defogging algorithm based on image enhancement can achieve the defogging effect on the one hand, and has good real-time performance on the other hand. In addition, considering that the automatic driving technology will also be applied to the fog-free environment, it will waste time to defog the fog-free images, so this paper adds a fog detection algorithm based on RSV calculation [26] on the basis of this algorithm, which can first judge the quality of the images and decide whether to defog them or not. After defogging by dark channel prior algorithm, the brightness of the defogged image is dark, so the image is subjected to adaptive brightness enhancement. The above-mentioned image processing process has a general effect when it is carried out alone, and the effect is quite good when it is fused.

After image processing, the original YOLOv4-tiny model is used to detect the data set in foggy environment. Because the original model only has two prediction scales of 13×13 and 26×26, and there are a lot of small objects in our data set, it is found that the effect of the model on small object detection is not very good. Therefore, after replacing and pruning the backbone network, the neck network is also improved, multi-feature fusion is completed, and a small target detection head is added. Finally, while ensuring the performance, we added a lightweight module PPM to the model to increase the receptive field and enhance its feature extraction ability. A lightweight attention module CBAM is added to improve the performance of detection tasks. Among the measures mentioned above, our contribution can be divided into five points:

1. We extend the dark channel prior algorithm based on image enhancement, and combine it with the fog detection algorithm based on RSV calculation. Firstly, we judge the quality of the image, and then choose whether to defog all the images. In addition, the defogged image is subjected to adaptive brightness enhancement.

2. The backbone network is replaced and the number of convolution cores is reduced, and a lightweight CSP-MobileNet structure is proposed.
3. We modified the neck network, proposed a new multi-feature fusion structure, and added a small target detection head to deal with the low performance of small target detection.
4. We add PPM module to the middle area of the network to increase the receptive field of the model, so as to improve the feature extraction ability.
5. An improved CBAM attention mechanism module is added to the modified multi-feature fusion partial structure to obtain important information in the feature map.

2 RELATED WORK

2.1 Image Defogging

Image defogging is mainly divided into traditional image defogging and based on deep learning defogging. Traditional defogging algorithms include image enhancement and image restoration, and image enhancement is one of the most basic contents of digital image processing technology. In practical application, no matter what kind of device is used to collect images, the visual effect of the acquired images is not ideal due to noise, illumination, weather and other reasons. For example, the images obtained in foggy days are blurred and it is difficult to extract detailed information.

Generally, the defogging algorithm based on image enhancement does not consider the reasons of image degradation in foggy scenes, but directly processes the foggy images, so as to enhance the global characteristics or local images of the foggy images, improve the image quality, enrich the information in the images and make them look clearer. This kind of algorithm includes histogram equalization, wavelet transform, Retinex algorithm, etc. The histogram equalization algorithm makes the pixel distribution of the image more uniform and enlarges the details of the image. Wavelet transform algorithm decomposes the image and enlarges the useful part. According to the imaging principle, Retinex algorithm eliminates the influence of reflected components and achieves the effect of image enhancement and defogging. On the basis of this kind of algorithm, many improved algorithms based on the principle of image enhancement have appeared [27].

The algorithms based on image restoration are defogged by atmospheric scattering model. This kind of algorithm will first analyze the reasons that degrade the original image, and then establish a physical model to defog. The most classical algorithm is the dark channel prior algorithm. By analyzing the features of a large number of fog-free images, the prior relationship between fog-free images and some parameters in the atmospheric scattering model is found. Therefore, the detailed information of the image can be kept to a great extent to achieve the purpose of defogging, and many improved algorithms based on dark channel prior are

proposed [28, 29, 30]. Deep learning-based defogging is to train the defogging model through a large number of rich image defogging data sets as data drivers, so as to estimate the transmittance map or fog-free model for defogging. CNN or GAN can also be used to defog blurred images directly. Typical representative algorithms are Dehaze-Net and AOD-Net. Although the defogging effect of these algorithms is good, it takes a long time to process data sets.

2.2 Target Detection Based on Deep Convolution Neural Network

Early feature detection models such as Viola-Jones detector [31], HOG (Histogram of Oriented Gradients) [32] and DPM (Deformable Parts Model) [33] are constructed by integrating a series of hand-designed feature extractors. These models are characterized by a slow speed, low accuracy and poor cross-domain performance. In 2012, Krizhevsky et al. proposed AlexNet [34], an image classifier based on convolutional neural network, which achieved higher performance than the best model at that time. AlexNet used a variety of convolutional kernels to obtain image features, and also used dropout and ReLU for regularization and accelerated training respectively. Let the convolutional neural network enter the public eye, and soon caused a series of research upsurge.

Detectors based on convolutional neural networks can be divided into two categories [35]: two-stage detectors and one-stage detectors. Among them, the two-stage detector has a separate module for generating the region candidate box. The first-stage detector directly separates and locates semantic objects through intensive sampling. R-CNN is the first article in a series of two-stage detectors, which proves that CNNs can greatly improve the performance. R-CNN uses a region proposals CNN module with an unknown category to transform detection into classification and location problems. He et al. proposed to use SPP [36], a pool layer of spatial pyramid, to process pictures with arbitrary size and width ratio. This network reduces the amount of computation by shifting convolution layer and adding pooling layer, so that the network does not depend on size. Because both R-CNN and SPP-Net are trained separately in multiple stages, Faster-RCNN solves this problem by creating a single end-to-end trainable system, which is 146 times faster than R-CNN model. Lin et al. considered that in the face of small target detection, image pyramids would be used for multiple levels to obtain feature pyramids, but the calculation time would be correspondingly increased. Therefore, the feature pyramid network [37] is proposed, which adopts the top-down horizontal connection structure to construct high-level semantic features on different scales. Dai et al. proposed a method combining R-FCN [38] with Faster R-CNN to solve the translation invariance problem of convolutional neural network, and realized a fast and more accurate detector. Mask-CNN is based on Faster R-CNN, which adds a branch for parallel pixel-level target instance segmentation. DetectoRS [39] combines the above-mentioned systems to improve the performance of detectors, and is equipped with the most advanced two-stage detectors. Its RFP and SAC modules are universal and can be used in other detection models. Although the proposed two-stage

detector has good performance in target detection, it is not suitable for real-time detection because of its numerous deep convolution neural network.

Considering that the speed of two-stage detectors is really slow, the first-stage detectors directly classify and locate semantic targets through intensive sampling, and they use predefined boxes with different proportions and aspect ratios to locate targets. YOLOv1 reconstructs the detection problem, regards it as a regression problem, and directly predicts image pixels as targets and their bounding box attributes. SSD is the first one-stage detector that can maintain its performance and be compatible with real-time, but its performance for small target detection is somewhat difficult. YOLOv2 replaces the backbone network on the basis of YOLOv1, and combines a variety of technologies, such as adding BN to improve convergence, and training classification and detection systems to improve the number of detection categories. The accuracy and speed have been improved. Although the first-stage detector has achieved good results in speed, its performance is a bit low. The reason is that the background class is unbalanced. So Lin et al. proposed a modified cross entropy loss in RetinaNet [40] detector to solve this problem. YOLOv3 is based on YOLOv2 and integrates various technologies, such as data enhancement, multi-scale training, batch standardization, etc. Duan et al. proposed CenterNet [41] to model the object as a point, and the input image generates heatmap through FCN, and the peak value of heatmap corresponds to the center of the detected object. EfficientDet [42] constructed the idea of an extensible detector with higher accuracy and efficiency, and introduced effective multi-scale features, BiFPN and model scaling. YOLOv4 model combines various methods to design a target detector that can work quickly and easily in the existing system. Using the bag-of-freebies method, it only increases the training time without affecting the reasoning time.

2.3 Feature Fusion

Feature fusion is an important method in the field of pattern recognition. In many jobs, fusing features of different scales is an important means to improve detection performance. The low-level features have higher resolution and contain more location information and detail information, because of less convolution, the semantics are lower and the noise goes up. High-level features semantic information is stronger, but the resolution is low and the perception of details is poor. By fusing the features of high and low levels, we can make use of various image features, realize the complementary advantages of multiple features, and obtain more robust and accurate recognition results.

At present, the most common feature fusion methods are FPN, PANet [43] and NAS-FPN [44]. Among them, FPN uses semantic information of low-level and high-level features at the same time, and fuses features of different levels to achieve the prediction effect. Moreover, the prediction is performed separately on each fused feature layer, which is different from the conventional feature fusion method. PANet is a feature fusion structure proposed in YOLOv4, which adds a layer of FPN network from bottom to top, extracts features from each feature layer

for each proposal, and finally obtains the features to be detected by convolution-upsampling and full connection layer fusion. NAS-FPN mainly reorganizes feature maps with multiple scales, and then performs merging cell operation on them. This operation is divided into three steps: First, two candidate feature layers are selected as input feature layers, then select the resolution of the output feature, and finally, a binary operation is selected to integrate the two input feature layers into a new output feature layer. After completing this series of operations, the cyclic operation continues to obtain the final output feature layer for detection.

2.4 Attention Mechanism

With the development of science and technology, more and more information comes to us, and there is a large amount of information around us all the time. However, the information we receive in a limited time is limited, but researchers have found that the human visual system has a strong visual information processing capability in a limited field of vision. When we process information in the early stage, we will focus our attention on the important things. This choice allows us to reduce the amount of information to be processed, so that we can suppress unimportant stimuli when processing complex visual information, and provide easier and more relevant new information for higher-level perceptual reasoning and more complex visual processing tasks (such as target recognition, target classification, video comprehension, etc.). In view of this advantage, researchers put forward the idea of attention mechanism. The main idea of attention mechanism is to get the difference of the importance of each feature map through some measures, so as to use more resources for more important tasks, and use the results of tasks to guide the weight update of feature maps in reverse, thus completing the corresponding tasks efficiently and quickly.

At present, the attention mechanisms proposed are mainly divided into two types: single-channel attention and multi-channel attention. There is only one module in single-channel attention to obtain attention in the channel, and the representative networks mainly include SE-Net [45] and ECA-Net [46]. The main idea of SE-Net is to estimate the loss function value LOSS through the network model, so as to learn the feature weight. Generally speaking, the weight of the feature graph with obvious task effect becomes larger, while the weight of the feature graph with no obvious or no effect becomes smaller, and then the model is trained to achieve better results. ECA-Net is an improvement on SE-Net, and proposes a local cross-channel interaction strategy without dimensionality reduction and a method of adaptively selecting the size of one-dimensional convolution kernel. More accurate attention information is obtained by summarizing cross-channel information in one-dimensional convolution layer. There are two modules in the multi-channel attention mechanism, which mainly capture the attention between channels and feature pixels. The representative networks mainly include SK-Net [47], CBAM [48] and DA-Net [49]. SK-Net is an attention mechanism based on convolution kernel, that is, by comparing the importance of different images passing through different

convolution kernels. CBAM module puts forward that the channel of feature image not only contains a lot of attention information, but also contains rich attention information inside the channel, that is, between the pixels of feature image. Therefore, CBAM has built two modules, CAM and BAM, to collect the attention in the channel and empty space respectively, and then synthesize the collected attention to avoid wasting the attention in the space. Although the idea of DA-Net (Dual Attention Network) network is the same as that of CBAM, its way of obtaining two channels of attention information is different from that of CBAM, and it is obtained through parallel mode.

2.5 Model Pruning

In order to improve the performance of the model while maintaining the speed of the model, some pruning measures are taken to the model. Pruning method can explore the redundancy of model weights and try to trim redundant and non-critical weights [50, 51]. Model pruning is mainly divided into unstructured pruning [52] and structured pruning [53]. Unstructured pruning mainly changes the combined structure of neurons in the single layer of the network model, and its representative pruning includes fine-grained pruning [54], vector pruning [55] and nuclear pruning [56]. Although this kind of pruning can achieve a high compression rate, while maintaining a high performance. But it needs enough hardware structure to support sparse operations. Structured pruning is to change the structural characteristics of the network model, so as to achieve the effect of compressing the model. Its representative pruning includes filter pruning [57]. Although there are many types of model pruning, the main purpose is to prune the neural network structure, and the general ideas can be summarized into three types: standard pruning, pruning based on sub-model sampling, and pruning based on search. The idea of standard pruning is to carry out pre-training, then pruning, then fine-tuning, and then repeat the above processes in turn, and finally get a suitable pruning structure. Sub-model-based sampling process is to randomly sample the trained model according to the pruning target, and then prune each sampled network structure to obtain the sampling model and evaluate the best pruning model. Search-based pruning is based on unsupervised learning or semi-supervised learning algorithm, and the optimal substructures are searched by selecting pruning target.

3 MODEL DESIGN

3.1 YOLOv4-Tiny

Figure 1 shows the model structure of YOLOv4-tiny, which is simplified based on YOLOv4 model. The model consists of three parts: Backbone network to extract features, feature pyramid network to fuse features, and YOLO head to predict the acquired features.

The backbone network of this model is CSPDarknet53-tiny network, which is mainly composed of Conv and CSPBlock. Conv not only performs convolution operation on it, but also performs batch standardization and activation function operation, in which BN (Batch Norm) is used in batch standardization, and the activation function is modified to Leaky ReLU (Leaky Rectified Linear Unit). In the CSPBlock module, CSP-net is mainly used. In fact, the original residual block stack is split into two parts: the main part continues to stack the original residual block, and the other part is like a residual edge. After a small amount of processing, it is directly connected to the last two parts, and the final output is obtained by merging them. The FPN structure can fuse feature maps of different scales, which can not only ensure the rich semantic information of the deep network, but also obtain the geometric details of the low-level network, so as to strengthen the ability of feature extraction. YOLO head predicted the features of the fused feature map, and finally formed two prediction scales of 13×13 and 26×26. Although YOLOv4-tiny model has good detection performance and real-time performance on VOC and COCO data sets, its effect is not so optimistic if the model is applied to the target detection of autonomous driving technology in foggy scenes. Therefore, this paper proposes UDP-YOLO network for this defect and combines it with the defogging algorithm. Firstly, the improved defogging algorithm is used to process the image of the data set. Then, CSP-MobileNet structure is introduced into the backbone network, and a new idea of multi-feature fusion is proposed. The receptive field module is added to the extracted feature maps with dimensions of 128×128, 256×256, and 512×512. Finally, attention mechanism is added to some stages of feature fusion.

3.2 Image Processing

3.2.1 Judge Whether There is a Fog or Not

Considering that the defogging of data sets is a time-consuming process, and when the data sets contain normal pictures, defogging will make the high-frequency components contained in them change greatly, and the detection effect will be poor. Therefore, we need to improve the real-time performance and effectiveness of our improved model in foggy scenes. Firstly, the fog detection algorithm is used to calculate the ratio of saturation (S) and color value (V) of the picture, and then the defogging process is judged, as shown in Figure 2. Firstly, find out the general driving direction of vehicles, and find out their intersection point by extending the driving direction, which is the vanishing point proposed by us. Then, take this point as the center to select an area, which is called ROI box, and then calculate the ratio of saturation (S) and color value (V) in HSV color model domain in this box. In addition, considering that setting one ROI box may lead to inaccurate results, four ROI boxes are set, and the ratio of each ROI box is calculated separately and the average value is added to make a more reasonable fog judgment, which is defined as

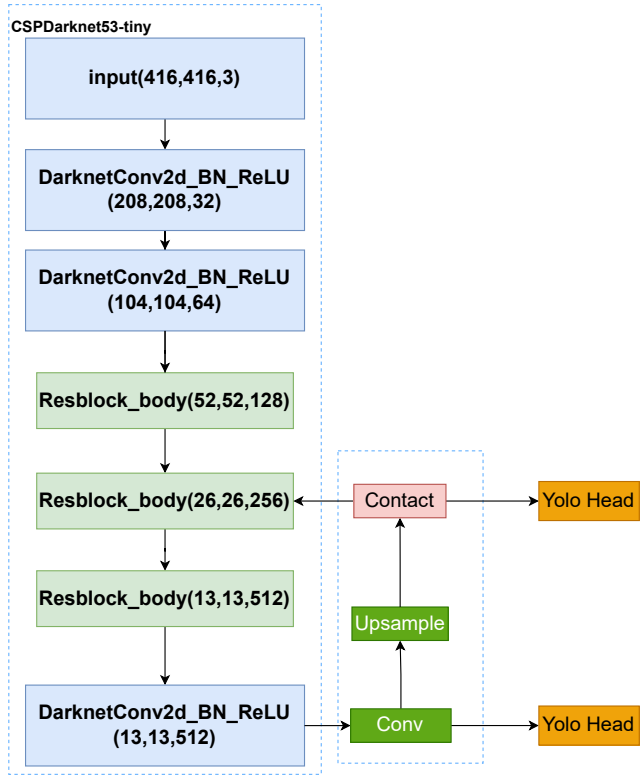


Figure 1. YOLOv4-tiny model structure

Equation (1) and Equation (2):

$$u(i) = \frac{s(i)}{v(i)}, \tag{1}$$

$$u = \frac{\sum_{i=1}^{4} u(i)}{4}, \tag{2}$$

where i is the selected i^{th} ROI box, $s(i)$ is the saturation of the i^{th} box, and $v(i)$ is the color value of the i^{th} box. $u(i)$ represents the ratio of the saturation of the i^{th} frame to the color value of the i^{th} frame, and u represents the average value of $u(i)$ of the four frames obtained respectively. It is compared by comparing the value with the set prior value $u0 = 3.5$. If the value is greater than 3.5, defogging is required, otherwise, it is not required.

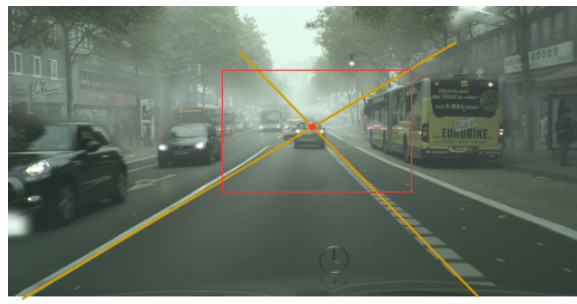


Figure 2. Judgment principle diagram of fog

3.2.2 Improve the Ability of Fog Environment Detection

After judging whether the image is foggy or not, the foggy image is restored, which uses the atmospheric scattering model to defog the image. The atmospheric scattering model is shown in Figure 3.

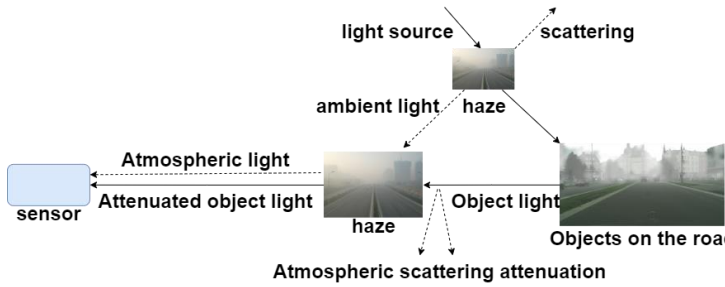


Figure 3. Atmospheric scattering model

By modeling the imaging process of fog scene, we can get:

$$F(x) = J(x)t(x) + A(1 - t(x)). \tag{3}$$

Here, x represents a certain pixel in $F(x)$, $F(x)$ represents a foggy image scattered by atmospheric particles and reaching the sensor, $J(x)$ is a clear fog-free image that is not scattered by atmospheric particles, A is atmospheric illumination, and $t(x)$ represents the transmittance in a foggy scene. In the above formula: $t(x) = e^{-\beta d(x)}$ where β represents the atmospheric scattering coefficient and $d(x)$ represents the scene depth.

Based on the atmospheric scattering model, priority of an image defogging algorithm based on dark channel is proposed. The theoretical basis of this algorithm is that when the image is taken in the normal weather, at least one channel in RGB image will have pixel values with intensity approaching 0 after the sky area of the

image is removed. Therefore, take any shot image and define it as $J(x)$, it can also be written as:

$$J^{dark}(x) = \min_{t\in\Omega(x)}(\min, J^c(t)) \quad (4)$$

In Equation (4), J^c is the image of a certain color channel in RGB color space in $J(x)$. $\Omega(x)$ is a neighborhood centered on pixel X in image $J(x)$. The image defogging algorithm based on dark channel prior includes the following steps:

1. Estimate and refine the transmittance value. The idea is as follows: Assuming that the atmospheric care A is known, and $A>0$, divide both the left and right sides by A^c to get Equation (5):

$$\frac{I^c(x)}{A^c} = t(x)\frac{J^c(x)}{A^c} + 1 - t(x), c \in (r, g, b). \quad (5)$$

   By minimizing the above formula, you can get:

$$\min_{y\in\Omega(x)}(\min_{c\in(r,g,b)}\frac{I^c(y)}{A^c}) = \tilde{t}(x)\min_{y\in\Omega(x)}(\min_{c\in(r,g,b,)}\frac{J^c(y)}{A^c}) + 1 - \tilde{t}(x). \quad (6)$$

   In Equation (6), it is constant only in a small neighborhood, so it is not needed to minimize it. And the dark channel in the clear image is 0. The predicted transmittance can be corrected by introducing a factor w between 0 and 1 (typically 0.95) to make the defogged image more natural.
2. Estimate the atmospheric illumination value.
3. Substituting the transmittance value $t(x)$ and atmospheric illumination value A obtained in Equation (1) into Equation (2), and obtaining the defogged image of the input image.
4. The image that needs to be defogged is restored by a dark channel prior algorithm based on guided filtering. After the image is restored, considering that the color quality of the image is a little black, the final image is obtained by brightness enhancement.

3.3 UDP-YOLO Network Structure

3.3.1 Improved Multi-Scale Prediction Network

For the application of autonomous driving technology in foggy scenes, there are usually small objects such as people and bicycles in the scenes, while YOLOv4-tiny model has only two prediction scales of 13×13 and 26×26. Using YOLOv4-tiny model to detect the data set we selected in this paper, it is found that the detection effect of small objects such as people and bicycles is poor. Therefore, inspired by FPN, PANet and NAS-FPN, this paper proposes a new neck network for feature fusion without adding too many model parameters, and adds a small target detection head. The improved network structure is shown in Figure 4.

As can be seen from Figure 4, this paper has improved the original MobileNetv1 network, and made the following modifications to the original stage1, stage2, and stage3, reducing the number of convolution kernels of stage1 and making it output a feature map with a size of 64×64. Then, CSP module was added between stage1 and stage2 to make it input a feature map with a size of 128×128 and extract it. Stage2 is also modified to output and extract the feature map with the size of 256×256. Finally, the number of convolution kernels of stage3 is trimmed to output and extract the feature map with the size of 512×512. The characteristic graphs of these three dimensions are respectively marked as F1, F2 and F3. Next, feature fusion is performed. Firstly, F3 is convolved and downsampled to get F3.1, and F2 is convolved to get F2.1. F1 is convolved and upsampled to get F1.1, and then it is fused to get our first fused feature. After that, F1.1 is up-sampled, and fused with the feature map of F3 after convolution operation to obtain the second feature. At last, F2.1 is downsampled to get F2.2, and it is fused with the feature map obtained by convolution operation of F1 to get the third feature. These three fusion features not only contain strong detail information, but also have great semantic information, so the detected results are very comprehensive.

3.3.2 Expand Receptive Field

Expanding receptive field in the model, a low-cost measure, is helpful to improve the feature extraction ability, thus improving the performance. In this paper, we put the PPM module into the feature maps of P3, P4 and P5 extracted from the improved YOLOv4-tiny model in Figure 4, so as to increase the feature extraction ability. The PPM module can divide the extracted feature maps into two branches, one of which is divided into multiple sub-areas for GAP (Global Average Pooling) operation, then adjust the channel size through convolution operation, and then obtain the unpooled feature map through bilinear interpolation. The PPM module [58] consists of five steps:

1. Pool the feature map extracted from the backbone network to obtain a feature pyramid.
2. Get the characteristic maps with the size of 1×1, 2×2, 3×3, 6×6 and channel $= 1/N$ through the 1×1 depth convolution descending channel.
3. Bilinear interpolation filling and upsampling the feature map to the original feature map size.
4. Channel splicing with the feature map to obtain a feature map with double channel number.
5. Using 1×1 convolution kernel to deeply convolve and channel down the spliced feature map to obtain the final prediction result which is consistent with the channel number of the input feature map.

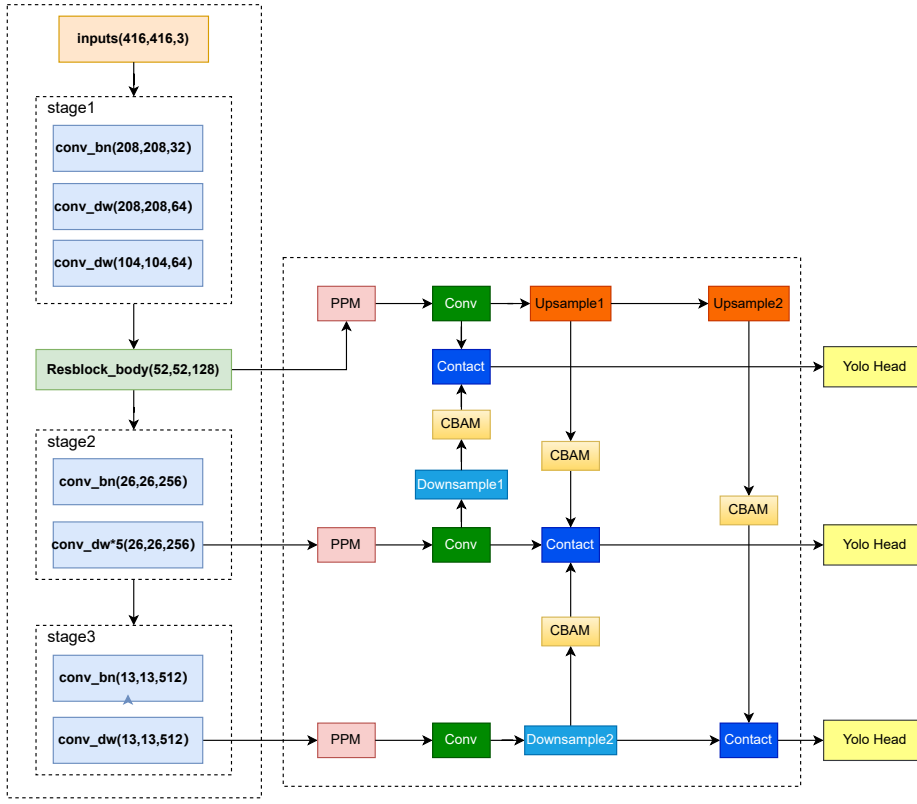


Figure 4. UDP-YOLO model structure

3.3.3 Attention Mechanism

Deep Convolutional Neural Network (CNN) has been widely used in computer field, and has made great progress in image recognition, object detection and semantic segmentation. Because the performance of the original model will be slightly degraded by pruning, this paper adds a lightweight attention module CBAM module while considering the speed and performance. This module can conduct attention mechanism in space and channel, deduce the attention weight coefficient along the channel and space dimensions, and then multiply it with feature map to adjust the features adaptively. Because CBAM is a lightweight general-purpose module, it can be seamlessly integrated into any CNN architecture, and its computational cost is basically negligible. And can carry out end-to-end training with basic CNN. On different classification and detection data sets, after integrating CBAM into different models, the performance of the models has been consistently improved, showing its wide applicability.

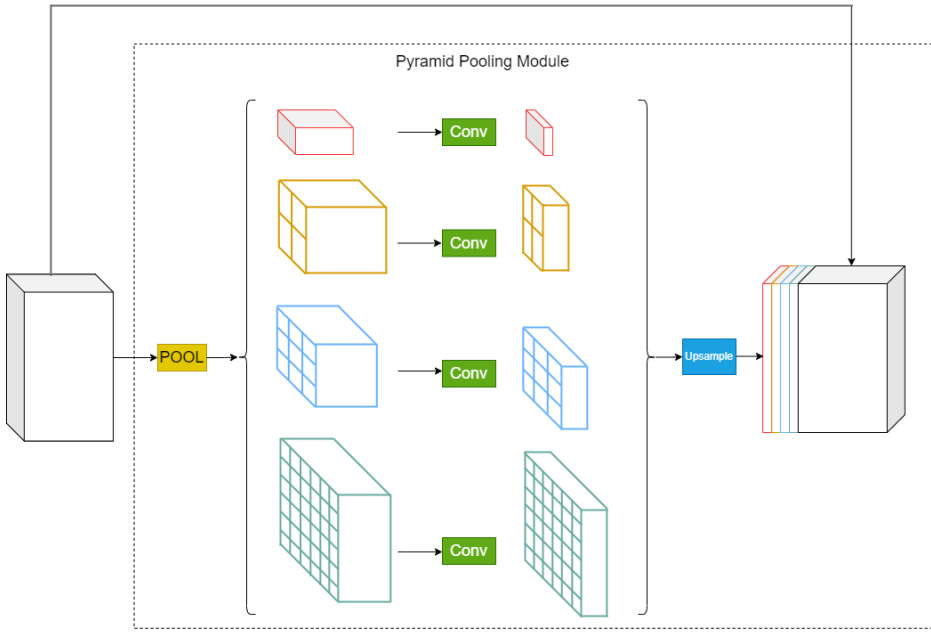


Figure 5. The module of PPM

As shown in Figure 6, CBAM module is divided into two sections: channel attention module and spatial attention module. This module can not only save parameters and attention, but also ensure that it can be integrated into the existing network architecture as a plug-and-play module.

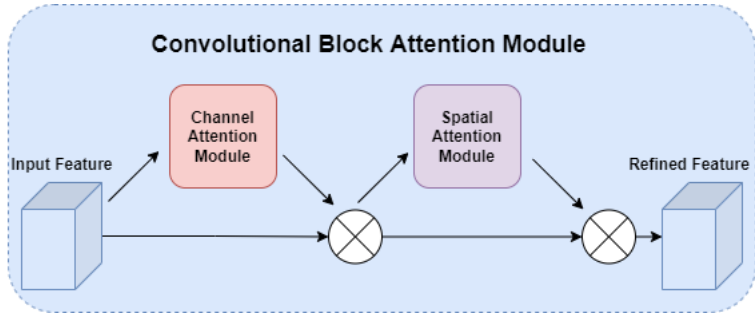


Figure 6. The module of CBAM

The CAM module is shown in Figure 7, the attention module on the channel firstly passes the input feature map F through global max pooling and global average pooling based on width and height respectively to obtain two $1 \times 1 \times C$ feature

maps C1 and C2. C1 will be upsampled. Then C1 and C2 respectively pass through a neural network sharing two layers, the number of neurons in the first layer is C/r (r is the reduction rate), and the number of neurons in the second layer is C. The features of MLP output are added based on element-wise, and then the final channel attention feature, namely M_C, which is generated by LA (LeakyRelu Activation) operation. Finally, M_C and input F are multiplied by element-wise, and the required input features of the next module are obtained.

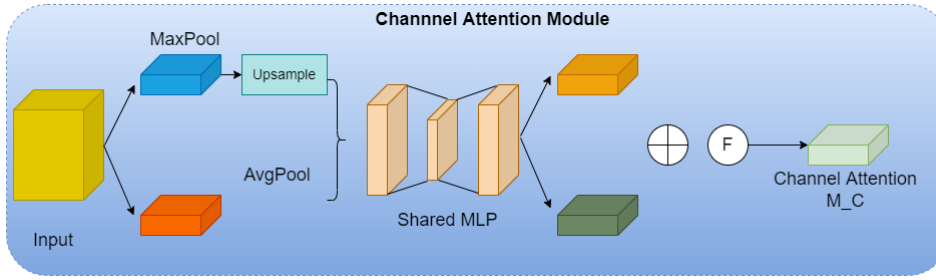


Figure 7. The module of CAM

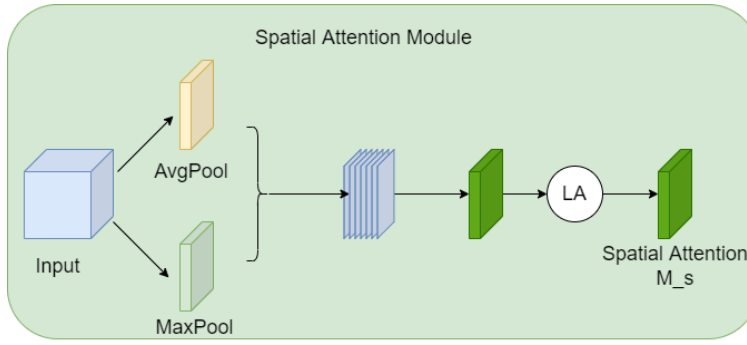


Figure 8. The module of SAM

As shown in Figure 8, SAM module uses the features obtained in the last round as a global max pooling and a global average pooling based on channel to obtain two $H \times W \times 1$ feature maps, then performs channel splicing operation on these two feature maps based on channel, and then reduces the dimension to one channel through a 7×7 convolution operation. Then the Leaky ReLU generates the spatial attention feature, namely M_s. Finally, the feature and the original input features of the module are multiplied to obtain the final features for detection.

4 EXPERIMENT

4.1 Experimental Data and Experimental Platform

The data sets adopted in this paper are Foggy-cityscape data set and BDD100k data set. Foggy-cityscape data set is a data set obtained by photographing the road conditions of many foreign cities, which can be used for object detection and segmentation. BDD100K data set is the largest and most diverse open driving data set published by Berkeley Artificial Intelligence Laboratory. In this work, a part of BDD100K data set and Foggy-cityscape data set are selected for fusion experiment, and the fused data set is divided into training set for experiment according to the proportion, and the test set is used as evaluation. In this paper, mAP and FPS are used to evaluate the performance of the model. Table 1 describes the details of the data set we selected.

	Foggy-cityscape and BDD100k
Number of classes	6
Training datasets	5 976
Test datasets	960

Table 1. Datasets details

The operating system used in this experimental platform is Windows 10, the processor is Intel (R) Core (TM) i7-4790 KCPU @ 4.00 GHz, the running memory is 32 GB, the GPU is NVIDIA GeForce GTX 3060, and the parallel computing framework version is CUDA 11.6.

The flow chart of this experiment is shown in Figure 9. Firstly, select the desired dataset from the Foggy-cityscape dataset and the BDD100k dataset, then discriminate the selected data set by fog judgment algorithm, defog the foggy data set, and finally fuse the defogged data set and the fog-free data set to divide the training set and the test set, and carry out adaptive brightness enhancement. Finally, the original YOLOv4-tiny model and our proposed UDP-YOLO model are used for detection. Figure 10 shows the effect diagram after each step of operation.

4.2 Training Model

After the image processing of the data set pair, we start to train the processed data set. During the training, we adopt the default size of YOLOv4-tiny (416, 416), and set the input batch size to 8 and the momentum to 0.9. Firstly, without using the pre-training weight, only the backbone network is loaded for training to get a better weight. Then, this weight is put into the model as the pre-training weight to train 50 epochs, and then 250 epochs are trained to get the performance of our original YOLOv4-tiny model test data set. During the pre-training, the learning rate of our first 50 training sessions was set at 0.001, and then the learning rate was gradually reduced from 0.001 to 0.0001 by adopting cosine annealing. Then we

add our improved structure in turn, and take the weight of the best performance measured last time as the pre-training weight for training.

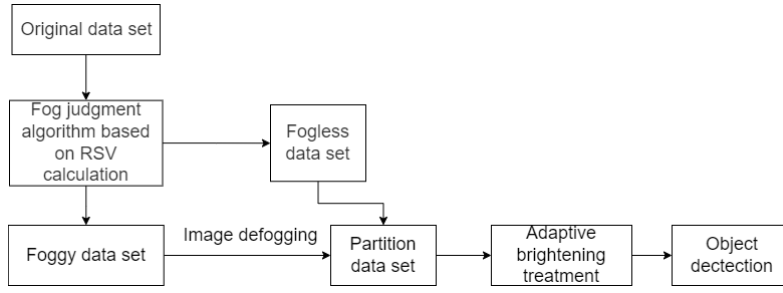


Figure 9. Experimental flow chart

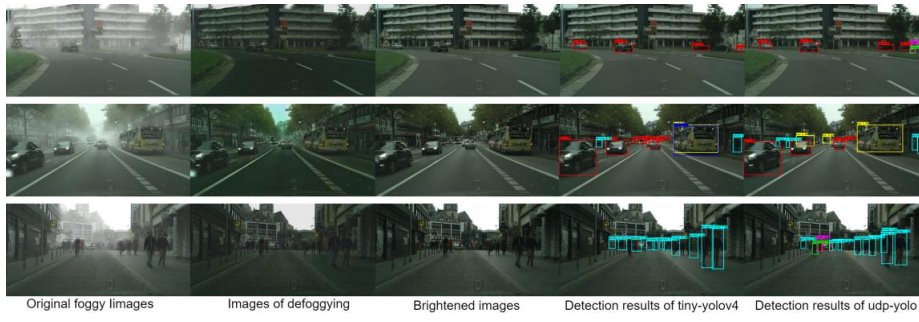


Figure 10. Image processing and target detection process

4.3 Ablation Experiment

To prove the effectiveness of our improved model, we first processed the original data through image processing, and detected the processed data by using YOLOv4-tiny model. By replacing the backbone network with MobileNetv1 and reducing the network parameters, we improved the neck network to form a new multi-feature fusion structure. Add PPM module and CBAM module in turn. The validity of the improved model is verified by mAP, FPS.

As shown in Table 2, after replacing the backbone network from CSPDarknet-tiny with MobileNetv1, and greatly modifying the neck network to form a new feature fusion structure, our average performance of detecting six types of objects has increased from 19.75 % to 32.41 %, but its speed has also decreased from the

	YOLOv4-tiny	UDP-YOLO				
MobileNetv1	–	+	+	+	+	+
Delete parameter	–	–	+	+	+	+
CSP–MobileNetv1	–	–	–	+	+	+
Add PPM	–	–	–	–	+	+
Add CBAM	–	–	–	–	–	+
mAP (%)	19.36	32.41	31.74	34.27	36.86	40.54
FPS	100.3	78.3	90.2	83.1	68.5	61.7

Table 2. The result of Ablation experiment. The '–' sign indicates that the operation corresponding to the first column of the table was not performed in the ablation experiment, and the '+' sign indicates that the operation corresponding to the first column of the table was performed in the ablation experiment.

original 100.3 to 78.3. Therefore, in order to maintain its speed, the convolution kernel of the backbone network is pruned. After pruning, the observation results show that while the performance decreases by less than one point, our speed increases by about 15 %. After that, we added CSP module to the backbone network to form CSP-MobileNet, and the detection performance increased from 31.74 % to 34.27 % when the speed decreased by less than 8 %. After that, by adding PPM module to increase the receptive field of our model, the performance is improved to 36.86 % and the speed is reduced to 68.5. Finally, we added a lightweight attention module CBAM to the model, which improved our performance to 40.54 % and the speed to 61.7.

4.4 Comparison of UDP-YOLO and YOLO Series Models

In this section, we compare the performance of the non-lightweight model of YOLO series with that of UDP-YOLO model proposed in this paper, because the performance of YOLOv4-tiny model in detecting the data set selected in this paper is low. YOLOv3 is the third version of YOLO series. The test result of this model on the selected data set is 41.35 %, but its FPS is only 48.6. Compared with UDP-YOLO proposed in this paper, its performance is 0.81 % higher, but its speed is much lower. After that, this paper replaces the backbone network of YOLOv3, Darknet53, with EfficientNet, and finds that its performance and speed are not as good as YOLOv3. YOLOv4 is based on YOLOv3, and the measured performance of this model is 42.85 %, which is 3.6 % higher than YOLOv3 and 5.6 % higher than UDP-YOLO. But the speed is reduced by 31.7 % compared with UDP-YOLO. MobileNetv2-YOLOv4 replaces the CSP-Darknet53 backbone network of YOLOv4 with MobileNetv2 for training, and evaluates the best weight after training. Although compared with YOLOV4 in speed, it is still not as fast as UDP-YOLO proposed in this paper. And its performance has been greatly reduced.

From Figure 3, we can see that. Compared with the model of YOLO series, the effect of replacing the backbone network of YOLO series and detecting it is not

	YOLOv3	EfficientNet-YOLOv3	YOLOv4	MobileNetv2-YOLOv4	UDP-YOLO
mAP (%)	41.35	27.68	42.85	28.05	40.54
FPS	48.6	46.5	42.3	52.1	61.7

Table 3. Comparison of UDP-YOLO and YOLO series models

as good as the original effect. However, the UDP-YOLO proposed in this paper, although the performance of the backbone network is a little reduced after it is replaced by a new network, is indeed much faster. In fact, we can completely increase the performance by reducing the speed, but this principle is not adopted due to the real-time requirement of autonomous driving technology. The comparison of mAP and FPS shows that the UDP-YOLO model proposed by us is completely feasible.

4.5 Comparison of UDP-YOLO and Other Lightweight Models

In this section, we use MobileNetv2-SSD, YOLOv5s, YOLOx-tiny and EfficientNet to test our selected data set, and compare the performance with our UDP-YOLO model. The experimental environment and details are the same as before.

	MobileNetv2-SSD	YOLOv5s	YOLOx-tiny	EfficientNet	UDP-YOLO
mAP (%)	22.91	38.50	37.56	34.33	40.54
FPS	75.5	25.3	48.5	30.1	61.7

Table 4. Comparison of UDP-YOLO and other lightweight models

As can be seen from Table 4, the UDP-YOLO model is at the optimal performance compared to all four of these networks. Compared to the MobileNetv2-SSD model, its performance almost doubles, although its speed is reduced by about 22 %, which is a good indication of the efficiency of our model.

4.6 Comparison Experiments Using Our Model on Road Defect Dataset

In autonomous driving technology, in addition to avoiding vehicles travelling in the road, self-driving cars also need to avoid some road defects by predicting them in advance. To demonstrate the efficient generalisation of the model proposed in this paper, experiments are conducted on the GRDDC2020 road defect detection dataset using the UDP-YOLO model. Table 5 shows the experiments comparing our proposed model with some other lightweight models.

By looking at Table 5, we see that the performance of our proposed model is better than other models, and we can show after these experiments that our proposed model is effective for the application of autonomous driving technology in this direction of computer vision.

	YOLOv4-tiny	YOLOv4	YOLOv5s	Tiny-YOLOX	UDP-YOLO
mAP (%)	52.45	54.56	56.79	56.95	57.30
FPS	101.5	30.6	24.8	45.3	62.8

Table 5. Comparison experiments using our model on road defect dataset

5 CONCLUSIONS

In this work, we propose an improved model based on YOLOv4-tiny to deal with the problem of avoiding vehicles while driving, and to demonstrate the effectiveness of our proposed model, we also carry out generalisation experiments in case of road defects. The evaluation metrics of our experimental results show that our model outperforms these comparative lightweight models, allowing us to detect the target briskly. However, there is still room for improvement in our proposed model. For example, the performance of our detection is not robust enough, which may prevent us from avoiding a vehicle in an accident while driving because we do not detect the target. Therefore, we will continue working on this issue and further to improve and experiment with the model to come up with a more efficient model that detects the target in the shortest possible time.

Acknowledgements

This research was supported by the National Natural Science Foundation of China under the project "Research on Theory and Control Protocols of Converged Multiple Access Communication Networks" (No. 61461053), as well as the National Innovation and Entrepreneurship Program for College Students (No. 202210673062) and the expert workstation of Ding Hongwei.

Author Contributions

- Yonghao Liu: Proposal of the method, experimentation, writing;
- Hongwei Ding: Supervision, revision of the original;
- Zhijun Yang: Supervision, validation;
- Guangen Ding: Data collection;
- Peng Hu: Investment, supervision;
- Qianxue Xu: Data collection.

REFERENCES

[1] Lian, J.—Yang, Z.—Liu, J.—Sun, W.—Zheng, L.—Du, X.—Yi, Z.—Shi, B.—Ma, Y.: An Overview of Image Segmentation Based on Pulse-Coupled

Neural Network. Archives of Computational Methods in Engineering, Vol. 28, 2021, No. 2, pp. 387–403, doi: 10.1007/s11831-019-09381-5.

[2] Li, J.—Zhang, C.—Zhou, J. T.—Fu, H.—Xia, S.—Hu, Q.: Deep-LIFT: Deep Label-Specific Feature Learning for Image Annotation. IEEE Transactions on Cybernetics, Vol. 52, 2021, No. 8, pp. 7732–7741, doi: 10.1109/TCYB.2021.3049630.

[3] Liu, S.—Liu, D.—Srivastava, G.—Połap, D.—Woźniak, M.: Overview and Methods of Correlation Filter Algorithms in Object Tracking. Complex & Intelligent Systems, Vol. 7, 2021, No. 4, pp. 1895–1917, doi: 10.1007/s40747-020-00161-4.

[4] Brunetti, A.—Buongiorno, D.—Trotta, G. F.—Bevilacqua, V.: Computer Vision and Deep Learning Techniques for Pedestrian Detection and Tracking: A Survey. Neurocomputing, Vol. 300, 2018, pp. 17–33, doi: 10.1016/j.neucom.2018.01.092.

[5] Minaee, S.—Luo, P.—Lin, Z.—Bowyer, K.: Going Deeper into Face Detection: A Survey. CoRR, 2021, doi: 10.48550/arXiv.2103.14983.

[6] Zhu, Y.—Du, J.: TextMountain: Accurate Scene Text Detection via Instance Segmentation. Pattern Recognition, Vol. 110, 2021, Art. No. 107336, doi: 10.1016/j.patcog.2020.107336.

[7] Liu, R. W.—Guo, Y.—Lu, Y.—Chui, K. T.—Gupta, B. B.: Deep Network-Enabled Haze Visibility Enhancement for Visual IoT-Driven Intelligent Transportation Systems. IEEE Transactions on Industrial Informatics, Vol. 19, 2023, No. 2, pp. 1581–1591, doi: 10.1109/TII.2022.3170594.

[8] Cheng, G.—Si, Y.—Hong, H.—Yao, X.—Guo, L.: Cross-Scale Feature Fusion for Object Detection in Optical Remote Sensing Images. IEEE Geoscience and Remote Sensing Letters, Vol. 18, 2021, No. 3, pp. 431–435, doi: 10.1109/LGRS.2020.2975541.

[9] Girshick, R.—Donahue, J.—Darrell, T.—Malik, J.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[10] Girshick, R.: Fast R-CNN. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[11] Ren, S.—He, K.—Girshick, R.—Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, 2017, No. 6, pp. 1137–1149, doi: 10.1109/TPAMI.2016.2577031.

[12] He, K.—Gkioxari, G.—Dollár, P.—Girshick, R.: Mask R-CNN. Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.

[13] Redmon, J.—Divvala, S.—Girshick, R.—Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[14] Redmon, J.—Farhadi, A.: YOLO9000: Better, Faster, Stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7263–7271, doi: 10.1109/CVPR.2017.690.

[15] Redmon, J.—Farhadi, A.: YOLOv3: An Incremental Improvement. CoRR, 2018, doi: 10.48550/arxiv.1804.02767.

[16] Bochkovskiy, A.—Wang, C. Y.—Liao, H. Y. M.: YOLOv4: Optimal Speed and Accuracy of Object Detection. CoRR, 2020, doi: 10.48550/arXiv.2004.10934.

[17] Wang, C. Y.—Bochkovskiy, A.—Liao, H. Y. M.: Scaled-YOLOv4: Scaling Cross Stage Partial Network. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 13029–13038, doi: 10.1109/CVPR46437.2021.01283.

[18] Liu, W.—Anguelov, D.—Erhan, D.—Szegedy, C.—Reed, S.—Fu, C. Y.—Berg, A. C.: SSD: Single Shot Multibox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.): Computer Vision – ECCV 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9905, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[19] Jiang, Z.—Zhao, L.—Li, S.—Jia, Y.: Real-Time Object Detection Method Based on Improved YOLOv4-Tiny. CoRR, 2020, doi: 10.48550/arXiv.2011.04244.

[20] Chen, D.—He, M.—Fan, Q.—Liao, J.—Zhang, L.—Hou, D.—Yuan, L.—Hua, G.: Gated Context Aggregation Network for Image Dehazing and Deraining. 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019, pp. 1375–1383, doi: 10.1109/WACV.2019.00151.

[21] Wu, H.—Tan, Z.: An Image Dehazing Algorithm Based on Single-Scale Retinex and Homomorphic Filtering. In: Liang, Q., Wang, W., Liu, X., Na, Z., Jia, M., Zhang, B. (Eds.): Communications, Signal Processing, and Systems (CSPS 2019). Springer, Singapore, Lecture Notes in Electrical Engineering, Vol. 571, 2020, pp. 1482–1493, doi: 10.1007/978-981-13-9409-6_178.

[22] Wang, L. J.—Zhu, R.: Image Defogging Algorithm of Single Color Image Based on Wavelet Transform and Histogram Equalization. Applied Mathematical Sciences, Vol. 7, 2013, No. 79, pp. 3913–3921, doi: 10.12988/ams.2013.34206.

[23] Fan, T.—Li, C.—Ma, X.—Chen, Z.—Zhang, X.—Chen, L.: An Improved Single Image Defogging Method Based on Retinex. 2017 2nd International Conference on Image, Vision and Computing (ICIVC), IEEE, 2017, pp. 410–413, doi: 10.1109/ICIVC.2017.7984588.

[24] Cai, B.—Xu, X.—Jia, K.—Qing, C.—Tao, D.: DehazeNet: An End-to-End System for Single Image Haze Removal. IEEE Transactions on Image Processing, Vol. 25, 2016, No. 11, pp. 5187–5198, doi: 10.1109/TIP.2016.2598681.

[25] Li, B.—Peng, X.—Wang, Z.—Xu, J.—Feng, D.: AOD-Net: All-in-One Dehazing Network. Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4770–4778, doi: 10.1109/ICCV.2017.511.

[26] Jeong, K.—Choi, K.—Kim, D.—Song, B. C.: Fast Fog Detection for De-Fogging of Road Driving Images. IEICE Transactions on Information and Systems, Vol. E101.D, 2018, No. 2, pp. 473–480, doi: 10.1587/transinf.2017EDP7211.

[27] Liang, Y. T.—Li, L.—Zhao, K. B.—Hu, J. H.: Defogging Algorithm of Color Images Based on Gaussian Function Weighted Histogram Specification. 2016 10th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), IEEE, 2016, pp. 364–369, doi: 10.1109/SKIMA.2016.7916248.

[28] Xu, H.—Guo, J.—Liu, Q.—Ye, L.: Fast Image Dehazing Using Improved Dark

Channel Prior. 2012 IEEE International Conference on Information Science and Technology, 2012, pp. 663–667, doi: 10.1109/ICIST.2012.6221729.

[29] Jiang, X.—Yao, H.—Zhang, S.—Lu, X.—Zeng, W.: Night Video Enhancement Using Improved Dark Channel Prior. 2013 IEEE International Conference on Image Processing, 2013, pp. 553–557, doi: 10.1109/ICIP.2013.6738114.

[30] Fu, Z.—Yang, Y.—Shu, C.—Li, Y.—Wu, H.—Xu, J.: Improved Single Image Dehazing Using Dark Channel Prior. Journal of Systems Engineering and Electronics, Vol. 26, 2015, No. 5, pp. 1070–1079, doi: 10.1109/JSEE.2015.00116.

[31] Viola, P.—Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Vol. 1, 2001, doi: 10.1109/CVPR.2001.990517.

[32] Dalal, N.—Triggs, B.: Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05), Vol. 1, 2005, pp. 886–893, doi: 10.1109/CVPR.2005.177.

[33] Krizhevsky, A.—Sutskever, I.—Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM, Vol. 60, 2017, No. 6, pp. 84–90, doi: 10.1145/3065386.

[34] Felzenszwalb, P. F.—Girshick, R. B.—McAllester, D.—Ramanan, D.: Object Detection with Discriminatively Trained Part-Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, 2010, No. 9, pp. 1627–1645, doi: 10.1109/TPAMI.2009.167.

[35] Zaidi, S. S. A.—Ansari, M. S.—Aslam, A.—Kanwal, N.—Asghar, M.—Lee, B.: A Survey of Modern Deep Learning Based Object Detection Models. Digital Signal Processing, Vol. 126, 2022, Art. No. 103514, doi: 10.1016/j.dsp.2022.103514.

[36] He, K.—Zhang, X.—Ren, S.—Sun, J.: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 37, 2015, No. 9, pp. 1904–1916, doi: 10.1109/TPAMI.2015.2389824.

[37] Lin, T. Y.—Dollár, P.—Girshick, R.—He, K.—Hariharan, B.—Belongie, S.: Feature Pyramid Networks for Object Detection. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2117–2125, doi: 10.1109/CVPR.2017.106.

[38] Dai, J.—Li, Y.—He, K.—Sun, J.: R-FCN: Object Detection via Region-Based Fully Convolutional Networks. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 29 (NIPS 2016). Curran Associates, Inc., 2016, pp. 397–387.

[39] Qiao, S.—Chen, L. C.—Yuille, A.: Detectors: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 10213–10224, doi: 10.1109/CVPR46437.2021.01008.

[40] Lin, T. Y.—Goyal, P.—Girshick, R.—He, K.—Dollár, P.: Focal Loss for Dense Object Detection. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[41] Duan, K.—Bai, S.—Xie, L.—Qi, H.—Huang, Q.—Tian, Q.: CenterNet: Keypoint Triplets for Object Detection. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6568–6577, doi: 10.1109/ICCV.2019.00667.

[42] Tan, M.—Pang, R.—Le, Q. V.: EfficientDet: Scalable and Efficient Object Detection. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10778–10787, doi: 10.1109/CVPR42600.2020.01079.

[43] Liu, S.—Qi, L.—Qin, H.—Shi, J.—Jia, J.: Path Aggregation Network for Instance Segmentation. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.

[44] Ghiasi, G.—Lin, T. Y.—Le, Q. V.: NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 7029–7038, doi: 10.1109/CVPR.2019.00720.

[45] Hu, J.—Shen, L.—Sun, G.: Squeeze-and-Excitation Networks. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[46] Wang, Q.—Wu, B.—Zhu, P.—Li, P.—Zuo, W.—Hu, Q.: ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11531–11539, doi: 10.1109/CVPR42600.2020.01155.

[47] Li, X.—Wang, W.—Hu, X.—Yang, J.: Selective Kernel Networks. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 510–519, doi: 10.1109/CVPR.2019.00060.

[48] Woo, S.—Park, J.—Lee, J. Y.—Kweon, I. S.: CBAM: Convolutional Block Attention Module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.): Computer Vision – ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11211, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.

[49] Fu, J.—Liu, J.—Tian, H.—Li, Y.—Bao, Y.—Fang, Z.—Lu, H.: Dual Attention Network for Scene Segmentation. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3141–3149, doi: 10.1109/CVPR.2019.00326.

[50] He, Y.—Zhang, X.—Sun, J.: Channel Pruning for Accelerating Very Deep Neural Networks. Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1398–1406, doi: 10.1109/ICCV.2017.155.

[51] Zhu, M.—Gupta, S.: To Prune, or Not to Prune: Exploring the Efficacy of Pruning for Model Compression. CoRR, 2017, doi: 10.48550/arXiv.1710.01878.

[52] Kwon, S. J.—Lee, D.—Kim, B.—Kapoor, P.—Park, B.—Wei, G. Y.: Structured Compression by Weight Encryption for Unstructured Pruning and Quantization. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 1906–1915, doi: 10.1109/CVPR42600.2020.00198.

[53] Anwar, S.—Hwang, K.—Sung, W.: Structured Pruning of Deep Convolutional Neural Networks. ACM Journal on Emerging Technologies in Computing Systems

(JETC), Vol. 13, 2017, No. 3, Art. No. 32, doi: 10.1145/3005348.

[54] Tan, Z.—Song, J.—Ma, X.—Tan, S. H.—Chen, H.—Miao, Y.—Wu, Y.—Ye, S.—Wang, Y.—Li, D.—Ma, K.: PCNN: Pattern-Based Fine-Grained Regular Pruning Towards Optimizing CNN Accelerators. 2020 57th ACM/IEEE Design Automation Conference (DAC), 2020, pp. 1–6, doi: 10.1109/DAC18072.2020.9218498.

[55] de Kruif, B. J.—de Vries, T. J. A.: Pruning Error Minimization in Least Squares Support Vector Machines. IEEE Transactions on Neural Networks, Vol. 14, 2003, No. 3, pp. 696–702, doi: 10.1109/TNN.2003.810597.

[56] Yang, M.—Faraj, M.—Hussein, A.—Gaudet, V.: Efficient Hardware Realization of Convolutional Neural Networks Using Intra-Kernel Regular Pruning. 2018 IEEE 48th International Symposium on Multiple-Valued Logic (ISMVL), 2018, pp. 180–185, doi: 10.1109/ISMVL.2018.00039.

[57] He, Y.—Kang, G.—Dong, X.—Fu, Y.—Yang, Y.: Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks. CoRR, 2018, doi: 10.48550/arXiv.1808.06866.

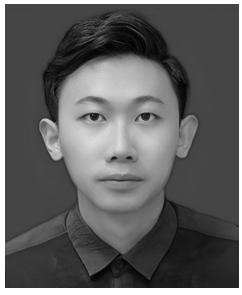
[58] Zhao, H.—Shi, J.—Qi, X.—Wang, X.—Jia, J.: Pyramid Scene Parsing Network. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6230–6239, doi: 10.1109/CVPR.2017.660.

Yonghao Liu received his Master degree in the Department of Communication and Information System, Yunnan University. He is currently pursuing his Ph.D. in the School of Computer Science and Technology, HUST. His research interests include image processing and neural network.

Hongwei Ding is Professor and Ph.D. Supervisor with the Yunnan University. He is mainly engaged in deep reinforcement learning and generative adversarial networks and has published many scientific papers indexed by SCI and EI.

Zhijun Yang is External Professor and Master's Tutor of the Yunnan University, mainly engaged in deep reinforcement learning and generative adversarial network research, and has published several academic papers indexed by SCI and EI.

Qianxue Xu is currently pursuing her B.Sc. degree in communication engineering at the Yunnan University. Her research interests include deep learning, image processing and object detection.



Guangen Ding is currently pursuing his M.Sc. degree in real estate management at the Yunnan University of Finance and Economics. His research interests include deep learning, building crack detection.

Peng Hu is working in the R & D Department of Kunming Ubay Technology Co. in China. His research interests include communication and information systems, deep learning and object tracking.