IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 28, 2020, accepted October 15, 2020, date of publication October 22, 2020, date of current version November 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3032981

Object Recognition at Night Scene Based on DCGAN and Faster R-CNN

KUN WANG AND MAO ZHEN LIU
College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China

Corresponding author: Mao Zhen Liu (maozhenliu@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant U1733119, and in part by the Central University Basic Scientific Research Business Fee Project of Civil Aviation University of China under Grant 3122018C001.

ABSTRACT In the past few years, with the rapid development of computer execution capabilities, target recognition strategies based on convolutional neural networks have become mainstream algorithms in the field of object detection. However, due to the blurred background and dim light, the object detection task in the night environment still faces greater visual challenges. This article is strongly inspired by DCGAN (Deep Convolution Generative Adversarial Networks). We use night images as input, generate virtual target scenes similar to the daytime environment through game training of generators and discriminators; and to obtain high-precision detection results, we combine the currently very advanced Faster R-CNN (Region-based Convolution Neural Networks) target detection system, through deep convolution feature fusion and multi-scale ROI (Region Of Interest) pooling. A series of experimental results show that our method achieves an mAP of 82.6% in the detection of its own night scene dataset, which is significantly higher than the original Faster R-CNN alone of 80.4%. Therefore, our method can meet the actual needs of target detection tasks in night scene. We sincerely hope that our approach will contribute to future research.

INDEX TERMS Convolutional neural network, DCGAN, faster R-CNN, night scene, object detection.

I. INTRODUCTION

The improved performance of computer hardware makes it possible to efficiently process large-scale training data. In recent years, with the rise of machine learning and deep learning algorithms, computer vision has performed well in fields such as visual recognition, speech recognition, and natural language processing. In the field of object detection, the task of recognizing night scenes is of great significance to discover potential objects in the environment in time. Some scholars use excellent algorithms to reduce the negative effects of weak light on the judgment of foreground objects, and some of the models have been applied in daily life and military use. Meis *et al.* [1] proposed an algorithm that mainly includes three parts to improve the accuracy of night traffic obstacle detection and classification. It uses a passive far-infrared sensor to focus on non-luminous objects: a classifier first finds the region of interest that contains hidden targets, and a region-based segmentation algorithm is used to re-segment the ROI obtained by the classifier. The type of object is then determined by a quadratic polynomial classifier. In 2012, Zhang *et al.* [2] proposed a night-time moving target detection algorithm based on dual-scale Approximate Median Filter back-ground models. The algorithm first performs block-wise down-sampling on the original image to reconstruct the low-resolution image and is used to roughly detect the ROI of the moving object; and then the moving object contour is refined by using coarse detected result ROI and the original image. Govardhan *et al.* [3] proposed a robust algorithm with a high detection rate and a low false alarm rate for night pedestrian detection systems. It uses different image subsets and different sizes to train a tree-like classifier to solve the problem of large intra-class differences in pedestrian poses. Sonu *et al.* [4] introduced the hot-spot and the background subtraction algorithms for detecting humans in night vision video. The former uses black body radiation theory and the latter uses difference images obtained from the input image and the generated background image. In 2018, Bazán Caballero and Zamudio Beltrán [5] proposed a computer vision system capable of identifying traffic panels (TP) in night scenes. The size of the original image can be reduced by 74% during the preprocessing stage of the system, cropped to obtain the upper two thirds of the image, and finally converted to grayscale. In the processing

The associate editor coordinating the review of this manuscript and approving it for publication was Tomasz Trzcinski.

stage, in order to identify the region of interest (ROI), cascade object detector (COD) is used to process the output image from the final stage. In the classification stage, the algorithm uses COD to divide each ROI into one of TP and no TP, and finally highlights each ROI classified as TP with a bounding box. Recently, an estimation algorithm of illumination maps like LIME [6] was proposed by X. Guo to construct the illumination maps by finding the maximum intensity of the original image channel. Similarly, some scholars have further proposed Robust-Retinex [7] based on retinex theory. However, the decomposition of observed brightness is an ill-posed problem that has not been solved well so far. For those enhancement algorithms with low illumination, it is very difficult to obtain the ground truth of the corresponding object. LLNet [8] first applied the deep auto-encoding method to identifying signals from low-brightness images and adaptively brightening the image without over-magnifying the brighter part. Tao *et al.* [9] proposed a two-step strategy based on the atmospheric scattering illumination model to enhance low-light images. Wei *et al.* [10] proposed a Retinex-Net algorithm that includes DecomNet for decomposition and EnhanceNet for brightness adjustment. In the last two years, Xiao *et al.* [11] used a specially designed feature pyramid network and context fusion network to enhance images under low-illuminance to improve detection results. Li *et al.* [12] used HSV color space features instead of the traditional RGB space to enhance the robustness of video contrast and color distortion. Zhu *et al.* [13] proposed a novel multitemporal monitoring image change detection algorithm under low-illuminance by fusing different images to remove noise, and achieved good experimental results. The dimness of light is usually the key factor that affects our judgment of foreground objects. How to accurately classify and learn the distribution of target subjects our concern. Generative Adversarial Networks' game theory has brought inspiration to our nighttime object detection task. Therefore, due to the difficulty of night target recognition, we propose a night object detection method based on Faster R-CNN and Deep Convolution Generative Adversarial Networks. To some extent, the traditional method has the problems of high cost and complicated process. The deep learning method of our model uses only two main modules, which effectively reduces the complexity of the network and at the same time has a better way to deal with the dark objects that are not shining in the previous work.

Convolutional neural network is a common deep learning framework proposed by some scholars inspired by the visual neural system in which the organism perceives external things. It has the characteristics of local connection, weight sharing and automatic feature extraction. Its development can be traced back to the study of the visual system in the cat brain by Hubel and Wiesel [14] in 1962. In 1980, Kunihiko and Sei [15] proposed a neural network structure consisting of convolution and pooling layers. In 1990, Yann Lecun first applied back propagation algorithm (BP) in training of neural network structure, which formed the prototype of modern convolutional neural network, but due to the difficulty of network training and the poor performance in actual tasks, worse than SVM, Boosting algorithm once fell into a low tide.

With the improvement of GPU accelerator performance and the expansion of public data sets, the deepening of the number of CNN layers and the improvement of detection accuracy has revealed the potential development of network, which has once again attracted the attention of researchers. In 2012, AlexNet [16] used a classic CNN structure to produce a great performance breakthrough in image recognition. After the success of AlexNet, Ross Girshick *et al.* proposed a regional convolutional neural network in 2014. The network uses a selective search algorithm to generate region proposals and train SVM for classification. In order to solve the shortcomings of input images size and large resource consumption, Kaiming He *et al.* proposed a SPP-net using spatial pyramid pooling algorithm [17]. In 2015, Ross Girshick proposed fast region-based CNN (Fast R-CNN) [18], which uses two different layers that are fully connected to complete the tasks of target classification and bounding box positioning, instead of training SVM separately, which saves a lot of storage space. Unfortunately, Fast R-CNN still applies the selective search algorithm to obtaining fixed-size region proposals. The system cannot achieve end-to-end network training, and the back-propagation algorithm cannot improve the extraction process of region proposals. Based on Fast R-CNN, S. Ren *et al.* proposed the Faster R-CNN [19], the network uses a novel RPN to obtain region proposals instead of the previous selective search algorithm. It reduces some calculations through parameter sharing and reaches a new height in the field of object detection. Afterwards, a large number of scholars have made improvements, such as [20]–[22]. The continuous improvement of network performance has benefited from the continuous innovation and optimization of the network structure. In order to obtain more comprehensive information of night images, a good strategy is convolution feature fusion, which combines rich detailed in-formation with abstract semantic information [23].

GAN [24] is a model proposed by Goodfellow *et al.* In 2014, its core idea was two-player game theory. Experimental results show that the network seems to be able to generate sample images that look like genuine things. The DCGAN [25] introduces the ideas in GAN into the convolutional neural network, which uses the latter's powerful feature extraction capabilities in the field of image processing to generate higher quality sample images. Subsequently, the GAN series network was continuously concerned and developed by researchers [26]–[29]. Recently, some scholars have studied the problem of DCGAN and convolutional networks for target detection [30] [31]. We all know that low-light scenes are closely related to our life. At present, the development of the corresponding intelligent vision system still needs further research. In this paper, we use the preprocessed night image as the input signal of the DCGAN network, instead of the original 100-dimensional random noise. The system generates virtual images similar to daytime scenes and uses

them with advanced detectors to complete actual tasks. This method achieves good performance in night object detection.

II. RELATED WORK

A. DCGAN ALGORITHM PRINCIPLE

DCGAN introduces convolutional networks into the structure of GAN. By optimizing hyper parameters and network topology, it makes up for the gap between CNN in the fields of supervised learning and unsupervised learning. It is mainly composed of two multi-layer perceptrons(generator and discriminator). The former attempts to capture the potential distribution of real samples, establish spatial pixel connections and generate sample data at the same time. The latter is essentially a binary classifier, used to determine whether the input of the model comes from the real sample or the sample generated by the generative model. In other words, D and G play the following two-player mini-max game with value function $V\ (G, D)$. During the network training phase, both players maximize their profits, and finally reach a Nash equilibrium point, i.e. the generator generates sample data that the discriminator cannot judge the true and false, and the value function finally converges to V^*:

$$V^* = \arg\min_G \max_D V(G, D) \tag{1}$$

The definition of the objective function is as follows:

$$V(G, D) = E_{x\sim P_{data}}[\log(D(x))] + E_{x\sim P_G}[\log(1 - D(x))] \tag{2}$$

P_{data} and P_G in the expression represent the true sample probability distribution and model distribution respectively.

Fig. 1 illustrates the details of the deep convolutional generative adversarial networks. The generator G takes a 100-dimensional noise vector as input and projects, reshapes as a small-scale convolution space, then uses a 4-layer fractionally-strided convolutions operation to topological space structure, and gets a sample image with a size of $64 \times 64 \times 3$. The structure of the discriminator can be regarded as a flip of the generator. The difference is that the final network only outputs a simple discriminant value for judging the source of the data. It is worth noting that the training of the generator and the discriminator is carried out alternately. When updating the parameters of one side, the other side remains unchanged, i.e. the G is fixed first, and the parameters in D are trained to maximize the value of V(G, D), immediately following keeping the weight parameters in D stable, the network G is trained to minimize the value of $\max_D$ V(G, D) to get the excellent generator as expected.

B. FASTER R-CNN

Faster R-CNN is mainly composed of two modules, the first module is a fully convolutional network for generating regional proposals, and the second module is a Fast R-CNN detector. The entire system can be trained end-to-end by back-propagation and stochastic gradient descent (SGD), and the RPN network uses a popular attention mechanism

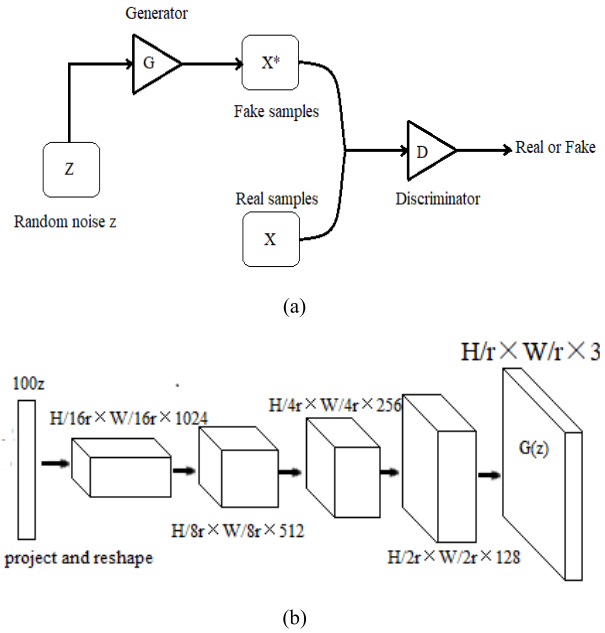


FIGURE 1. Algorithm flows and structure diagram of DCGAN. (a) Network algorithm flows diagram (b) Generator architecture diagram, the parameter z represents standard normal distribution or uniformly distributed random noise.

to tell Fast R-CNN the spatial location of the target. The biggest highlight of the object detection network is to propose a method to effectively locate the target area, and then index the features on the feature map according to the area, which greatly reduces the time consumption of convolution calculation. Fig. 2 shows the structural details of Faster R-CNN.

The deep convolutional network in this target detection system, such as VGGNet [32] or ResNet [33], can extract the main target features of grayscale or color maps. However, with the deepening of the network, while enriching the semantic information, the low-resolution feature map also causes the loss of a lot of detailed information. The anchor box algorithm assigns k (usually $k = 9$) bounding boxes to each pixel of the shared convolution feature map output by the deep convolutional network, and uses a 3×3 sliding window to perform the convolution operation from the upper left corner to the lower right corner of the convolutional layer. Two parallel 1×1 convolution layers respectively obtained the category (foreground or background) and quaternion coordinates of k bounding boxes. For the bounding box with positive label (foreground), the predicted bounding box is obtained through the first coordinate fine-tuning, crop the region of interest in the Align ROI Pooling layer and use bilinear interpolation to obtain the ROI and normalize it to a $H \times W$ feature map. Finally, two completely connected layers and softmax are used to complete accurate classification and bounding box regression (the second fine-tuning operation, using a non-maximum suppression algorithm, where the predicted bounding box of IOU > 0.7 is assigned to a positive label, and IOU < 0.3 is assigned a negative label) task.

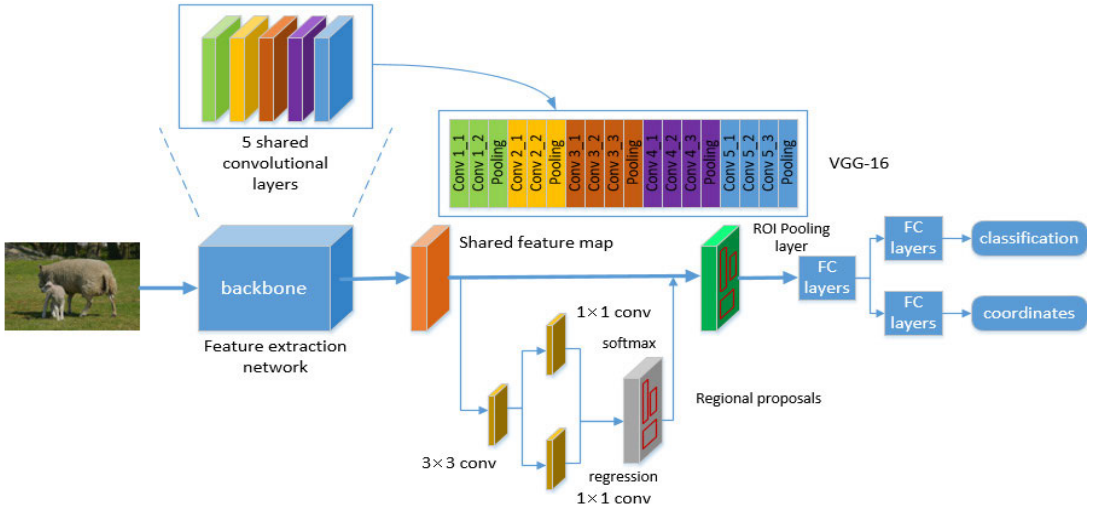


FIGURE 2. Schematic diagram of Faster R-CNN.

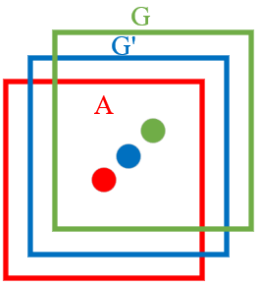


FIGURE 3. Schematic diagrams of the bounding box regression algorithm. The red box A is the raw predicted bounding box output by our fully connected layer, the green box G is the ground truth (gt), and the blue box G′ is our regression expectation after the fine-tuning operation.

1) BOUNDING BOX REGRESSION

Bounding box regression is a meaningful algorithm in Faster R-CNN. The system outputs the category of the foreground target and the predicted four-dimensional coordinate vector of the bounding box through two parallels fully connected layers. However, the coordinate information directly generated is inaccurate. Therefore, we need to perform bounding box correction to optimize the obtained coordinate information. The details of the algorithm are described below.

Fig. 3 graphically shows the position of the border and our operating intention. We tried to find a mapping relationship to make the predicted bounding box closer to ground truth and reduce the positioning bias. Previous work also showed that targets detection relies on border regression to achieve accurate positioning.

We have obtained window coordinated vectors for $A = (x_a, y_a, w_a, h_a)$ and $G = (x, y, w, h)$, now we need to find a transformation F to meet the requirements:

$$F(xa, ya, wa, ha) = (x', y', w', h') = G' \tag{3}$$

With:

$$(x', y', w', h') \approx (x, y, w, h) \tag{4}$$

Get the translation and scaling factor from A to G':

$$t_x = \frac{(x - x_a)}{w_a}, \quad t_y = \frac{(y - y_a)}{h_a},$$
$$t_w = \log(\frac{w}{w_a}), \quad t_h = \log(\frac{h}{h_a}) \tag{5}$$

In the regression task, the translation amounts (t_x, t_y) and the zoom scales factor (t_w, t_h) completed the fine-tuning of the border. In the Faster R-CNN network, there are two corrections of the frame coordinates. The first time is the fine-tuning of the anchor box in RPN, and the second time is in the classification part of the network, along with the NMS (non-maximum suppression) algorithm to retain mature bounding boxes. The formula also applies to the former.

2) LOSS FUNCTION

RPN uses shared convolutional feature maps as input for end-to-end training, and produces high-quality regional proposals while adjusting parameters. As part of the parallel network, Fast R-CNN uses feature maps, region proposals and back-propagates gradients based on the cross-entropy loss function to adjust subject parameters. The loss of Faster R-CNN mainly includes cls loss and bbox regression loss. The following is its calculation formula:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{6}$$

where i represents the index of the border in the mini-batch, and p_i represents the probability that the predicted bounding box contains the target. If the border corresponding to the special index value is assigned a positive label (IOU $>$ 0.7 for the Ground Truth bounding box), then $p_i^* = 1$; otherwise the case is that when the corresponding border is assigned a negative label (IOU $<$ 0.3), $p_i^* = 0$. Note that $p_i^* L_{reg}(t_i, t_i^*)$ indicates that the bounding box regression is

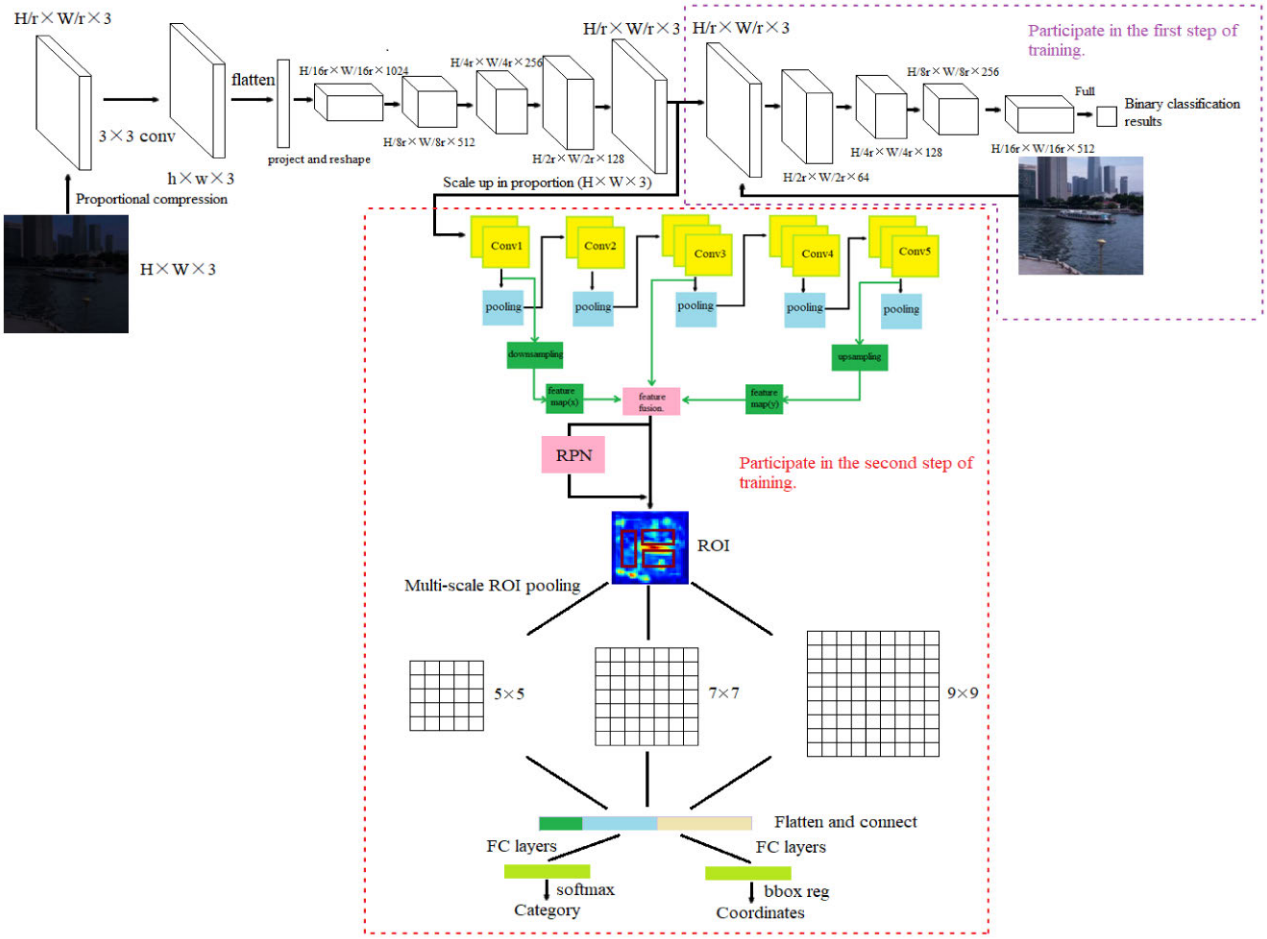


FIGURE 4. The overall framework of the algorithm used in this paper, where r is an adjustable proportional parameter. The purple dotted line surrounds the discriminator model, and the content in the red dotted frame is our improved Faster R-CNN.

only performed when the sample is positive. t_i and t_i^* are two independent four-dimensional coordinate vectors, where $t_i = \{t_x, t_y, t_w, t_h\}$ represents the parameterized coordinates of the predicted bounding box and $t_i^* = \{t_x^*, t_y^*, t_w^*, t_h^*\}$ is related to the details of the GT bounding box. N_{cls} is equal to the network mini-batch size (for example, $N_{cls} = 256$). N_{reg} is the number of anchor positions (i.e., $N_{reg} \sim 2400$). Due to the needs of actual tasks, RPN and Fast R-CNN are usually alternately trained separately. The classification loss of RPN is a two-class cross entropy loss. Unlike it, Fast R-CNN is a multi-class cross entropy loss. The purpose of introducing the adjustable parameter λ is to balance the classification and regression loss. L_{cls} and L_{reg} indicate classification and regression loss respectively, and their definitions are as follows:

$$L_{cls}(p_i, p_i^*) = -\log[p_i^* p_i + (1 - p_i^*)(1 - p_i)] \quad (7)$$

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \quad (8)$$

Among them, R is the Smooth L1 function.

III. DETAILS OF OUR ALGORITHM

Inspired by the idea of generating game theory against the network, we innovatively proposed an algorithm specifically for night scene object detection. We use night-time color images instead of 100-dimensional random noise in DCGAN as the input of the network, and daytime images of the same scene as real data samples, trying to make the network learn the spatial pixel distribution between the two as much as possible. Then we use the virtual data samples generated by the generator as the input of the improved Faster R-CNN network and display the bounding box and target category on the night image. In Part 4, the experiment shows that step training has a good convergence effect. When two steps are trained at the same time, we observe that it is difficult to learn the model parameters, and it is easy to collapse the model. The complete structure of our network is shown in Fig. 4. Without bells and whistles, the first step is to first train the generator and discriminator models unfairly, and keep the weight parameters of the generator unchanged after the model is stable; the second step is to use the generator and Faster R-CNN to complete the final training, and the

discriminator model in the first step no longer participates in the activities here. We have covered the details of the training in Section 4.2 and will not elaborate here.

A. DIFFERENT DCGAN

How to establish the potential relationship between nighttime images and daytime images is the core problem we are concerned about. In order to solve this problem, using DCGAN's image generation capabilities seems to be a good strategy. Initially, we directly flatten the original night image into a $1 \times 1 \times C$ tensor by pixel to replace the random noise in the traditional method. However, this method presents an experimental phenomenon of unstable oscillation and is accompanied by a large number of noise signals, and the generated image has no regularity and cannot be recognized. We believe that this is the result of forced conversion without considering the characteristics of the spatial dimension. We only use a 3×3 size convolution kernel to use the target features in space. Unfortunately, in the training process of this strategy, multiple images tend to collapse into the same scene, and the expected sample diversity is severely lacking. However, what is inspiring is that the output image begins to show virtual features and regular patterns that can be distinguished by human vision, which indicates that the model has begun to learn features and establish mapping relationships between different scenes. However, this method cannot avoid numerical problems caused by non-normalization. Reference [25] mentioned the significant impact of normalization operation on the performance of gradient descent method. Therefore, after exploring the above methods, we have identified a series of model architectures that can perform stable training on our nightly dataset and allow training of more complex and higher-resolution generative models. As shown in Fig. 4, due to the limitations of the device, we first perform a preprocessing operation on the image, and reduce the network parameters by compressing the graphics of size $H \times W \times 3$ to the size of $H/r \times W/r \times 3$. A 3×3 convolution operation with a step size of 1 is used to generate the intermediate relationship image, and the tanh activation function is used to transform the final output to the $(-1,1)$ interval. Finally, it is flattened and transformed into a $4 \times 4 \times 1024$ feature map through the project and reshapes operation similar to the role of the fully connected layer. After the generator output, part of the convolutional layer of the discriminator is used for feature extraction. The feature information of the data sample is transmitted to a specific layer f through convolution operation, and finally the feature matrix is flattened and used as the input of the classifier.

Similar to the original DCGAN, we continue to use fractional strided convolutions and strided convolutions in the generator model and discriminator model, respectively, which will allow the model to learn its own spatial upsampling and subsampling. In order to accelerate the convergence and slow down the overfitting of the system, we use a very important Batch Normalization method in the field of deep learning. It is mentioned in [18] that the initialization of the parameters during the initial training of the network is uncertain, and this method can greatly improve the training difficulties caused by improper initialization, and also facilitates gradient flows in deeper models. Previous work has shown that the use of BN operations in all layers will cause the generated samples to oscillate. In this paper, we still only use this algorithm in the input layer of the discriminator and the output layer of the generator. It is worth mentioning that the obvious difference between the training phase and the previous network rules is that we no longer alternate training each time, but adjust the training ratio of the two. In other words, we train the generator network x times to update the discriminator parameters again. This is necessary. Because of the complexity of the image, it is easy for the generator to fail to fully learn the spatial distribution of the data samples in a timely manner. The most intuitive phenomenon is that the discriminator network loss gradually approaches 0, while the generator network loss gradually increases. Subsequent experiments also prove that our practices are correct.

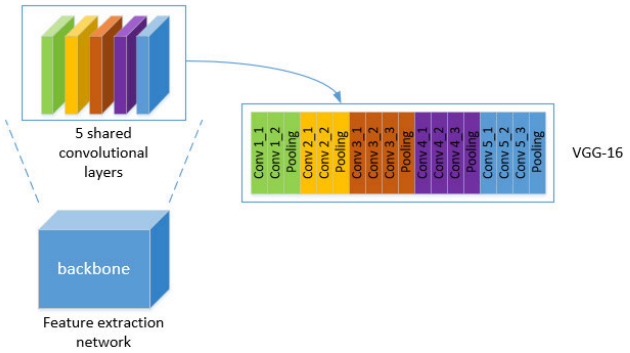


FIGURE 5. The structure of VGG-16 network.

B. MULTI-SCALE CONVOLUTION FEATURE FUSION

The traditional convolutional neural network extracts smaller feature maps related to object features through a series of convolution and pooling operations. This processing method usually causes the loss of some image information, making it difficult to fully express the target details in the night scene image. This is very unfriendly to the situation where we use the data samples generated by DCGAN with information distortion as the network training set.

To assist in expressing the ideas of our algorithm, we illustrate the structure of the previous VGG-16 network in Fig. 5. The network is mainly composed of 6 sub-modules. In general, some advanced object detectors only use the feature map output by Conv5_3 as the input of the subsequent network, such as Faster R-CNN.

The network structure based on the same-scale convolution feature fusion is shown in Fig. 6. As the depth of the network deepens, the detailed information gradually decreases. We use down-sampling and up-sampling methods to combine the features extracted by Conv1_2, Conv3_3, and Conv5_3 layers, respectively. A specific explanation is that Conv1_2 and Conv5_3 have the most comprehensive

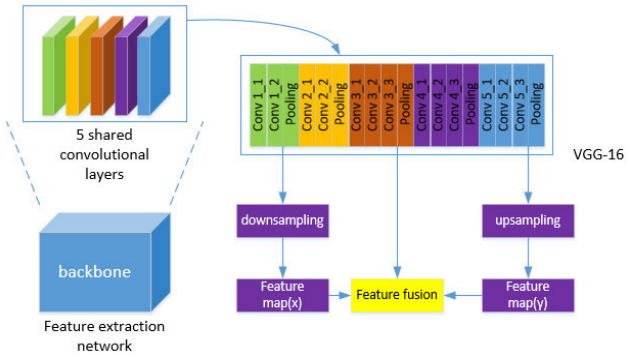


FIGURE 6. Schematic diagram of the network framework for the same-scale convolution feature fusion.

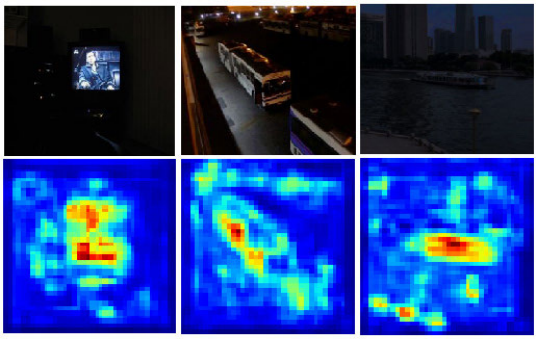


FIGURE 7. Visualization of the effect of the same-scale feature fusion strategy.

detailed information and the richest semantic information in the convolutional layer respectively, and they perfectly fit the balance information in the Conv_3 layer (Section 4.3 shows a series of ablation experiments). In order to solve the problems caused by the different sizes of the feature maps generated by each convolutional layer, we may use the size of Conv3_3 as a benchmark and use the down-sampling and up-sampling operations on the feature maps of Conv1_2 layer and Conv5_3 layer respectively, and make them reach the feature map with the same spatial resolution as the Conv3_3 layer. Therefore, a feature map containing three different levels of information is obtained.

An advanced approach before feature map fusion is to apply local response normalization to processing each extracted feature map so that they have the same activation value. The feature map after fusion has rich detailed information and abstract semantic information, as shown in Fig. 7.

C. MULTI-SCALE ROI POOLING

After the RPN generates a fine-tuned region proposal, the Align ROI Pooling layer first uses the bilinear interpolation operation to obtain the pixel value at the decimal point coordinate according to the obtained four-dimensional coordinate vector (x, y, w, h). Among them, x and y are the horizontal and vertical coordinate parameters of the center point of the region proposal, and w and h are the width and height parameters of the corresponding region, respectively. These four parameters are independent of each other. After cutting, many ROIs of different sizes are generated. Due to the limitation of the fully connected layer, maximum pooling is usually used to transform these efficient ROIs into smaller feature maps with a fixed H $\times$ W (i.e., 7 $\times$ 7) size.

ROI pooling reduces the calculation parameters and causes the loss of important features of the target object. In order to solve this unoptimistic problem, ROI pooling of different sizes is used. The motivation of our superior design may be to enable the system to capture more feature information. We designed three ROI pools of different sizes (5 $\times$ 5, 7 $\times$ 7, 9$\times$9). After performing maximum pooling on the feature map in parallel, they are connected for subsequent classification and bounding box regression.

IV. EXPERIMENT AND ANALYSIS

In this paper, we perform experimental analysis on the night dataset and day dataset we have marked. At the same time, in order to prove the effectiveness of our method, we primarily evaluate detection mean Average Precision (mAP), because this is a recognized and strict performance indicator for target detection.

A. PRODUCTION OF DATA SETS

We directly use the computer's camera to capture images of the daytime scene. These images constitute our daytime data set A. That is a multi-target data set with a total of 6,833 images in 10 categories. The detailed category descriptions are as follows: "person," "dog," "car," "cat," "bicycle," "bus," "chair,"" sofa,"" boat," and "TV monitor." The limitations of the data set will greatly affect the ability of the model to generate. In order to avoid DCGAN's generating bad images, we selectively collect some images from the Internet and extend the daytime data set to make the model generate images of higher resolution. An enlarged day-time dataset B containing 62511 pictures was obtained. As mentioned earlier, in order to make the model establish the approximate pixel change relationship from night to daytime, we need sample data of different time periods in the same scene, which is almost impossible to rely on our artificial shooting. Therefore, we manually adjust the brightness and saturation of the daytime images to simulate the nighttime scene as much as possible to generate our nighttime data set C. This is very necessary. The experiment found that the number of samples contained in the training data set will directly affect the quality of the images generated by the DCGAN model, because more data samples mean rich diversity. Benefiting from the label-free requirements of the DCGAN method training, we can use the expanded data set B and the corresponding data set C that change the brightness and saturation to simulate night scenes to specifically train DCGAN to generate better images. It can be intuitively regarded as a pre-training process. However, this strategy is no longer suitable for Faster R-CNN with a special supervision mechanism because it requires labels for training. In order to reduce the time consumption of manually

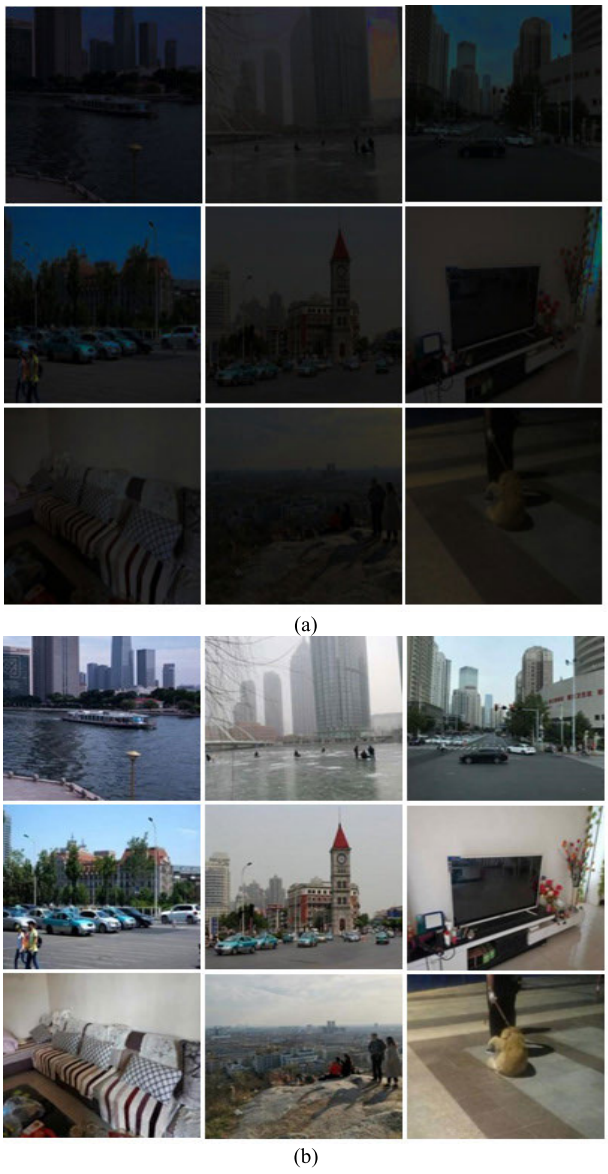


FIGURE 8. A display diagram of some samples in the dataset we used. Fig (a) is a night image. Fig (b) is the day-time image taken directly by our camera.

labeling objects, we choose to label data set A instead of large-capacity data set B to complete the overall training. Later experiments show that this method fully achieves the expected detection effect. In addition, in order to meet the needs of actual tasks, a meaningful approach is to finally test the performance of our model on real night images (the data set D that we shot directly and used only for model testing of actual tasks). In order to increase background diversity, different images have different brightness and saturation changes, and the images in the daytime and the nightly data set correspond to each other, as shown in Fig.8. Although the training process of DCGAN does not require the participation of labels, the parameters of the network are extremely huge. To reduce the image size too much to adapt to this situation will also affect the detection accuracy of Faster R-CNN for objects. In order to balance the different requirements of these two parts, we first unified the size of all collected images (day and night) to a base size of 256 × 256, and then completed the annotation work on the daytime data set.

Due to the limitations of the device, for the DCGAN part, we use proportional compression (for example, it becomes 128 × 128 after double compression) to preprocess the image for network input. For the Faster R-CNN part, the data samples (128 × 128) generated by the previous network are reduced to the size of the reference image through the proportional enlargement operation as the input of the detection network. Faster R-CNN training requires a lot of data and labels to support the work. However, data labeling is very time-consuming and tedious, which is also the main reason for limiting the size of our data set A. The lack of data samples will increase the probability of overfitting of the system, resulting in the model simply remembering the training set. In order to enhance the generalization ability of the model, we expand the original data set by flipping. The images in the final data set A were expanded to 13,666. In order to make full use of our data resources, we decided to use 60% of the samples (8,199 images) for training, 30% of the samples (4,100 images) for verification, and the remaining 10% of the samples (1,367 images) for testing. The roles to which the samples are assigned are randomly selected by the computer.

B. MODEL TRAINING

Traditionally, target detection in night scenes is usually based on image enhancement strategies, but the effect is extremely limited. Therefore, we consider a mechanism that can break this limitation and reconstruct the features to solve the problem. The DCGAN algorithm with a generation-confrontation game process is adopted. We use the Python programming language to build our overall detection model under the framework of Tensorflow deep learning. The basic program related to DCGAN used in the paper has been open source and can be obtained from this link: https://github.com/carpedm20/DCGAN-tensorflow. We only need to do a small amount of program modification, i.e. to switch the random noise input port in DCGAN to the night image we need, and then an important convolution operation and normalization strategy is used to ensure the gradient flow to build a new input. In addition, we keep the names of the night images and day images input into DCGAN consistent to ensure the correct relationship mapping during the batch training process. In the experiment, the model training is mainly divided into DCGAN training and Faster R-CNN training. The parameter details are shown below.

First, for the DCGAN model, in order to establish a potential spatial variation relationship, we use the daytime data set B as the real sample input of the discriminator, and the nighttime data set matching it as the generator input. The network goes through mini-batch stochastic gradient descent (SGD) and sets a learning rate of 0.0002 to complete the training, where mini-batch is set to 128 a priori by us. A normal distribution with zero mean and standard deviation

of 0.02 is used as the initialization method for all layer weight parameters. It is mentioned in [25] that all layers of the network use the Batch Normalization (BN) algorithm, which will lead to sample oscillation and model instability, so that we still use the BN algorithm only in the output layer of the generator and the input layer of the discriminator. We use the tanh activation function of the input and output layers of the generator model, and the remaining layers are activated using relu. However, all layers of the discriminator model use the leaky relu activation function to enable the generator to generate high-resolution images, where the slope of the leak was set to 0.2. In addition, the system uses Adam optimizer with tuned hyper-parameters and momentum $\beta_1 = 0.5$. We train all the pictures for 75 epochs unevenly instead of the traditional alternating training, and we set the training ratio of the generator to the discriminator to 2: 1. In other words, the discriminator updates the parameters only once after the generator is trained twice. This method can train the generator model more fully and enable the generator to generate high-resolution images.

After the previous first step training, we keep the generator model weights unchanged and use the DCGAN generator model to match the Faster R-CNN detector to perform the second step training, and the discriminator no longer participates in this process. Faster R-CNN's RPN can be trained end-to-end through back-propagation and stochastic gradient de-scent (SGD). We use VGG-16 pre-trained on ImageNet classification as a deep convolution feature extraction network, and randomly initialize all other new layers by drawing weights from a zero-mean Gaussian distribution with standard deviation 0.01. We use a learning rate of 0.001 for 30k mini-batches, and 0.0001 for the next 10k mini-batches on our benchmark nightly dataset, where mini-batches are set to 128. We use a momentum of 0.9 and a weight decay of 0.0005. A dropout algorithm to prevent network overfitting is used in the fully connected layer, and the correlation factor is set to 0.5. Finally, the model was trained by GPU acceleration. Because the images of the night and day scenes contain the same target information, our previous label in one scene can be applied to another different scene, which is convenient for the recognition task here. The system extracts the feature information of the virtual image generated by the generator, and draws the target's bounding box and category on the night image.

C. EXPERIMENTAL RESULTS AND EVALUATION

In this paper, in order to be able to perfectly combine the DCGAN models with the improved Faster R-CNN model, we intend to study how to fully establish the feature distribution relationship between the night scene graph and the daytime scene graph. The use of pixel encoding of the image to introduce the image into the network and realize the back propagation of the gradient flow is the key to the problem. Through a series of attempts, the size of a 3×3 convolution algorithm is used in the initial feature extraction. The tanh activation functions transform the data information into the interval $(-1, 1)$. After flattening, it is a directed vector that is approximately equivalent to the original noise. Our method makes the model tend to be stable, and the sample data generated by the generator is good and has distinctive object characteristics, as is shown in Fig. 9. As can be seen from the randomly presented virtual image, the generator model generates a multi-channel image with clear outline and easy identification. We can use it as an intermediate image and combine the invariance of object labels to train the subsequent detection network, and finally successfully display the bounding box and specific category information on the night scene image, which is very meaningful.

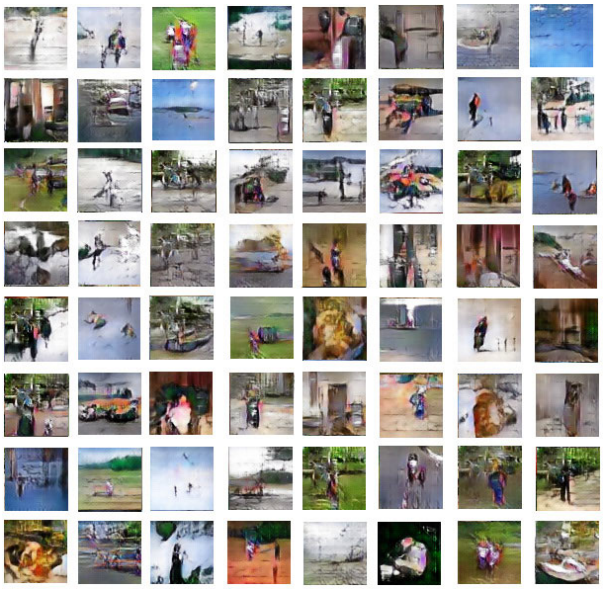


FIGURE 9. Sample images generated by DCGAN on the night dataset.

It is worth mentioning that this potential relationship between night and day images can be described as a spatial mapping [25], and the establishment of this relationship is done by DCGAN. In this paper, for convenience, we set the night image sample set input by the generator to $\psi = \{x_1, x_2, \ldots, x_n\}$, and the day image sample set input by the discriminator to $\phi = \{y_1, y_2, \ldots, y_n\}$, where n represents the number of samples contained in the sample set and x_N corresponds to y_N. The key to our solution is to find an intermediate mapping set $f = \{f_1, f_2, \ldots, f_n\}$ that satisfies $f(\psi) \approx \phi$, and each sub-mapping in f is obtained by training. They can essentially be regarded as a series of nonlinear combinations of network weights. Specifically, for different night images x_1 and x_3, due to their different characteristics, they will get different mapping results f_1 and f_3 after being combined with excellent weight parameters. Therefore, $f_1(x_1) \approx y_1$ and $f_3(x_3) \approx y_3$ are obtained, for f can be divided into two processes. In the training phase, the iteration of the sample makes the virtual image generated by the generator at night tend to have the characteristics of the day image, and the weighted combination of those network parameters forms a bridge to establish the relationship between the two. When the loss between the virtual image generated by this

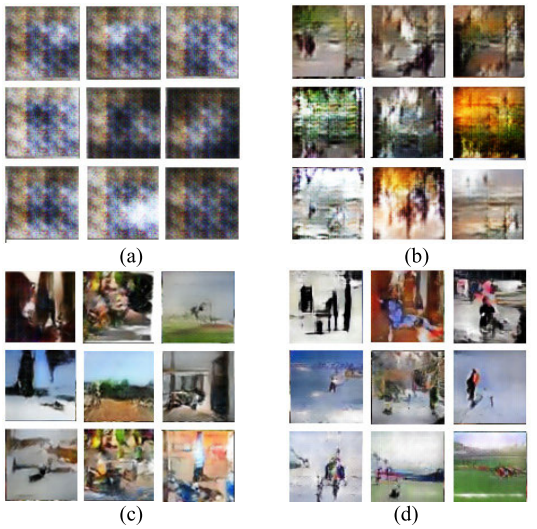


FIGURE 10. A visual diagram of the process of DCGAN establishing the image relationship mapping between night and day scenes. The result generated by the generator: (a) After 10epochs (b) After 30epochs (c) After 50epochs (d) After 70epochs.

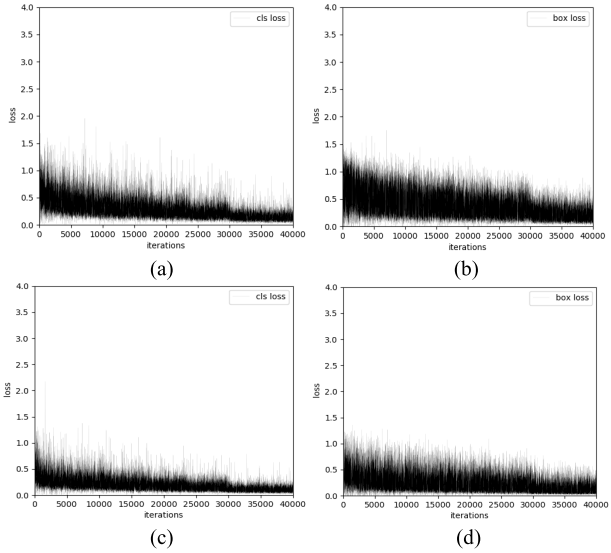


FIGURE 11. Fluctuation graphs of the loss function trained by the different system on the night dataset. Original Faster R-CNN: (a) cls loss curve (b) box loss curve. Our method: (c) cls loss curve (d) box loss curve.

algorithm and the day image (standard or final target) is within an acceptable range, the weight parameters will be retained because they contain good feature maps. In the testing phase, we directly use the generator as the front-end network, and use its generated virtual image instead of the night scene image as the original input for the subsequent detection network. The visualization process of DCGAN's gradually establishing the potential relationship (mapping) between night and day images is shown in Fig. 10.

In order to prove the effectiveness of our method, we compared the detection effects of different algorithms on the night dataset. Fig. 11 is the change curves of two loss functions that use Faster R-CNN and our method to train 40,000

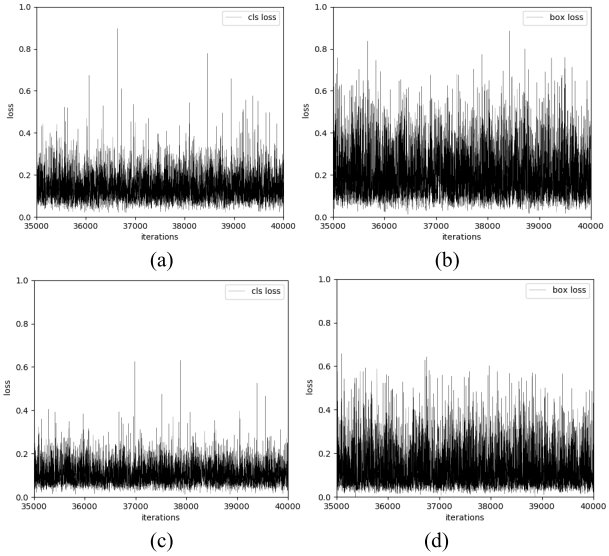


FIGURE 12. The change curves of the loss function of the different system after 5000 iterations. Original Faster R-CNN: (a) cls loss curve (b) box loss curve. Our method: (c) cls loss curve (d) box loss curve.

iterations on the same nightly data set. It can be observed from the displayed trend graph that both methods can be stably trained and gradually converged on our nightly data set. The difference is that our model has better performance in object classification and bounding box regression tasks. In particular, we show the change of the loss function in the last 5,000 iterations of the system to further more easily observe the details of the oscillation and con-vergence of the curve, as is shown in Fig. 12.

Experiments show the effectiveness of our method on the boundary boxes positioning and target classification tasks. In the last 5000 iterations of the system, our method reduced the loss of both tasks to less than 0.7, while the maximum fluctuation amplitude of the original method is large, which is around 0.9 (see Fig. 12). Next, we evaluated the mean average precision (mAP) of the different methods to make the strategy more convincing. Table 1 shows the detailed test results of the ablation experiment. Our method has an mAP of 82.6%, higher than the original Faster R-CNN (7×7) which has an mAP of 80.4%. The line chart in Fig. 13 visually compares the changes in average precision of the different network models in various categories. Fig. 14 presents the comparison results of the algorithm for mAP values. The model has overall competitiveness and the potential to handle more types of actual target detection tasks.

D. RESEARCH ON FUSION OF DIFFERENT CONVOLUTIONAL LAYERS

In this part, we paid special attention to the influence of the fusion of different combinations of convolutional layers in VGG-16 on the detection accuracy. The core of the feature fusion algorithm is to combine the complete detailed information in the low layer and the rich semantic information

TABLE 1. Detection results on the night scene test set with two methods and VGG-16. For RPN, the train proposals for Faster R-CNN are 2000. "Faster": Faster R-CNN, "Faster R-CNN": Original Faster R-CNN.

Network model	Data	mAP(%)	bic	boat	bus	car	cat	chair	dog	person	sofa	tv
Faster R-CNN (5×5)	night	76.4	77.5	69.7	78.0	77.2	79.8	76.4	78.7	79.2	79.9	68.6
Faster R-CNN (7×7)	night	80.4	82.8	74.1	83.9	83.3	83.1	80.0	80.8	82.8	82.7	70.7
Faster R-CNN (9×9)	night	81.1	83.2	75.7	84.1	81.4	84.9	82.3	81.5	83.6	82.8	71.1
Faster + Multi-scale pooling	night	81.6	83.1	75.9	83.8	83.5	85.3	82.7	82.1	83.3	83.9	72.4
Faster + Feature fusion	night	81.3	82.5	76.4	84.7	82.9	84.0	83.1	81.6	82.9	83.4	71.2
DCGAN + Faster R-CNN	night	81.8	83.4	76.6	84.6	83.3	85.9	82.0	82.5	83.1	83.8	72.9
Our method	night	**82.6**	**85.6**	**78.0**	**85.1**	**84.0**	**86.2**	82.9	**83.1**	**84.4**	**84.6**	**73.1**

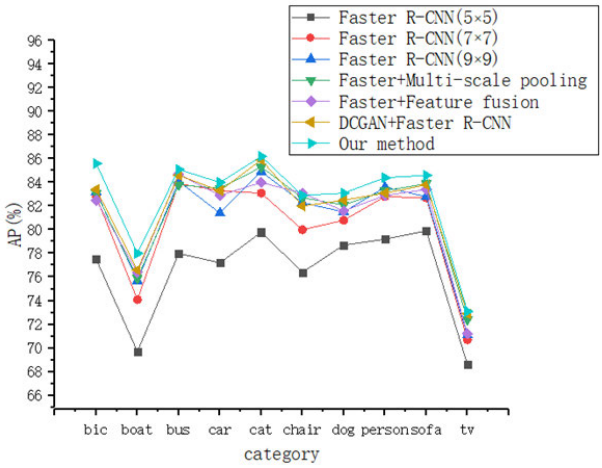


FIGURE 13. A line chart of Average Precision in 10 categories when using different methods.

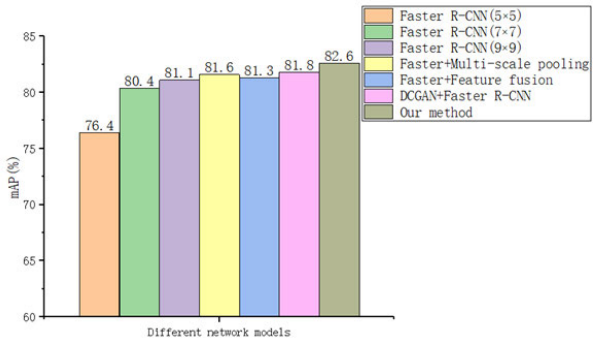


FIGURE 14. The overall detection results of different system models on the night data set.

in the high layer to achieve a similar compensation effect to deal with the non-robust information loss caused by the pooling operation. Table 2 shows the detailed experimental results of the fusion of any three levels of convolutional layers in VGG-16. The feature map output by Conv1_2 maintain the most complete details because it has not gone through the parameter reduction process in the pooling operation. At the same time, the feature map output by Conv5_3 condenses the adjustment features of multiple previous convolutional layers, thus containing the most abstract semantic information. As an intermediate feature between the two, Conv3_3 is necessary to be considered. The experimental details also prove the advanced nature of this fusion algorithm.

E. COMPARISON OF DIFFERENT ALGORITHMS

In order to verify the detection performance of the proposed DCGAN-Faster R-CNN model, other state-of-the-art detection methods and image enhancement strategies are used for evaluation and comparison—R-CNN, Fast R-CNN, Faster R-CNN, SSD, LIME, Retinex-Net, Robust-Retinex, Xiao *et al.*, Li *et al.* and Zhu *et al.*. Without bells and whistles, we use the traditional methods LIME, Retinex-Net, and Robust-Retinex to enhance the low-illumination night data set, and obtain three enhanced data results. For the fairness of the experiment, we still use Faster R-CNN as the basic network to perform detection tasks on those data results to prove the effectiveness of the algorithm. Table 3 shows the experimental results of different methods on the night data set. In Table 3, our method has an mAP of 82.6% compared with Retinex-Net, LIME and Robust-Retinex, which are improved by 1.1%, 0.7% and 1.2% respectively. Compared with other object detection algorithms based on deep learning, the model has more significant recognition advantages. It can be seen from Table 3 that for R-CNN, when we do not use DCGAN for feature space transformation and directly apply the detector, it will cause a 1.2% drop in mAP. For Fast R-CNN, this value is 1.7%, which means that DCGAN has successfully reconstructed the daytime scene by using night features, and has distinctive features that are more easily distinguished compared with night images. In addition, because we use the combination of the two modules to solve the problem of the actual night scene, the network parameters are increased. Interestingly, our method only increases the time consumption of 0.04s compared with using only the Faster R-CNN

TABLE 2. The detection results in the fusion of different convolutional layers in VGG-16. Convn_2 represents the feature map output by the second layer of the nth convolution module in the feature extraction network (for example, VGG-16). Convn_3 represents the third layer.

Network model	Data	mAP(%)	bic	boat	bus	car	cat	chair	dog	person	sofa	tv
Conv1_2+Conv 2_2+Conv3_3	night	78.1	80.7	71.9	79.1	80.5	82.6	78.4	79.6	81.2	80.9	66.8
Conv1_2+Conv 2_2+Conv4_3	night	78.9	81.5	70.3	82.2	81.4	83.8	79.6	80.1	82.7	79.5	68.2
Conv1_2+Conv 2_2+Conv5_3	night	80.7	82.4	74.7	84.3	82.4	83.7	82.5	80.6	81.6	82.6	71.9
Conv1_2+Conv 3_3+Conv4_3	night	80.3	81.8	74.5	83.1	82.5	83.3	81.7	81.2	82.4	82.7	70.6
Conv1_2+Conv 3_3+Conv5_3(Ours)	night	**81.3**	**82.5**	**76.4**	**84.7**	**82.9**	84.0	**83.1**	81.6	**82.9**	**83.4**	**71.9**
Conv1_2+Conv 4_3+Conv5_3	night	81.0	81.1	75.6	83.5	82.7	85.9	82.3	82.9	82.8	82.2	70.4
Conv2_2+Conv 3_3+Conv4_3	night	80.6	82.1	76.2	84.2	80.8	82.5	82.9	83.1	81.5	82.1	70.1
Conv2_2+Conv 3_3+Conv5_3	night	80.5	80.6	75.8	84.0	81.3	85.7	81.1	82.2	81.9	80.5	71.1
Conv2_2+Conv 4_3+Conv5_3	night	80.9	81.7	74.3	83.2	82.1	83.9	82.6	84.1	82.5	83.1	70.7
Conv3_3+Conv 4_3+Conv5_3 (C. Cao et al.)	night	81.1	82.0	75.9	84.1	81.6	84.2	81.4	84.3	82.8	83.0	71.6

TABLE 3. A comparison of different detection methods, where # indicates that the improved Faster R-CNN is used for object recognition after data enhancement.

Network model	Data	mAP(%)	bic	boat	bus	car	cat	chair	dog	person	sofa	tv	Runtime
R-CNN	night	44.3	45.2	39.1	46.6	45.9	48.0	44.7	44.5	45.6	47.8	35.9	-
R-CNN+DCGAN	night	45.5	47.7	40.3	48.2	46.8	49.1	45.6	46.9	47.0	46.8	36.4	-
Fast R-CNN	night	76.1	76.3	70.9	77.5	77.3	79.8	76.4	77.9	78.2	79.6	67.1	0.40s
Fast + DCGAN	night	77.8	79.0	73.6	78.3	78.6	81.9	78.2	79.1	80.8	80.1	68.4	-
Faster R-CNN	night	81.3	82.5	76.4	84.7	82.9	84.0	83.1	81.6	82.9	83.4	71.2	**0.27s**
Retinex-Net	night	81.5	83.7	78.2	80.3	82.5	86.7	82.8	84.0	83.2	81.6	72.4	0.81s
Retinex-Net #	night	82.1	83.3	77.6	82.4	85.8	84.2	84.7	82.8	85.5	84.1	70.9	-
LIME	night	81.9	83.0	78.6	81.2	84.1	85.6	83.4	85.3	82.7	81.5	73.6	1.30s
LIME#	night	82.2	84.6	77.1	85.6	83.3	83.7	85.9	83.4	83.6	82.8	71.8	-
Robust-Retinex	night	81.4	83.8	77.9	83.9	82.4	84.5	82.0	81.7	83.8	81.3	72.7	1.28s
Robust-Retinex#	night	81.7	84.1	76.8	84.3	83.5	85.1	82.4	81.6	84.2	82.7	71.5	-
Xiao et al.	night	81.0	83.9	77.5	84.2	82.6	84.2	81.6	82.3	83.8	82.5	70.4	0.33s
Li et al.	night	82.5	85.2	77.3	84.7	83.8	86.0	85.3	82.9	83.5	84.1	72.8	2.43s
Zhu et al.	night	81.8	84.5	77.8	83.3	82.7	85.9	82.2	82.1	83.6	84.7	71.2	0.69s
Our method	night	**82.6**	**85.6**	**78.0**	**85.1**	**84.0**	**86.2**	82.9	**83.1**	**84.4**	**84.6**	**73.1**	0.31s

detector (0.31s vs. 0.27s). This phenomenon is because in the testing phase we only applied part of the structure of the DCGAN model (i.e., the generator) and directly established the image mapping of the two scenes based on the mature weight parameters instead of the passive participation of the entire network. Compared with the current popular low-light detection algorithms, the algorithm in this paper has a certain level of accuracy improvement compared to the algorithms recently announced by Xiao *et al.*, Li *et al.* and Zhu *et al.*, and the speed is also in the priority. At the same time, our method has almost the same execution time as the method proposed by Xiao *et al.*, yet with a 1.6% increase in accuracy (82.6% vs 81.0%). Compared with the algorithm proposed by Zhu *et al.*, although with only a 0.1% increase in mAP value (82.6% vs 82.5%), a significant speed advantage is observed. In summary, we believe that the algorithm in this paper is more effective when dealing with recognition tasks in night environments. Different from traditional night image enhancement methods that are essentially based on the image itself (for example, feature, brightness, saturation and noise reduction), DCGAN applies spatial mapping on features to directly establishing the connection between night and daytime scenes. This method of reconstructing virtual features successfully breaks the limitation of previous work that can only enhance the image itself very limitedly. This is also the biggest difference between DCGAN and the existing technology. A series of experiments show that DCGAN has better enhancement advantages in image processing. We hope that the research in this paper can be helpful to future engineering development.

TABLE 4. The recognition results of Faster R-CNN with different backbones on the night data set. "Rn": ResNet-n, "V16": VGG-16, "Inv3": Inception-v3, "+DCGAN": The virtual image processed by the DCGAN algorithm is used as the input of Faster R-CNN containing the designated backbone.

Model	Data	mAP(%)	bic	boat	bus	car	cat	chair	dog	person	sofa	tv
ResNet-18	night	78.5	80.8	72.5	80.3	84.6	81.4	78.2	71.1	81.5	80.0	68.2
R18+DCGAN	night	79.7	82.2	73.9	82.3	84.9	82.7	79.2	78.9	81.4	81.6	69.5
VGG-16	night	80.4	82.8	74.1	83.9	83.3	83.1	80.0	80.8	82.8	82.7	70.7
V16 + DCGAN	night	81.8	83.4	76.6	84.6	83.3	85.9	82.0	82.5	83.1	83.8	72.9
ResNet-50	night	85.2	88.1	80.2	86.3	86.5	88.3	86.2	86.1	86.9	87.4	76.4
R50+DCGAN	night	86.1	88.9	79.3	89.5	88.1	88.7	87.9	87.8	86.5	88.2	75.7
ResNet-101	night	86.8	89.3	81.3	88.6	89.9	87.1	90.0	89.7	87.2	87.4	77.1
R101+DCGAN	night	87.9	90.2	84.2	89.4	91.3	88.6	89.8	90.6	86.5	89.3	78.7
Inception-v3	night	88.3	91.6	85.0	88.7	90.6	89.4	90.7	91.2	87.9	88.1	79.4
Inv3+DCGAN	night	89.5	92.7	86.2	91.4	91.2	89.6	92.9	92.3	89.6	89.0	79.5

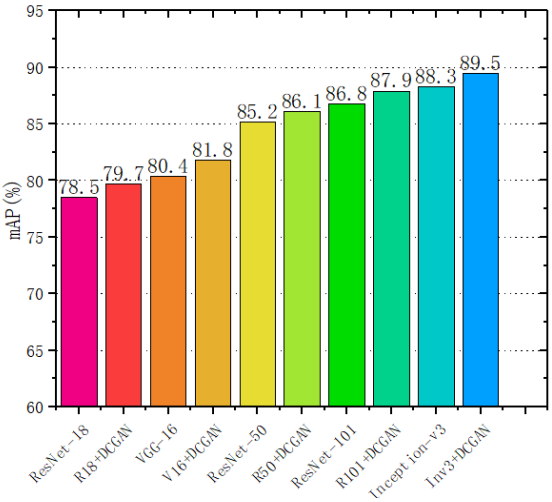


FIGURE 15. The experimental results of Faster R-CNN with different backbones on the night data set. Where Rn refers to ResNet-n, V16 represents VGG-16 and Inv3 represents Inception-v3.

F. RESEARCH ON DIFFERENT BACKBONES

We also presented research results on other frameworks to prove the acceptable flexibility of the core ideas of this paper. As is shown in Fig. 15, ResNet-18, ResNet-50, ResNet-101 and Inception-v3 are additionally used. The algorithmic combination of DCGAN and different backbones will improve the performance of the original network (see Fig.15). For ResNet-50, the DCGAN algorithm will show a 0.9% improvement in the mAP value (85.2% vs 86.1%), while for ResNet-101 this value is 1.1% (86.8% vs 87.9%). Further, the strategy of introducing DCGAN can also observe a 1.2% improvement effect under the Inception-v3 backbone with the highest mAP value. This shows that the model can find a feature map similar to the daytime scene for the input night image so as to improve the quality of the initial image. As part of the description, Table 4 shows the detection details of the algorithm for different types of objects. We hope that we can provide a positive reference for the work of future researchers.

G. MODEL TESTING OF ACTUAL TASKS

The test task is carried out on the night images taken in life. The overall effect of actual detection through the algorithm in this paper and the original Faster R-CNN is shown in Fig. 16. The renderings provided here are easy to observe. It can be seen that the positioning and recognition of targets in the night

FIGURE 16. Schematic diagram of the comparison of the detection effects of different methods. The first column of images is the recognition result of using the original Faster R-CNN directly, and the second column is the detection result using our method.

background is better and the detection accuracy is higher. Our model successfully mines the hidden object information in the dark environment to help us complete the task of night targets recognition in daily life, which is very practical.

V. CONCLUSION

Based on DCGAN and Faster R-CNN, we propose a detection algorithm to efficiently identify targets in night scenes, and evaluate the performance of our model in a real night environment. First, we use the specified nighttime data set and daytime data set as the input of the network, and use the powerful generation capability of the generator model to establish a unidirectional spatial distribution relationship. Next, we indirectly use the generated virtual samples as the training set of Faster R-CNN. In order to make the network capture the feature information of the image as much as possible, a multi-scale feature fusion strategy is used in the feature extraction network, which can be combined with more comprehensive details in the low convolutional layer with the rich semantic information in the higher-level convolutional layer. In addition, a multi-scale pooling strategy reduces the losses caused by traditional ROI Pooling operations. Experimental results show that our method has a significant accuracy improvement in classification and target positioning. At the same time, the algorithm ideas we proposed in this paper can also be flexibly extended to other fields, such as semantic segmentation, human pose evaluation, gesture recognition, etc. This work has a certain contribution to practical engineering applications, and we expect to develop more rigorous theoretical knowledge and contribute to deep learning research in future work.

REFERENCES

[1] U. Meis, W. Ritter, and H. Neumann, "Detection and classification of obstacles in night vision traffic scenes based on infrared imagery," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Shanghai, China, vol. 2, Oct. 2003, pp. 1140–1144, doi: 10.1109/ITSC.2003.1252663.

[2] Z. Yunchu, Z. Jianbin, and L. YiBin, "Moving object detection in the low illumination night scene," in *Proc. IET Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Shenzhen, China, 2012, pp. 1–4, doi: 10.1049/cp.2012.2296.

[3] P. Govardhan and U. C. Pati, "NIR image based pedestrian detection in night vision with cascade classification and validation," in *Proc. IEEE Int. Conf. Adv. Commun., Control Comput. Technol.*, Ramanathapuram, India, May 2014, pp. 1435–1438, doi: 10.1109/ICACCCT.2014.7019339.

[4] S. K. Sharma, R. Agrawal, S. Srivastava, and D. K. Singh, "Review of human detection techniques in night vision," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Chennai, India, Mar. 2017, pp. 2216–2220, doi: 10.1109/WiSPNET.2017.8300153.

[5] C. U. Bazan Caballero and Z. Zamudio Beltran, "Detection of traffic panels in night scenes using cascade object detector," in *Proc. Int. Conf. Mechatronics, Electron. Automot. Eng. (ICMEAE)*, Cuernavaca, Mexico, Nov. 2018, pp. 32–37, doi: 10.1109/ICMEAE.2018.00013.

[6] X. Guo, "LIME: A method for low-light IMage enhancement," in *Proc. 24th ACM Multimedia Conf. (MM)*, Amsterdam, The Netherlands, 2016, pp. 87–91.

[7] M. Li, J. Liu, W. Yang, X. Sun, and Z. Guo, "Structure-revealing low-light image enhancement via robust retinex model," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2828–2841, Jun. 2018.

[8] K. G. Lore, A. Akintayo, and S. Sarkar, "LLNet: A deep autoencoder approach to natural low-light image enhancement," *Pattern Recognit.*, vol. 61, pp. 650–662, Jan. 2017.

[9] L. Tao, C. Zhu, J. Song, T. Lu, H. Jia, and X. Xie, "Low-light image enhancement using CNN and bright channel prior," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, Sep. 2017, pp. 3215–3219.

[10] C. Wei, W. Wang, W. Yang, and J. Liu, "Deep retinex decomposition for low-light enhancement," in *Proc. Brit. Mach. Vis. Conf.*, Newcastle, U.K., 2018, pp. 1–12.

[11] Y. Xiao, A. Jiang, J. Ye, and M.-W. Wang, "Making of night vision: Object detection under low-illumination," *IEEE Access*, vol. 8, pp. 123075–123086, 2020, doi: 10.1109/ACCESS.2020.3007610.

[12] Z. Li, Z. Jia, J. Yang, and N. Kasabov, "Low illumination video image enhancement," *IEEE Photon. J.*, vol. 12, no. 4, pp. 1–13, Aug. 2020, doi: 10.1109/JPHOT.2020.3010966.

[13] Y. Zhu, Z. Jia, J. Yang, and N. K. Kasabov, "Change detection in multi-temporal monitoring images under low illumination," *IEEE Access*, vol. 8, pp. 126700–126712, 2020, doi: 10.1109/ACCESS.2020.3008262.

[14] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *J. Physiol.*, vol. 195, no. 1, pp. 215–243, Mar. 1968.

[15] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A neural network model for a mechanism of visual pattern recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 5, pp. 826–834, Sep. 1983.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," presented at the NIPS, Lake Tahoe, NV, USA, Dec. 2012.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[18] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[20] Y. Ren, C. Zhu, and S. Xiao, "Small object detection in optical remote sensing images via modified faster R-CNN," *Appl. Sci.*, vol. 8, no. 5, p. 813, May 2018.

[21] H. Jipeng, S. Yinghuan, and G. Yang, "Multi-scale faster-RCNN algorithm for small object detection," *Comput. Res. Develop.*, vol. 56, no 2, pp. 319–327, 2019.

[22] Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," *Proc Conf. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[23] C. Cao, B. Wang, W. Zhang, X. Zeng, X. Yan, Z. Feng, Y. Liu, and Z. Wu, "An improved faster R-CNN for small object detection," *IEEE Access*, vol. 7, pp. 106838–106846, 2019, doi: 10.1109/ACCESS.2019.2932731.

[24] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 3, Jun. 2014, pp. 2672–2680.

[25] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," presented at the ICLR, San Juan, PR, USA, May 2016.

[26] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," 2017, *arXiv:1704.00028*. [Online]. Available: http://arxiv.org/abs/1704.00028

[27] L. Yi-Lun, D. Xing-Yuan, L. Li, W. Xiao, and W. Fei-Yue, "The new frontier of AI research: Generative adversarial networks," *Acta Electronica Sinica*, vol. 2018, vol. 44, no. 5, pp. 775–792.

[28] W. Kun-Feng, Z. Wang-Meng, T. Ying, Q. Tao, L. Li, and W. Fei-Yue, "Generative adversarial networks: From generating data to creating intelligence," *Acta Electronica Sinica*, vol. 44, no. 5, pp. 769–774, 2018.

[29] A. Jaiswal, W. AbdAlmageed, Y. Wu, and P. Natarajan, "CapsuleGAN: Generative adversarial capsule network," 2018, *arXiv:1802.06167*. [Online]. Available: http://arxiv.org/abs/1802.06167

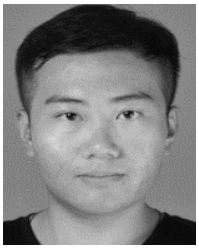
[30] W. Fang, F. Zhang, V. S. Sheng, and Y. Ding, "A method for improving CNN-based image recognition using DCGAN," *Comput., Mater. Continua*, vol. 57, no. 1, pp. 167–178, 2018.

[31] W. Fang, Y. Ding, F. Zhang, and J. Sheng, "Gesture recognition based on CNN and DCGAN for calculation and text output," *IEEE Access*, vol. 7, pp. 28230–28237, 2019, doi: 10.1109/ACCESS.2019.2901930.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," presented at the ICLR, San Diego, CA, USA, May 2015.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

KUN WANG received the bachelor's degree in automation, the master's degree in control theory and control engineering, and the Ph.D. degree in pattern recognition and intelligent systems from Northeastern University, in 2000, 2005, and 2008, respectively. She was a Visiting Scholar with Purdue University, USA, in 2011. She is currently an Associate Professor and a Master Tutor with the Civil Aviation University of China. Her research interests include pattern recognition and intelligent systems, and fault detection and analysis.



MAO ZHEN LIU was born in Jining, Shandong, China, in 1995. He received the bachelor's degree from Dezhou University. He is currently pursuing the master's degree with the Civil Aviation University of China. His research interests include pattern recognition and object detection.