CGC-NET: Aircraft Detection in Remote Sensing Images Based on Lightweight Convolutional Neural Network

Ting Wang, Xiaodong Zeng, Changqing Cao, Wei Li, Zhejun Feng, Jin Wu, Xu Yan, and Zengyan Wu

***Abstract*—In the past few years, aircraft detection in remote sensing (RS) images has been an important research hotspot, and it is very crucial in plenty of military applications. Based on the high computational cost of the model and numerous parameters, deep convolution neural networks-based algorithms have excellent performance in the aircraft detection task. However, it is still difficult to detect aircraft due to the complex background of RS images, various types of aircraft, and so on. In addition, it is difficult and costly to make labels for satellite-based optical RS images. Consequently, we propose an end-to-end lightweight aircraft detection framework called CGC-NET (a network based on circle grayscale characteristics), which can accurately detect aircraft with a few training samples. There are only a small number of trainable parameters in CGC-NET, which greatly reduces the need for large datasets. Extensive evaluations indicate the excellent performance of CGC-NET, in which the *F*-score can reach 91.06% and the model size is only 0.88 M. Therefore, CGC-NET can be used to accurately detect aircraft targets simply and effectively.**

***Index Terms*—Circle frequency filter (CFF), deep learning framework, few-shot learning, lightweight convolutional neural network (CNN), small samples.**

I. Introduction

With the rapid development of remote sensing (RS) technology and the ever-increasing of spatial resolution of RS images, the number of RS images has exploded. Object detection is of increasing interest to researchers as it can provide valuable information for a high-level semantic understanding of RS images. Aircraft detection in RS images is one of the main tasks of object detection, and it is widely used in various fields. It is helpful for airline supervision, scheduling airport flights, etc. Especially in the military field, it is of great importance to the estimation of the battlefield situation and the formulation of military decisions.

The past few decades have witnessed the development of aircraft detection algorithms. In the light of the feature extractor used, aircraft detection algorithms in RS images can be divided into traditional methods and CNN-based methods [1]. Traditional methods include Viola–Jones detector [2], Histogram of Gradients (HOG) [3] + support vector machines (SVM) [4], etc. These methods usually use a sliding window to select the region of interest (ROI). Extract hand-crafted features from the ROI, and then use machine learning algorithms to classify them.

CNN-based methods use CNN to extract deep features, which can capture more local and global information. Deep CNN has made significant progress in aircraft detection with its powerful learning capabilities [5]–[8].

The poor robustness of hand-crafted features has led to traditional methods being less capable of detecting aircraft in complex scenes. The CNN-based method also faces two challenges in aircraft detection. First, the performance of CNN-based methods will be limited, in the case of relatively small samples. Second, most of the available CNN-based models are resource-hungry, and the detection performance is improved by increasing the width, depth, and resolution of the network. Therefore, a lightweight network based on few-shot learning [9] should be proposed to address the above problems.

Cai and Su propose an unsupervised circle frequency filter (CFF) algorithm [10] to detect aircraft. The CFF algorithm first calculates the amplitude value of each pixel according to (2) and then sets a fixed threshold to filter out the pixels below the threshold. Finally, the candidate pixels are clustered to get the detection result. The algorithm can obtain good aircraft detection results without any training samples.

When the CFF algorithm is used to detect aircraft on our dataset, two problems emerged: many false alarms and inaccurate localization. Therefore, we combine the powerful feature extraction capabilities of CNN-based methods with the simplicity and ability of traditional methods to detect targets with small samples to solve these two problems. As a result, a lightweight aircraft detection algorithm named CGC-NET is provided. The proposed method uses amplitude filtering to select candidate pixels of interest. CGC-NET has a powerful feature extraction and representation capability, which can remove a lot of false alarms and improve the localization accuracy of the center of

Manuscript received December 11, 2021; revised February 25, 2022; accepted March 13, 2022. Date of publication March 16, 2022; date of current version April 19, 2022. This work was supported in part by the National Natural Science Foundation of Shaanxi Province under Grant 2020JM-206, in part by the National Defense Basic Research Foundation under Grant 61428060201, and in part by the 111 Project under Grant B17035. *(Corresponding author: Changqing Cao.)*

Ting Wang, Xiaodong Zeng, Changqing Cao, Zhejun Feng, Jin Wu, Xu Yan, and Zengyan Wu are with the Xidian University, Xi'an 710071, China (e-mail: tingw2016hg@126.com; xdzeng@xidian.edu.cn; chqcao@mail.xidian.edu.cn; zhjfeng@mail.xidian.edu.cn; jinw9824@163.com; yanxu_xy@163.com; zengyanw0@163.com).

Wei Li is with the Shenzhen Aerospace New Power Technology Company, Ltd., Shenzhen 518057, China (e-mail: fzjh513@126.com).

Digital Object Identifier 10.1109/JSTARS.2022.3159981

the aircraft. Hence, the proposed method can produce excellent detection results.

Evaluation results of CGC-NET show that the *F*-score can reach 91.06% and the model size is only 0.88 M. It can be seen that CGC-NET obtains better results than CornerNet and Faster R-CNN with a small training set.

The crucial contributions of this article are presented as follows.

1) We develop a lightweight end-to-end framework to extract the gray value features on the circumferential, which increases the robustness of the features. The experimental results demonstrate the effectiveness of this framework.
2) CGC-NET only uses CNN to obtain a part of the features of the input image. Thus, it considerably reduces trainable parameters and improves the inference speed while almost not losing detection accuracy.
3) CGC-NET only needs a small number of samples to train, which greatly reduces the manual annotation of the dataset.
4) The proposed method is pixel-level, which is capable of locating the center of the aircraft with high accuracy, especially for small targets.

The rest of this article is organized as follows. Section II outlines some works on aircraft detection. The theoretical background of CGC-NET is illustrated in Section III. Section IV presents the details of CGC-NET framework and illustrates its implementation in Section V. Finally, Section VI concludes this articles.

II. Related Works

Aircraft detection has been extensively studied for decades. This section will briefly review some works associated with object detection and aircraft detection. Most of these object detection methods can also be used to detect aircraft objects only.

Based on the development process of the object detection algorithms, they can be classified into traditional algorithms and deep CNN-based algorithms. Most traditional object detection algorithms use hand-crafted rules to extract features [11], while deep CNN-based algorithms adopt CNN.

A. Traditional Object Detection Algorithms

Early object detection methods are mainly based on template matching [12], [13], which can only detect objects with relatively simple spatial location relationships.

The subsequent object detection methods are principally based on the geometric representation and the statistical classification of appearance features, such as NN [14], SVM [4], Adaboost [2].

Thereafter, a large number of local feature descriptors for object detection have emerged, such as Haar-like features [15], LBP [16], SURF [17], SIFT [18], and HOG [3]. Subsequently, a great deal of effort is devoted to exploring approaches to group descriptors into higher level representations in object detection, such as spatial pyramid matching [19], Bag of Words [20], and Fisher Vector [21].

Deformable part-based model [22] is the pinnacle of the traditional object detection and the championship of the Pascal VOC 2007-2009 challenge [23]. This method is a component-based detection method, which has strong robustness to the deformation of the object and has become the core part of many machine vision algorithms.

Although traditional methods have achieved good detection results, they are often more complicated in design and unable to extract high-level deep features of the image, which limits the accuracy and speed of detection.

B. Deep CNN-Based Object Detection Algorithms

Compared to traditional methods, deep CNN-based algorithms have powerful feature representation capabilities [24], to reach state-of-the-art detection accuracy.

In 2012, AlexNet [25] demonstrated tremendous success in image classification. Subsequently, researchers used deep CNNs for object detection, yielding the classical R-CNN [26] model. This method generates a large number of region proposals on the input image. SVM classifies the features extracted by the CNN so that the class to which the feature belongs can be determined. Boundary regression is used to obtain the exact region where the object is located. R-CNN [26] outperforms traditional methods significantly, creating an era of object detection.

YOLO [27] eliminates the step of generating candidate regions and performs regression and classification directly on the original image. Although it sacrifices a certain accuracy, it greatly speeds up the inference process.

The two methods above belong to two major branches of object detection: two-stage and one-stage methods. The biggest difference is whether the candidate regions are generated or not. The representation of the two-stage method is R-CNN [26] series, involving Fast R-CNN [28], Faster R-CNN [29], Mask R-CNN [30], R-FCN [31], Cascade R-CNN [32], etc. The one-stage method is represented by the YOLO [27] series (YOLO v2 [33], YOLOv3 [34], YOLOv4 [35], etc.) and SSD [36] series (SSD [36], DSSD [37], etc.), consisting of FCOS [38], CornerNet [39], CenterNet [40], RefineDet [41], ExtremeNet [42], EfficientDet [43], etc.

Faster R-CNN [29] uses a Region Proposal Network to generate region proposals, which enormously saves inference time. It is an end-to-end network that can meet real-time requirements. At the moment, Faster R-CNN [29] is still an essential branch of object detection methods.

YOLOv3 [34] has some incremental improvements on YOLO [27] and YOLO 9000 [33], and its processing speed has been considerably improved. It takes only 22 ms to process an image [34]. For small objects, YOLOv3 [34] integrates multiscale information, resulting in superior detection accuracy.

RefineDet [41] consists of two interconnected modules that imitate the structural design of the two-stage detection model, but it belongs to the one-stage method. It incorporates the advantages of Faster R-CNN [29] and SSD [36] while achieving higher accuracy and speed.

CornerNet [39] is a new one-stage object detection method. It converts object detection into the detection of a pair of key points. The backbone used for feature representation is followed by two prediction modules, one for predicting the top-left corner and the other for the bottom-right corner. Grouping all the prediction points yields the bounding boxes of all the objects. This method greatly simplifies the output of the network and eliminates the anchor boxes.

EfficientNet [44] uses a composite scale expansion method to simultaneously scale all backbone, feature networks, and box/class prediction networks with the uniform resolution, depth, and width. The backbone of EfficientDet [43] is EfficientNet [44] with the addition of a simple and effective weighted bidirectional feature pyramid network as its neck. On several datasets, EfficientDet emerges as the state-of-the-art approach with fewer parameters and higher inference speed.

He *et al.* [45] propose a lightweight network (DABNet) to perform cloud detection. The network has only 4.12 M parameters and 8.29 G multiadds. The detection accuracy of DABNet [45] is high, its detection boundary is extremely clear and the false alarm rate is quite low. It proposes a deformable contextual feature pyramid module that can improve the multiscale feature representation capability. DABNet [45] achieves state-of-the-art performance in cloud detection.

Most of the above object detection frameworks can be used for aircraft detection and achieve excellent detection results. However, most of these networks have a huge number of parameters and require large labeled datasets to be trained to obtain decent detection performance.

C. Aircraft Detection Algorithm in RS Image

Aircraft detection from RS images with complicated backgrounds is a challenging task. In the field of object detection, deep CNN has achieved a remarkable breakthrough. Therefore, in recent years, researchers have focused on deep CNN-based aircraft detection [5], [46], [47].

OE-FCN [48] is an end-to-end network for addressing the intraclass variance of aircraft in RS images. It brings in an online exemplar mining mechanism in the CNN and adopts exemplars to characterize the distinct intraclass features of the aircraft. As a result, accurate and fast aircraft detection is achieved.

Concerning very high-resolution RS images, Zhang *et al.* [49] propose a weakly supervised aircraft detection algorithm. The model can simultaneously extract proposals and locate aircraft with only an image-level training dataset. This weakly supervised framework can alleviate the cost of human annotation without degrading detection accuracy.

X-LineNet is a novel model [50] that has been used to detect aircraft in RS images. It transforms aircraft detection into the detection and clustering of a pair of intersecting lines, so that richer information can be learned. Among the one-stage aircraft detection algorithms, X-LineNet is state-of-the-art and has considerable detection accuracy with advanced two-stage detectors.

III. Theoretical Analysis

In general, the aircraft can be distinguished based on the grayscale, texture, shape, and pattern of the RS images. Most aircraft are similar in shape, consisting of a fuselage and two wings. The connecting part of the aircraft fuselage and wingspan can be approximated as a rectangle, which is shown in Fig. 1(a). It is assumed that the centroid of the rectangle in Fig. 1(a) is the center of the aircraft. It is then considered to be the center of a certain circle shown in Fig. 1(b). The diameter of this circle is greater than the length and width of the rectangle while being smaller than the length of the fuselage and wingspan of the aircraft. Seventy-two pixels are uniformly extracted clockwise from this circle, and its gray value curve is shown in Fig. 1(c). These gray values exhibit a particular characteristic of 4–5 peaks and 4—5 troughs [10], named the circle grayscale (CG) characteristic. The number of peaks and troughs depends on the position and direction of the aircraft. A circle does not have the CG characteristic if its center is not on the center of the aircraft. Accordingly, aircraft can be detected from complex and cluttered backgrounds depending on whether the circle has this unique characteristic.

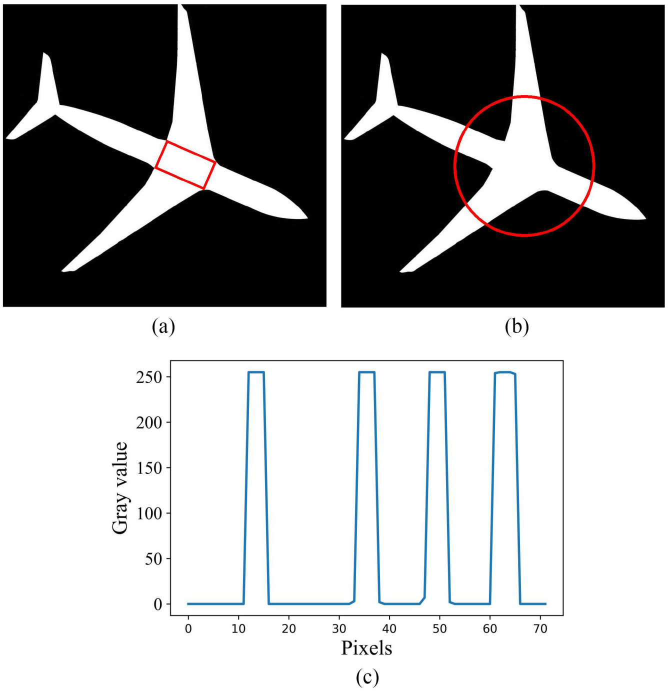


Fig. 1. (a) Intersection of the aircraft fuselage and the wingspan is approximated as a rectangle. (b) Circle on the aircraft, the size of the original image is 1024 × 958. The center of the circle is the center of the rectangle in (a), and the radius of the circle is 220 pixels. (c) Gray value curve of 72 pixels on the circle.

Fourier principle shows that: any sequence or signal of continuous measurement can be expressed as an infinite superposition of sine wave signals of different frequencies. Sine has a unique property—fidelity, which is not shared by the original signal. In other words, if a sine signal is an input, the output is also a sine signal. During this process, only the amplitude and phase may change, the frequency and waveform remain the same. Hence, we substitute the original signal with sine and cosine signals, which is helpful for the computer to process the original signal more simply.

It is supposed that one-dimension array p_k ($k = 0, 1, 2, \ldots, N - 1$) is the gray value of the pixels uniformly extracted clockwise on the circle centered at (i, j), and R is the radius of this circle. The discrete Fourier transform (DFT) of the original signal p_k is as follows:

$$P = \sum_{k=0}^{N-1} p_k e^{-j\frac{2\pi}{N}kn} (k = 0, 1, 2 \ldots N-1). \tag{1}$$

With the help of Euler's formula, the DFT formula in exponential form is converted to the Cartesian coordinate system. The Fourier transformed array is squared to obtain the amplitude value. The larger the amplitude value, the more likely it is that the circle area is an aircraft

$$\text{Amplitude} = \left(\sum_{k=0}^{N-1} p_k \cos\frac{2\pi}{N}kn\right)^2 + \left(\sum_{k=0}^{N-1} p_k \sin\frac{2\pi}{N}kn\right)^2. \tag{2}$$

Here, n is the period of the sine and cosine functions in (2).

In RS images, the viewpoint is usually from the sky down to the ground [51]. Therefore, the aircraft is lying on the image, which makes them rich and diverse in detection. This will bring some difficulty to aircraft detection. However, CGC-NET has rotation invariance [52]. As shown in Fig. 2, the circle centered on the center of the aircraft still has the CG characteristic after the image is rotated by different angles. CGC-NET detects the aircraft based on the CG characteristic and therefore it has rotation invariance [52]. Thus, the complexity of the proposed method is greatly decreased, and the number of samples to be learned is reduced.

According to the statistical results, the original signal has about four periods. Therefore, based on the Fourier transform principle, set n in (1) and (2) to 4. A circle centered on the center of the aircraft has a large amplitude, while a circle not centered on the aircraft has a small amplitude. As the amplitude is the square of the result acquired from the DFT, the amplitudes for circles of the aircraft centers and nonaircraft centers vary significantly. Then, we can set a threshold in accordance with this property to filter out a portion of the input data, which is called amplitude filtering. Therefore, amplitude filtering will greatly reduce the amount of input data for the following networks.

IV. Framework Implementation

The flowchart of CGC-NET is depicted in Fig. 3. CGC-NET consists of three portions: amplitude filtering, detection network training, and nonmaximum suppression (NMS) [53]. The detailed implementation of our method is illustrated in Algorithm 1.

The first part is amplitude filtering. First, a black border is added to the input image to ensure that the center of the aircraft on the edge of the image can be fetched. Then, each pixel in the image is used as the center of a circle with the radii of 3, 10, 17, and 35 (obtained by statistics when generating the data). Ninety points are taken uniformly on each of the four circles, and their grayscale values form a two-dimensional (2-D) array with a size of 4 × 90. Before feeding the input data into CGC-NET, the amplitude sum of the array is calculated according to (2) and normalized to between 0 and 255. Setting the threshold to 80 so that pixels with amplitude less than the threshold can be filtered out, thus allowing a significant reduction in the amount of data. This threshold is based on extensive experiments and ensures that no object is lost in the amplitude filtering phase. The amplitude filtering is very fast to compute, thus there is no necessity to design a network specifically to extract ROIs or proposals, which saves time in the inference process. The remaining data is normalized by subtracting the mean and dividing it by the standard deviation. Normalization of the raw data can raise the convergence speed and the performance of the model. At the same time, it can prevent gradient explosion.

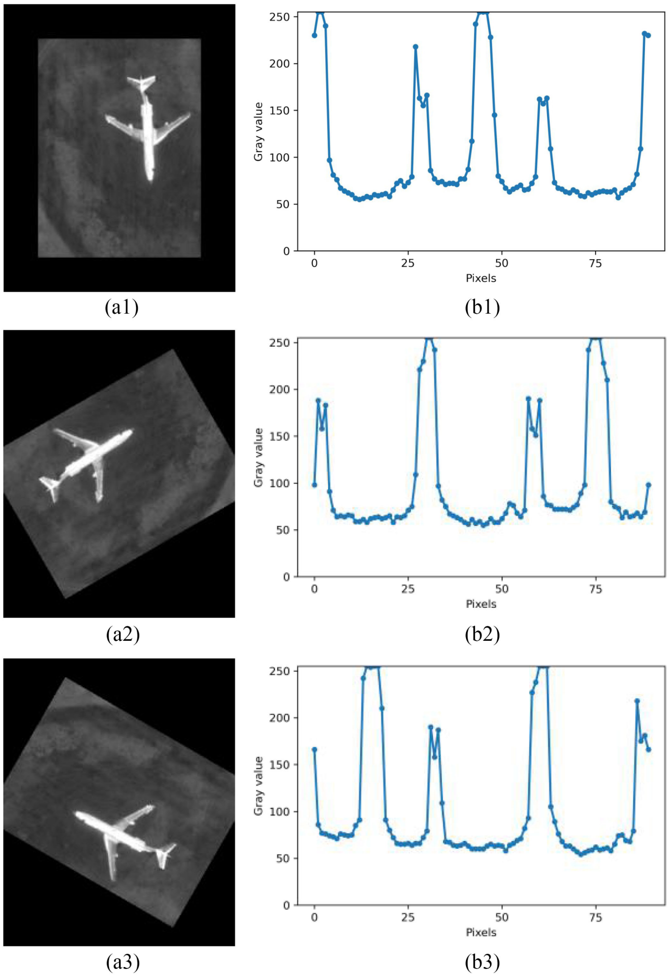

(a1) (b1)
(a2) (b2)
(a3) (b3)

Fig. 2. Rotation invariance of CGC-NET. (a1) Aircraft image with black borders. (a2), (a3) are the images of (a1) after rotating counterclockwise by 120° and 240°, respectively. (b) Take 90 points on a circle with the center of the aircraft as the center, and the gray value changes regularly.

The second part is the detection framework. An end-to-end lightweight model called CGC-NET is proposed for aircraft detection. It just consists of three convolutional layers, two pooling layers, one dropout layer, and one dense layer.

The first convolutional layer defines a feature detector so that the network can only learn a feature. Since the length of the input data is 90, we set the kernel size to 45 to extract as many CG characteristics as possible. The learning capacity of 1 filter is very limited, so we define 75 filters. This results in 46

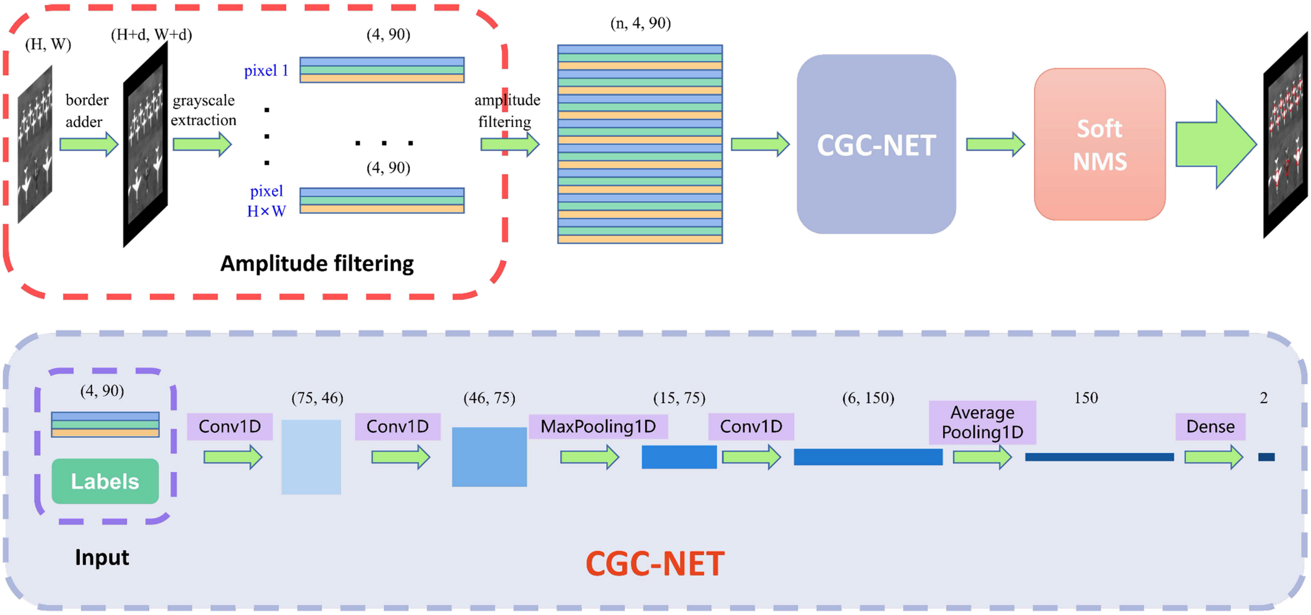


Fig. 3. Block diagram of the proposed method. The layers "Conv1D" represent one-dimensional convolution.

Algorithm 1: CGC-NET.

```
Require: A network CGC-NET, a labeled aircraft
  detection dataset Tr and test data Te.
Step 1: Amplitude filtering
1. Add a black border to each input image.
2. Take each pixel on the original image as the center and
   take the circles with radii of 3, 10, 17 and 35.
3. Evenly take the gray values of 90 points on each circle
   to form a $4 \times 90$ array.
4. Amplitude filtering: According to Eq. (2), the
   amplitude is calculated for each array. Set an
   appropriate threshold to filter out data whose
   amplitude is lower than the threshold.
Step 2: Train the detection network
Repeat
5. Randomly select a batch from the remaining data after
   amplitude filtering.
6. Optimize CGC-NET and update the network
   parameters following Eq. (3).
Until convergence
Step 3: Process the detection results
7. Obtain the detection result on Te utilizing the trained
   network CGC-NET.
8. NMS is used to filter out redundant bounding boxes
   with lower scores.
9. Mark the final result on the original image with circles.
   The radius of the circle is the one with the highest
   amplitude among the four selected radii.
```

diverse features. The output is a 75×46 matrix with each column containing a filter weight, and each filter contains 75 weights.

The output of the first layer is fed into the second layer. Similar to the first layer, 75 filters with a kernel size of 30 are defined on the second layer for training. The output is a matrix with a size of 46×75.

To deprecate redundant information and simplify the network, a MaxPooling layer of size 3 is added. As a consequence, the amount of output data is decreased by two-thirds.

To learn higher level features, a 1-D CNN layer is chosen. Its output matrix size is 6×150. To further apply dimensionality reduction to the input data, an AveragePooling layer is deployed. The output is a vector of length 150, which means that each feature detector has one single weight.

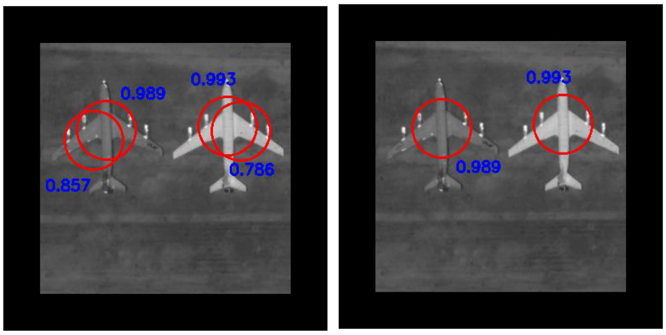
The Dropout layer removes the neurons in the hidden layer at random so that the fully connected network is sparsified to some extent. This renders the network less sensitive to subtle variations in the data, thereby boosting the detection accuracy of unknown data. The output is still a vector of length 150.

The dense layer is activated by the Softmax function, which will minimize the length of the vector to 2. The proposed method converts aircraft detection into a binary classification problem, so two categories, "aircraft" and "nonaircraft," need to be predicted. Hence, the output represents the probability of each of the two classes happening.

CGC-NET uses Adam optimizer [54], which can be regarded as a fusion of RMSprop [55] and stochastic gradient descent [56] with momentum. It is a binary classifier, so binary cross-entropy is employed as the loss to train the network. The loss is as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log\left(p\left(y_i\right)\right) + \left(1 - y_i\right) \cdot \log\left(1 - p\left(y_i\right)\right) \tag{3}$$

where y is the label (1 for aircraft, 0 for nonaircraft) and $p(y_i)$ is the predicted probability that the ith input data is aircraft.



(a) (b)

Fig. 4. (a) Multiple boxes on each target after direct classification and regression. (b) Result of (a) after the Soft-NMS algorithm suppresses the redundant frame.

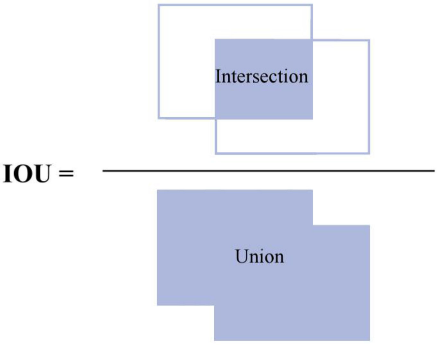
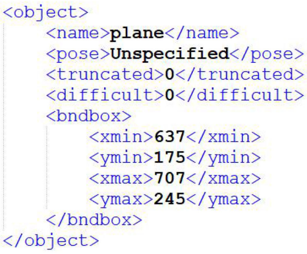


Fig. 5. Intersection over union.



```
<object>
    <name>plane</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
        <xmin>637</xmin>
        <ymin>175</ymin>
        <xmax>707</xmax>
        <ymax>245</ymax>
    </bndbox>
</object>
```

Fig. 6. Label file of the image that we collected.

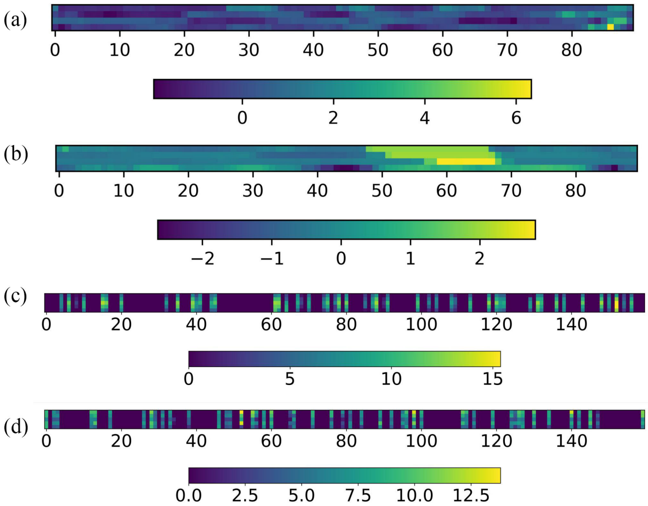


Fig. 7. (a) Grayscale values of the pixels on the circumference of the positive sample. (b) Grayscale values of the pixels on the circumference of the negative sample. (c) Feature map of (a) output by conv1d_3. (d) Feature map of (b) output by conv1d_3.

The third part is NMS. As shown in Fig. 4(a), multiple boxes are obtained on the same object after CGC-NET. In order to keep only one optimal box on the same target, the Soft-NMS algorithm [57] removes some redundant boxes during the test period. The result of filtering out redundant data through the NMS algorithm is exhibited in Fig. 4(b).

The process of Soft-NMS is as follows. First, the bounding boxes with the highest scores are chosen. Subsequently, T is taken as the IOU (intersection over union) threshold, and the boxes with IOU greater than or equal to T are eliminated. Here, the predicted circle is converted into its bounding rectangle to calculate its IOU. The process of NMS ensures that the bounding box with the maximum score is left so that the center of the aircraft can be precisely located. As shown in Fig. 5, IOU calculation is used to measure the overlap between two proposals. Finally, the above processes are repeated among the unprocessed boxes. In our experiments, T is set to 0.

The confidence score is as follows:

$$S_i = \begin{cases} S_i, \mathrm{IOU}\left(M, B_i\right) < T \\ S_i\left(1 - \mathrm{IOU}\left(M, B_i\right)\right), \mathrm{IOU}\left(M, B_i\right) \geq T \end{cases} \tag{4}$$

where B_i is the ith bounding box, S_i is the score of B_i, and M is the box with the maximum score.

After the above process, mark the selected circle with the highest score on the test image. Comparing the amplitude values of the input data of 4 radii, the radius corresponding to the maximum amplitude value is the radius of the circle. Thus, the aircraft detection results are obtained.

V. Experiments

A. Experimental Data and Evaluation

To evaluate CGC-NET, we have made a corresponding dataset, which contains 419 RS images. There are a total of 2132 aircraft on the images. To further evaluate the generalization ability of CGC-NET, RSOD [58] and UCAS_AOD [59] datasets are used to test our method. RSOD dataset includes 4993 aircraft, while UCAS_AOD dataset contains 7482 aircraft. The image size in these two datasets is different. But there is no need to scale the image to the same size. Because CGC-NET is applicable to images of any size, and the image is not directly fed to the model. The gray values on the circle are extracted pixel by pixel to form a 2-D array with a size of 4 × 90, as shown in Fig. 7(a) and (b). The size of the 2-D array is fixed and the 2-D array can be directly fed to the model.

The aircraft is manually marked as (x, y, r) in our training set, where (x, y) are the coordinates of the center of the circle and r is its radius. Since most object detection algorithms use the groundtruth box to represent the target. Therefore, in order

TABLE I
SUMMARY OF CGC-NET

Model: "CGC-NET"		
Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 75, 46)	13575
conv1d_2 (Conv1D)	(None, 46, 75)	103575
max_polling1d_1 (MaxPooling1D)	(None, 15, 75)	0
conv1d_3 (Conv1D)	(None, 6, 150)	112650
global_average_polling1d_1 (AveragePooling1D)	(None, 150)	0
Dropout_1 (Dropout)	(None, 150)	0
dense_1 (Dense)	(None, 2)	302

TABLE II
COMPARISON OF DETECTION PERFORMANCE OF DIFFERENT AIRCRAFT DETECTION ALGORITHMS

Evaluation indicator	**CGC-NET**	**Faster R CNN**	**CornerNet squeeze**	**EfficientDet D7**	**CFF**
Precision	91.35%	76.08%	76.54%	88.63%	73.07%
Accuracy	83.59%	66.50%	64.63%	78.05%	60.77%
Recall	90.77%	84.07%	80.6%	86.74%	78.32%
Missing Alarm	9.23%	15.93%	19.4%	13.26%	21.68%
False Alarm	8.65%	23.92%	23.46%	11.37%	26.93%
F-score	91.06%	79.88%	78.52%	87.67%	75.60%
Model size	**0.88 M**	**547 M**	**128 M**	**208 M**	**-**
Pretrain on	**-**	Imagenet	COCO	COCO	**-**

to compare the performance of CGC-NET with other methods, the collected images are also manually marked as (xmin, ymin, xmax, ymax) like RSOD and UCAS_AOD datasets. As shown in Fig. 6, (xmin, ymin) is the coordinates of the top left corner, and (xmax, ymax) is the coordinates of the bottom right corner.

Our self-made dataset is divided into two parts, about 40% is the training set and 60% is the test set. The training set has 160 images containing 765 aircraft. Manual annotation of large datasets is usually costly and unreliable. However, CGC-NET only needs a small number of training samples to obtain pleasant aircraft detection results, which just compensates for this shortcoming.

To quantitatively measure the performance of CGC-NET, *Precision*, *Accuracy*, *Recall*, and *F-score* are employed as metrics. The model size and inference time are used to evaluate the efficiency of the proposed method, and generally the smaller the better.

B. Implementation Details

The proposed model is implemented in Keras 2.3.1. The operating system is Ubuntu 18.04. The hardware platform is Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60GHz and NVIDIA Corporation GP102GL [Quadro P6000] for accelerating model training. The preset training parameters are as follows: the learning rate is 0.001, and the training epoch is 20.

The summary of CGC-NET is listed in Table I. The summary contains the layers and their order in the model, the output shape of each layer, and the parameters of the model.

Some intermediate visual results are given in Fig. 7 to know if CGC-NET can learn CG characteristics. (a) and (b) are the grayscale values of the pixels on the circumference of the positive and negative samples, respectively. Both positive and negative samples are arrays of $4*90$. Positive samples have at least one of the four rows of the array with CG characteristics. While the negative sample is haphazard and does not have CG characteristics. (c) and (d) are the (a) and (b) feature maps output from Conv1d_3, respectively. It can be seen that the feature maps of the positive and negative samples are easier to distinguish than the original images.

C. Comparison With State-of-the-Art and Classic Methods

1) Quantitative Analysis: To compare the performance of CGC-NET with the classic method Faster R-CNN [29], it is retrained on our training set. Its backbone is VGG16 that pretrained on the Imagenet [60] dataset, which is a natural scene dataset containing 1.2 million images. We finetune the Faster R-CNN with the training set. The maximum number of iterations is 2000 to ensure that there is no overfitting.

CornerNet [39] is a one-stage detector with high detection accuracy. It does not need to generate anchor boxes, which greatly reduces the computational complexity. To use CornerNet-squeeze [61] on our dataset, the model pretrained with 500k iterations on the COCO dataset [62] is trained on our training set.

Table II shows the performance of CGC-NET and other state-of-the-art methods on our dataset. CGC-NET has a significant advantage over other models in detection performance. Without pretraining on any other dataset, our method can yield a 91.06% F-score, which is much higher than Faster R-CNN, CornerNet-squeeze, EfficientDet, and CFF. Meanwhile, the number of parameters of the proposed method is about 1/622 of Faster R-CNN, about 1/145 of CornerNet-squeeze, and about 1/236 of EfficientDet. The existing aircraft detection methods usually employ multiple convolutional layers in the backbone for feature representation of the whole image, so the computational cost is generally large and the accuracy is high. Our method uses only three 1-D convolutional layers to extract partial features of the image, which can keep the computational cost very small without degrading the detection accuracy. This ensures that the overall structure of CGC-NET is lightweight.

The UCAS-AOD [59] is divided into two parts, where 800 images are considered as the training set and the other 200 as the test set. With each aircraft as the crop center, each image is cropped into three different sizes. In this way, the training set contains more than 15000 images. CornerNet and X-LineNet are trained on this training set, and the results on the test set are shown in Table III. CGC-NET is trained only on our small self-made training set which contains 765 aircraft and tested

TABLE III
COMPARISON OF THE AVERAGE PRECISION OF DIFFERENT AIRCRAFT DETECTION METHODS ON THE UCAS-AOD DATASET

Methods	Average Precision	Test on	Train on	Model Size
CornerNet 104-Hourglass	76.2%	200 images	15000 images	200.97M
X-LineNet 104-Hourglass	92.4%	200 images	15000 images	745M
X-LineNet ResNet-101	91.3%	200 images	15000 images	207M
X-LineNet DLA-34	91.3%	200 images	15000 images	98M
CGC-NET	91.17%	the entire dataset (1000 images)	160 images in our dataset	0.88M

TABLE IV
COMPARISON OF THE INFERENCE TIME (S) OF DIFFERENT AIRCRAFT DETECTION METHODS

Image size Methods	305×296	1280×659	1044×915	Pretrain on
CFF	0.21	1.78	2.05	-
Faster R CNN	0.28	2.53	2.93	Imagenet dataset
CornerNet-squeeze	0.24	2.31	2.65	COCO dataset
CGC-NET	0.27	2.31	2.62	-

on the entire UCAS-AOD dataset [59] to obtain the results in Table III.

As shown in Table III, the average precision of CGC-NET is higher than CornerNet [39] and lower than X-LineNet [50]. However, the average precision of the proposed method is the detection result on the whole UCAS-AOD dataset. As a result, CFF-NET has a competitive average precision compared to X-LineNet. The size of X-LineNet [50] is much larger than CGC-NET. The model size of X-LineNet [50] is 745 M for backbone 104-hourglass, 207 M for backbone ResNet-101, and 98 M for backbone DLA-34.

The inference time can be used to evaluate the complexity of the model. The smaller the parameters and calculations, the more efficient the models are and the shorter the inference time [45]. In Table IV, the inference times of these methods are also evaluated. The inference times are calculated for images of sizes 305 × 296, 1280 × 659, and 1044 × 915, respectively. It can be found that for small images, the inference time of CGC-NET is much lower than that of Faster R-CNN but larger than that of CornerNet-squeeze and CFF. However, for large images, the inference time of CGC-NET is comparable to or even smaller than that of CornerNet-squeeze. This may be due to the fact that most of the area in an RS image is the background and the area of the aircraft is only a tiny part. Most of the data are filtered out in the preprocessing stage of the proposed method, so the inference time decreases.

Among these methods, CGC-NET has the smallest number of parameters and calculations. Since the input data are preprocessed in the proposed method, the inference time is a little large. In the subsequent study, we will optimize the preprocessing part of the proposed method and the structure of CGC-NET to diminish the inference time and enhance the detection accuracy.

2) Qualitative Analysis: Fig. 8 illustrates a comparison of the detection results of the various methods on our dataset. The detection results of some representative images are selected, containing images with weak targets, small targets, multiple targets, and complex and cluttered backgrounds. The comparison reveals that the detection performance of the proposed method is better and relatively more targets are correctly detected. The proposed method and CornerNet-squeeze are comparable in the ability to accurately locate targets, and both are better than Faster R-CNN. There may be two reasons why the detection performance of CornerNet-squeeze is lower than the proposed method. First, CornerNet-squeeze is pretrained on the COCO dataset [62], but COCO [62] is a natural scene dataset and does not contain RS images. Second, our RS image training set contains a small number of images and only one category, which is not enough to exploit the powerful learning ability of CornerNet-squeeze and may cause overfitting. Meanwhile, compared with Faster R-CNN and CornerNet-squeeze, CGC-NET has a lightweight structure with only 2.3 M parameters. CGC-NET is only trained on our self-made dataset, but the detection accuracy on UCAS_AOD [62] and RSOD [58] is still high. Therefore, CGC-NET has good generalization ability.

In summary, CGC-NET has a strong detection performance.

D. Experimental Results and Discussion

There are 13842 aircraft in the test set, of which 13753 aircraft can be detected with the proposed method. Among them, 12564 aircraft can be successfully detected, indicated as TP. There are 1189 nonaircraft targets marked as aircraft by CGC-NET and 1278 aircraft unmarked. As a consequence, FP equals 1189 and TN equals 0.

Table II is a quantitative comparison of different detection methods. For the four methods, the greater the *Precision*, *Accuracy*, *Recall*, and *F-score*, the smaller the *Missing Alarm* and *False Alarm*, the better the detection results. The *F-score* of CGC-NET is much larger than CFF, Faster R-CNN, and CornerNet-squeeze, which shows that CGC-NET has good detection performance and noise suppression ability.

The results in Table II show that CGC-NET achieves better performance than CFF, Faster R-CNN, and CornerNet-squeeze. More specifically, the *F-score* of CGC-NET is 15.46% higher than CFF, 11.18% higher than Faster R-CNN, 12.54% higher than CornerNet-squeeze, and 3.39% higher than EfficientDet. The model size of Faster R-CNN is 547.254 M [63], which is approximately 622 times of CGC-NET. The model size of CornerNet-squeeze is 128 M [64], which is approximately 145 times of CGC-NET.

The proposed method still has good detection performance in some complex scenarios, but there are still some noticeable misclassifications, especially in regions that also have obvious CG

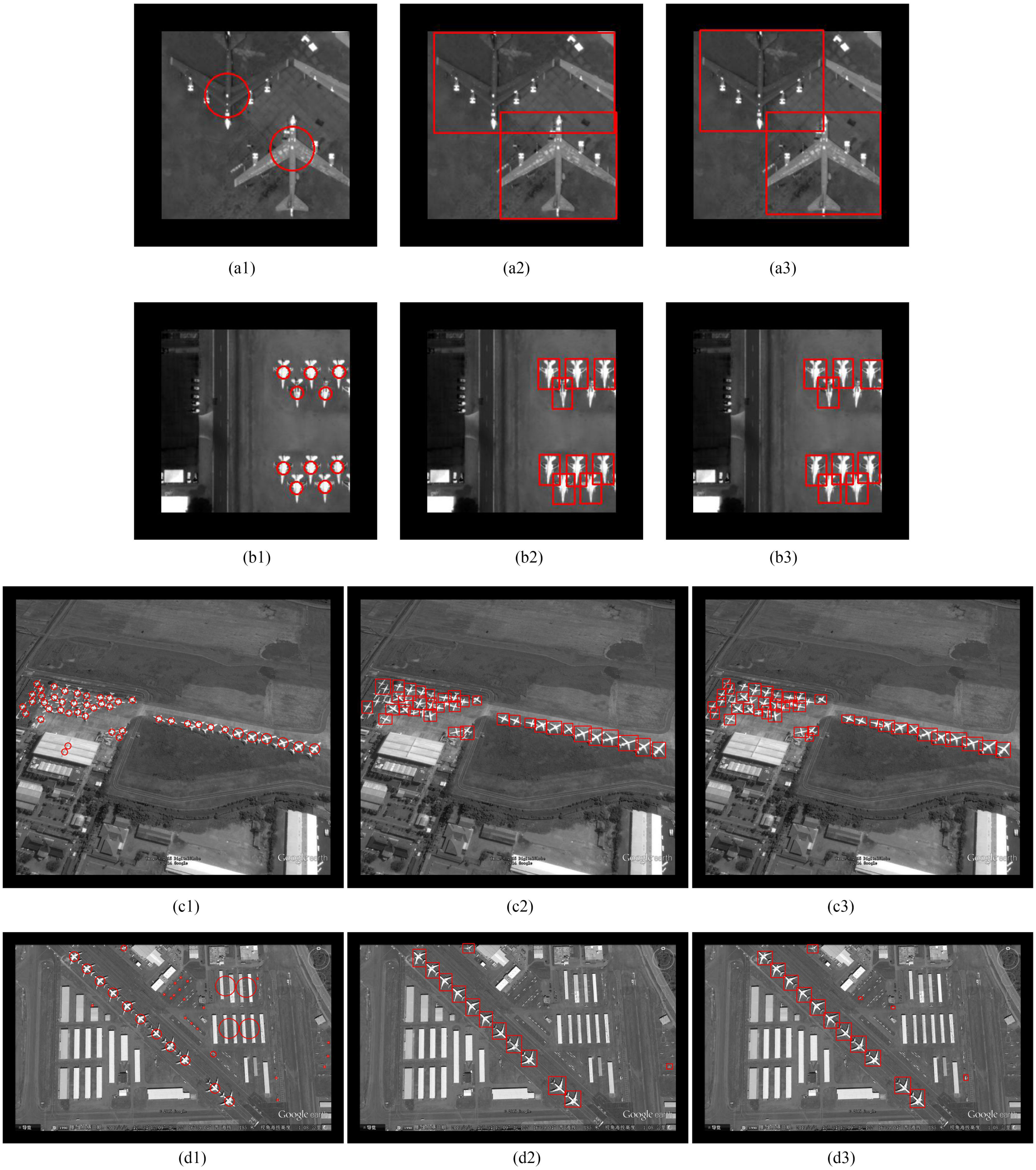


Fig. 8. (a1)–(d1) are the detection result of CGC-NET. (a2)–(d2) are the detection result of Faster R-CNN. (a3)–(d3) are the detection result of CornerNet.

characteristics. Subsequently, we will optimize the algorithm in three aspects of preprocessing, model structure, and postprocessing to further enhance the performance of the proposed method.

VI. Conclusion

An end-to-end lightweight CNN framework is proposed in this article for detecting aircraft targets in RS images with a few samples. During the training process, only a few parameters need to be learned. Theoretical proof and experiments have been provided to show that CGC-NET is effective. Experimental results show that the *F-score* of CGC-NET is 11.18% higher than that of Faster R-CNN and 12.54% higher than CornerNet-squeeze. There are 13842 aircraft in the test set, and 12564 aircraft can be correctly detected with CGC-NET, accounting for 90.77% of the test set. At the same time, the model size of CGC-NET is only 1/622 of Faster R-CNN and 1/145 of CornerNet-squeeze.

References

[1] J. Jiang, M. Chen, and J. A. Fan, “Deep neural networks for the evaluation and design of photonic devices,” *Nature Rev. Mater.*, vol. 6, no. 8. pp. 679–700, 2021.

[2] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Apr. 2001, pp. 511–518.

[3] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.

[4] E. Osuna, R. Freund, and F. Girosi, “Training support vector machines: An application to face detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 130–136.

[5] H. Wu, H. Zhang, J. Zhang, and F. Xu, “Fast aircraft detection in satellite images based on convolutional neural networks,” in *Proc. Int. Conf. Image Process.*, Sep. 2015, pp. 4210–4214.

[6] Q. Wu *et al.*, “Improved mask R-CNN for aircraft detection in remote sensing images,” *Sensors*, vol. 21, no. 8, pp. 1–13, 2021.

[7] S. Bouarfa, A. Doğru, R. Arizar, R. Aydoğan, and J. Serafico, “Towards automated aircraft maintenance inspection. A use case of detecting aircraft dents using mask r-cnn,” in *Proc. AIAA Scitech Forum Expo.*, Jan. 2020, pp. 1–19.

[8] Y. Zhang, K. Fu, H. Sun, X. Sun, X. W. Zheng, and H. Wang, “A multi-model ensemble method based on convolutional neural networks for aircraft detection in large remote sensing images,” *Remote Sens. Lett.*, vol. 9, no. 1, pp. 11–20, 2018.

[9] X. Sun, B. Wang, Z. Wang, H. Li, H. Li, and K. Fu, “Research progress on few-shot learning for remote sensing image interpretation,” *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.*, vol. 14, pp. 2387–2402, 2021.

[10] H. Cai and Y. Su, “Airplane detection in remote sensing image with a circle-frequency filter,” in *Proc. Int. Conf. Space Inf. Technol.*, Nov. 2005, Art. no. 59852T.

[11] G. Cheng and J. Han, “A survey on object detection in optical remote sensing images,” *ISPRS J. Photogramm. Remote Sens.*, vol. 117, pp. 11–28, 2016.

[12] Y. Ma and B. Kong, “A study of object detection based on fuzzy support vector machine and template matching,” in *Proc. World Congr. Intell. Control Automat.*, Jun. 2004, pp. 4137–4140.

[13] D. T. Nguyen, W. Li, and P. Ogunbona, “An improved template matching method for object detection,” in *Proc. Asian Conf. Comput. Vis.*, Aug. 2010, pp. 193–202.

[14] H. A. Rowley, S. Baluja, and T. Kanade, “Neural network-based face detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23–38, Jan. 1998.

[15] R. Lienhart and J. Maydt, “An extended set of Haar-like features for rapid object detection,” in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2002, pp. 1–4.

[16] J. Trefný and J. Matas, “Extended set of local binary patterns for rapid object detection,” *Comput. Vis. Winter Work.*, pp. 1–7, Feb. 2010.

[17] J. Li and Y. Zhang, “Learning SURF cascade for fast and accurate object detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3468–3475.

[18] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 1150–1157.

[19] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2169–2178.

[20] J. Sivic and A. Zisserman, “Video google: A text retrieval approach to object matching in videos,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1470–1477.

[21] F. Perronnin, J. Sánchez, and T. Mensink, “Improving the fisher kernel for large-scale image classification,” in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 143–156.

[22] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[24] G. Cheng, P. Zhou, and J. Han, “Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.

[28] R. Girshick, “Fast R-CNN,” *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[29] 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.

[30] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[31] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2016, pp. 379–387.

[32] Z. Cai and N. Vasconcelos, “Cascade R-CNN: High quality object detection and instance segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1483–1498, May 2021.

[33] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6517–6525.

[34] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[35] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, “YOLOv4,” in *Proc. CVPR Work. Future Datasets Vis.*, Jun. 2020, pp. 1–17.

[36] W. Liu *et al.*, “SSD: Single shot multibox detector,” in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 21–37.

[37] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, “DSSD: Deconvolutional single shot detector,” 2017, *arXiv:1701.06659*. [Online]. Available: http://arxiv.org/abs/1701.06659

[38] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 9626–9635.

[39] H. Law and J. Deng, “CornerNet: Detecting objects as paired keypoints,” *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 642–656, 2020.

[40] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “CenterNet: Keypoint triplets for object detection,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 6568–6577.

[41] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, “Single-Shot refinement neural network for object detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4203–4212.

[42] X. Zhou, J. Zhuo, and P. Krahenbuhl, “Bottom-up object detection by grouping extreme and center points,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 850–859.

[43] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and efficient object detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 10778–10787.

[44] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. 36th Int. Conf. Mach. Learn.,* Jun. 2019, pp. 10691–10700.

[45] Q. He, X. Sun, Z. Yan, and K. Fu, “DABNet: Deformable contextual and boundary-weighted network for cloud detection in remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Dec. 2021, Art. no. 5601216, doi: 10.1109/TGRS.2020.3045474.

[46] Y. Li, K. Fu, H. Sun, and X. Sun, “An aircraft detection framework based on reinforcement learning and convolutional neural networks in remote sensing images,” *Remote Sens,* vol. 10, no. 2, pp. 1–19, 2018.

[47] Q. Liu, X. Xiang, Y. Wang, Z. Luo, and F. Fang, “Aircraft detection in remote sensing image based on corner clustering and deep learning,” *Eng. Appl. Artif. Intell.*, vol. 87, pp. 1–11, 2020.

[48] B. Cai, Z. Jiang, H. Zhang, Y. Yao, and S. Nie, “Online exemplar-based fully convolutional network for aircraft detection in remote sensing images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 7, pp. 1095–1099, Jul. 2018.

[49] F. Zhang, B. Du, L. Zhang, and M. Xu, “Weakly supervised learning based on coupled convolutional neural networks for aircraft detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5553–5563, Sep. 2016.

[50] H. Wei, Y. Zhang, B. Wang, Y. Yang, H. Li, and H. Wang, “X-LineNet: Detecting aircraft in remote sensing images by a pair of intersecting line segments,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1645–1659, Feb. 2021.

[51] Y. Zhang, Y. Yuan, Y. Feng, and X. Lu, “Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5535–5548, Aug. 2019.

[52] A. Sedaghat, M. Mokhtarzade, and H. Ebadi, “Uniform robust scale-invariant feature matching for optical remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4516–4527, Nov. 2011.

[53] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. Int. Conf. Pattern Recognit.*, Jun. 2006, pp. 850–855.

[54] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 2015, *arXiv:1412.6980*, [Online]. Available: http://arxiv.org/abs/1412.6980

[55] F. Zou, L. Shen, Z. Jie, W. Zhang, and W. Liu, "A sufficient condition for convergences of adam and rmsprop," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 11119–11127.

[56] L. Bottou, "Stochastic gradient descent tricks," in *Proc. Neural Netw., Tricks Trade*, 2012, pp. 421–436.

[57] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS - Improving Object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5562–5570.

[58] Y. Long, Y. Gong, Z. Xiao, and Q. Liu, "Accurate object localization in remote sensing images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2486–2498, May 2017.

[59] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao, "Orientation robust object detection in aerial images using deep convolutional neural network," in *Proc. Int. Conf. Image Process.*, Dec. 2015, pp. 3735–3739.

[60] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* Jun. 2009, pp. 248–255.

[61] H. Law, Y. Teng, O. Russakovsky, and J. Deng, "CornerNet-Lite: Efficient keypoint based object detection," 2020, *arXiv:1904.08900*. [Online]. Available: http://arxiv.org/abs/1904.08900

[62] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 740–755.

[63] N. Wang, A. Cai, and S. Zhang, "The study of RNN enhanced convolutional neural network for fast object detection based on the spatial context multi-fusion features," in *Proc. 11th Int. Symp. Comput. Intell. Des.*, Dec. 2018, pp. 136–140.

[64] H. Yang, B. Fan, and L. Guo, "Anchor-free object detection with mask attention," *Eurasip J. Image Video Process.*, vol. 2020, no. 1, pp. 1–17, 2020.

Ting Wang received the undergraduate degree in optical engineering from Xi'an Technological University, Xi'an, China, in 2015. She is currently working toward the Ph.D. degree with Xidian University, Xi'an, China.

Her research interests include weak and small object detection and background subtraction.

Xiaodong Zeng received the graduate degree in optical heterodyne detection technology from Xidian University, Xi'an, China, in 1996.

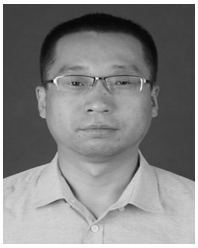
Now, he is a Professor in Xidian University. His research focuses on optoelectronic technology and application.

Changqing Cao received the Ph.D. degree in photoelectric detection technology from Xidian University, Xi'an, China, in 2010.

In 2011, he was an Associate Professor with Xidian University. His research focuses on laser technology and applications.

Wei Li was born in January 1981. He received the master's degree in electromagnetic fields and microwave technology from the School of Electronic Information, Northwestern Polytechnical University, Xi'an, China, in 2006.

He is currently with Shenzhen Aerospace New Power Technology Ltd., Shenzhen, China.



Zhejun Feng received the graduate degree in semiconductor laser from Xidian University, Xi'an, China, in 2008.

His research interests include photoelectric detection and signal processing.

Jin Wu received the undergraduate degree from Xi'an Technological University, Xi'an, China, in 2019. She is currently working toward the master's degree with Xidian University, Xi'an, China.

Her research focuses on object detection.

Xu Yan received the undergraduate degree in optical engineering from Xidian University, Xi'an, China, in 2017, where he is currently working toward the Ph.D. degree.

His main research interests include photoelectric detection and image processing.

Zengyan Wu received the undergraduate degree in optical engineering from the North University of China, Taiyuan, China, in 2018. She is currently working toward the Ph.D. degree in optical engineering with Xidian University, Xi'an, China.

Her research mainly focuses on solving the decoherence effect based on array detector.