Vehicle Detection in Aerial Images Based on Hyper Feature Map in Deep Convolutional Network

Jiaquan Shen, Ningzhong Liu*, Han Sun, Xiaoli Tao, Qiangyi Li
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics,
MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Jiangsu Nanjing 210016, China
[e-mail: lnz_nuaa@163.com]
*Corresponding author: Ningzhong Liu

Received September 3, 2018; revised October 17, 2018; accepted November 1, 2018; published April 30, 2019

Abstract

Vehicle detection based on aerial images is an interesting and challenging research topic. Most of the traditional vehicle detection methods are based on the sliding window search algorithm, but these methods are not sufficient for the extraction of object features, and accompanied with heavy computational costs. Recent studies have shown that convolutional neural network algorithm has made a significant progress in computer vision, especially Faster R-CNN. However, this algorithm mainly detects objects in natural scenes, it is not suitable for detecting small object in aerial view. In this paper, an accurate and effective vehicle detection algorithm based on Faster R-CNN is proposed. Our method fuse a hyperactive feature map network with Eltwise model and Concat model, which is more conducive to the extraction of small object features. Moreover, setting suitable anchor boxes based on the size of the object is used in our model, which also effectively improves the performance of the detection. We evaluate the detection performance of our method on the Munich dataset and our collected dataset, with improvements in accuracy and effectivity compared with other methods. Our model achieves 82.2% in recall rate and 90.2% accuracy rate on Munich dataset, which has increased by 2.5 and 1.3 percentage points respectively over the state-of-the-art methods.

Keywords: Car detection, deep convolutional network, hyper feature map, small object detection, feature fusion

1. Introduction

The invention of cars has brought great convenience to our work and life, but with the increase of the number of cars, some problems have been brought, such as traffic accidents, vehicle congestion, vehicle chaos and so on.

At present, Surveillance cameras are installed on the key nodes of the city, such as parking lots and high-speed roads, it used to take the electronic photos. However, this method cannot intuitively show the traffic conditions of the entire road. With the higher demand for intelligent transportation, the detection technology based on aerial vehicles has attracted the attention of many scholars in recent years [1–8]. The unmanned aerial vehicle (UAV) has the advantages of small volume, flexible maneuverability and convenient portability. The UAV aerial photography can monitor the road condition in an all-round and multi-scale way, which has great advantages in vehicle detection [9–13].

However, vehicle detection based on aerial image still faces many challenges and difficulties. The main reasons are,

1. Aerial images are generally very large but the detected vehicle object is very small. It is indeed difficult to detect so many small vehicle objects in a large range.

2. Vehicles have a variety of styles and colors, and vehicle objects are often accompanied with complex background information such as occlusion, shadows, which brings many difficulties to the detection.

In previous studies, many vehicle detection algorithms based on aerial images have been proposed, and the effectiveness of the algorithm has been improved in the past few years. Most of these algorithms are based on a sliding-window method which apply the filter to all possible locations and scales in the image. Then a classifier such as SVM or AdaBoost classifier was trained with the obtained features, which used to predict vehicle in each window [14–17]. In [2], the authors present a method that can detect the vehicles without an accurate scale information. The study employe a fast binary detector using integral channel features in a soft cascade structure, and then a multi-class classifier obtained the orientation and type of the vehicles. This method presents a competitive result in terms of rapidity and effectiveness. However, there are still some drawbacks in this method. Firstly, in terms of feature extraction, the hand-crafted features or shallow-learning based features restrict the ability to extract and represent features. Secondly, there are lots of redundant computations based on sliding window method, which would significantly increases the computational burden.

In recent years, deep learning has made a significant progress in object detection. The deep learning model has been continuously enriched and its detection performance has also been greatly improved. There are some typical models worked so well in object detection including R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD and so on [18–22]. The Faster R-CNN algorithm performs much better than the traditional sliding window methods, which achieves state-of-the-art detection performance. However, it is still facing many problems and challenges to directly apply these algorithms to vehicle detection in aerial images. The main reason is that vehicle detection in aerial images is more difficult than in natural scenes.

In this study, we propose an accurate and effective vehicle detection framework in aerial images (see **Fig. 1**). This model creates a hyper feature map (HFM) by fusing Concat layer and Eltwise layer. The Concat layer is used for learning the weights of the fusion of the object in formation and contextual information, which can reduce the interference of useless background noises. The Eltwise layer uses equivalent weights set manually and fuses the

multi-level features, which can enhance the effectiveness of useful context [23]. Our model contains rich detail features and shallow layer information and then fuses these features, which will be more suitable for the detection of small objects. Moreover, we set an appropriate scale and ratio of the anchor box according the size of object vehicle in the aerial image, which will further improve the effectiveness of the detection.

At the same time, the limited annotation data would easily leads to the model under-fitting, and the large-scale aerial images would reduce the detection speed. In order to overcome two problems mentioned above, we used the methods in [4] and [5] to preprocess the data, segment the aerial image into blocks, and increase the amount of data through rotation. Compared with [2], our method significantly improves the accuracy and efficiency of detection. Compared to [4] and [5], our model is more accurate and effective for feature extraction, and the effectiveness of detection has been further improved. Compared with Faster R-CNN, our method and model are more suitable for small target detection. Moreover, in order to verify the effectiveness of our method, we trained and tested it on the public dataset (Munich dataset) and our collected dataset. The main contribution of our work is: **(1)** we established an accurate and effective feature extraction and classification method for small targets through HFM network, **(2)** setting appropriate Anchor box according to the vehicle size, which improved the effectiveness of vehicle detection in the aerial image. **(3)** Moreover, we established our aerial vehicle image dataset with the ground truth and verified the effectiveness of our method on this dataset.

This paper is organized as follows: Section 2 discusses related works. The proposed method is detailed in Section 3. Section 4 reports the experimental results. Finally, Section 5 concludes the paper.

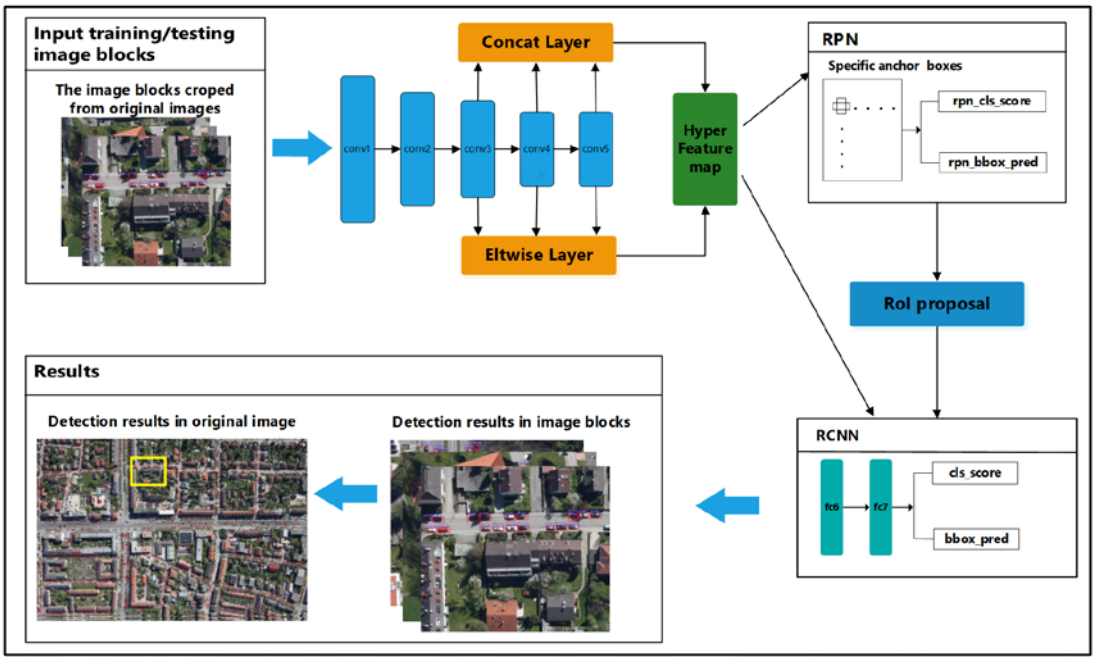


Fig. 1. The framework of our method in vehicle detection

2. Related Work

2.1 Object Detection Based on Traditional Handcrafted Features

The traditional object detection is divided into two parts: feature extraction and classification. At present, Haar wavelet, LBP, SIFT and HOG are the typical handcrafted features in object detection.

The Haar wavelet feature is proposed by Papageorgiou, which is first used for face detection. Paul proposes an integral image method for this feature calculation and adopted the AdaBoost algorithm to improve the accuracy of object detection. This research is successfully applied to face detection [24,25]. The LBP local binary pattern is used to extract the texture features of the image [26], and it has the characteristics of rotation invariance and gray invariance which can extract texture features of images. SIFT scale invariant feature is a local feature description operator [27], which can detect the key feature points in the image, and is robust to the transformation of light, noise as well as angle of view in the image. The HOG gradient histogram algorithm uses sliding window method to filter all possible positions and scales in the image [28]. The detectors determine whether there is a target and the category of the target according to the area passed by the sliding window. The DPM algorithm calculates the gradient histogram of HOG [29], which uses SVM to perform object matching and classification. As an important object detection algorithm, DPM has advantages in posture analysis and object location.

However, most of the feature extraction used in traditional object detection algorithms are manually designed, and the performance of the algorithm is unstable, which is mainly based on the understanding of the specific task by the designers, and it has less actual parameters. At the same time, the effectiveness of these object detection models is poor, which can only detect a specific task well, and do not have a good generalization performance. Although traditional object detection shows good detection performance for specific task, its disadvantages are also obvious, it cannot meet the requirements of large-scale data at the current stage.

For image classification, Bushra el al. [30] proposed a novel image representation that incorporates the spatial information to the inverted index of Bag-of-Visual-Words (BoVW) model, which outperformed the existing state-of-the-art in terms of classification accuracy. For image retrieval, Nouman el al. [31] presented an image representation method based on the histograms of triangles. This method added spatial information to the inverted index of bag of features representation, which enhanced the performance of image retrieval. However, the addition of spatial information in these methods inevitably increased the burden of calculation. In [32], the authors presented a novel visual words integration of Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF), which can be safely recommended as a preferable method for image retrieval tasks. However, these hand-crafted features are not good enough at separating the object from the background in complex environments. In object detection, Juang et al. [33] and Chen et al. [34] used hand-crafted features with a support vector machine (SVM) for candidate region classification. However, AdaBoost gradually replaced SVM due to its good performance. Recently, region CNN-based detection methods have achieved great success in object detection, owing to their powerful feature representation. The most popular is region-based convolutional neural networks (R-CNN) and their improved methods Fast R-CNN and Faster R-CNN. All of them achieve state-of-the-art performance.

2.2 Deep CNNs for Objection Detection

In recent years, deep learning has played a significant advantage in object detection. The deep

learning detection model was constantly improved, and its detection performance has also been greatly improved. There were typical models including R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD and so on.

The R-CNN algorithm used a selective search algorithm to generate object-like regions [18], and then extracted deep features for classification by SVM. Although the efficiency of the algorithm has been improved, there were a large number of repeated operations, and the efficiency was not high enough. The Fast R-CNN algorithm [19] has an advantage compared with the R-CNN. In this algorithm, a ROI pooling layer structure was designed to solve a large number of repetitive operations in R-CNN, so that the performance of the algorithm was greatly improved. However, the algorithm still needed selective search to generate positive and negative samples, which still restricted the efficiency of the algorithm. The Faster R-CNN algorithm generated a RPN (Region Proposal Networks) auxiliary network to determine whether there was an object in the candidate box [20], which determined the object type by the multitask loss of the classified location. In Faster R-CNN, the convolution neural network of the whole process can share the feature information, and the computational efficiency has been greatly improved. The YOLO(You Only Look Once) algorithm was designed based on the idea of regression [21], and the speed of the algorithm has been improved. The YOLO algorithm was based on the global information of the image. The algorithm was trained by the feature of the convolution neural network, it directly predicted the frame coordinates in each grid and the confidence of each class. However, this algorithm has some problems such as incorrect location and low recall rate. The SSD algorithm combined YOLO regression ideas with the anchor box mechanism to predict the region of the object on the feature maps of different convolutional layers [22], which output discrete multi-scale default coordinates. The research used local feature of different scales for regression on the entire image, which can maintain the rapidity of the algorithm and ensure the accuracy of the frame positioning. However, because the algorithm used multi-level feature classification, the characteristics of small objects were not obvious, which has difficulty in detecting small objects.

At present, the Faster R-CNN achieves the most advanced performance in object detection. This algorithm designs the RPN network that assists in sample generation and divides the algorithm structure into two parts. The algorithm first determines whether the candidate frame is object by the RPN network, and then predicts the object type by the multi-tasking loss of the classification and location. The entire network process can share the feature information of the convolutional neural network, which will save the cost of computing and avoid the algorithm accuracy rate decreasing. But directly using this algorithm for small object detection, especially in aerial images, it still faces many challenges. More details are as follows:

1. In the field of vehicle detection in aerial image, the number of objects are much greater than in natural scene.

2. The size of vehicle in the aerial image is much smaller than that in the natural scene. At the same time, the background of the aerial image is more complex than the natural scene, thus these increase the difficulty on the location and detection of the object.

3. The size of the aerial image is much larger than that in the natural scene, and the label data of the vehicle is very limited. All these have brought many difficulties and challenges to the vehicle object detection in the aerial image.

Considering the above problems and challenges, we believe that the following two reasons lead to poor performance of Faster R-CNN.

1. The RPN in Faster R-CNN is not suitable for detection of small objects, because the RPN only combines a relatively rough feature graph, and the small object detection often needs to integrate richer feature maps, especially the feature information of shallow layers.

2. Faster R-CNN shows good effect on object detection in natural scene, but in aerial image, the scale and size of object is much smaller than that in natural scene, so the size and scale of anchor box designed in Faster R-CNN is not suitable for small object detection.

2.3 Vehicle Detection in Aerial Image

Deep learning takes a huge advantage in object detection, and it becomes an important detection method in the field of object detection. However, the aerial image has its own characteristics, such as the object is small, and easily affected by the shadow, the background is complex and so on. Therefore, the deep learning model mentioned above cannot be directly used for vehicle detection in aerial images, it requires targeted improvement.

In recent years, deep learning model has its advantages in many fields [35-36], and its latest achievements in aerial image target detection are also noticeable [37–42]. Nassim proposed to segment the aerial image into similar regions, and determined the candidate regions of the vehicle, then located and classified the targets according to the convolutional neural network and SVM classifier [43]. This method can improve the detection speed by segmenting candidate regions, but it was easily affected by the shaded region and the recall rate of detection was not high. The RICNN algorithm was proposed to perform object detection on aerial image [44]. This study trained the rotationally invariant layer and then performed special fine-tuning of the entire RICNN network to further improve the detection performance. However, this algorithm also obviously increased the network overhead. In reference [5], the authors performed vehicle detection in aerial images by adding negative sample marks to the dataset and establishing an HRPN network. The HRPN was a feature fusion on different network layers, which improves the detection accuracy. However, the algorithm only combined the feature of part of the shallow layer. At the same time, the algorithm was easily affected by the image resolution, and the effectiveness of the algorithm was poor.

Currently, Faster R-CNN achieves state-of-the-art detection performance in the field of object detection. For this reason, in this paper, we make full use of the superior performance of Faster R-CNN in object detection, which establishing a model for the feature extraction of small objects, and set special anchor boxes according to the characteristics of small objects. We validated our model on the public dataset (Munich dataset) and our collected datasets, which show that our method can effectively improve the performance of the detection.

3. Model and Method

The method based on the convolutional neural network has high requirements for GPU memory in the image processing field, especially for processing large-size images. At the same time, the number of aerial image is less, which can easily lead to under-fitting. For this reason, we augment the dataset according to [4] and [5]. In the training stage, we divide the original large-size image into image blocks, and rotate the blocks with four angles (i.e., 45°, 135°, 225°, and 315°) that expanded the number of samples by fourfold. In the testing stage, the tested results are obtained based on the trained network, and the detection results of image blocks are merged into the original image.

3.1 Hyper Feature Map Network

There are two typical RPN network structures, ZF model [45] and VGG model [46]. The ZF model is a relatively lightweight model, which the parameters are few and the depth of the network is limited. The VGG model shows that the network performance can be improved by increasing the number of network layers. This model shows superior performance. Therefore,

in this study, our hyper feature map network is based on the VGG16+Faster R-CNN network model. The network of RPN in Faster R-CNN is passed by the conv5, different from Faster R-CNN, in our model we combine the last three convolution layer with Concat layer and Eltwise layer, and then this two layers extract the features together which transmit to hyper feature map. The network structure is shown in **Fig. 2**.

3.1.1 Concat layer and Eltwise layer

The function of the Concat layer is to splice two or more feature maps on the channel or number dimensions. For example, if you splice conv_1 and conv_2 on the number dimension, the number of dimension can be different, and the rest of the dimensions (channel, H, W) must be consistent. The operation at this time is the number k_1 of conv_1 plus the number k_2 of conv_2, and the blob output of the Concat layer can be expressed as:

$$blob_{Concat} = (k_1 + k_2) \times C \times H \times W \tag{1}$$

The Concat layer performs feature fusion on feature maps of different number of dimensions. This operation can increase the representation range of feature maps and increase the amount of information of the feature map.

There are three operations in the Eltwise layer: product (point multiplication), sum (addition and subtraction), and max (maximum value), where sum is the default operation. If you want to implement the eltwise sum operation of conv_1 and conv_2, you can add the corresponding elements together. Unlike the operation of the concat layer, the operation of the eltwise layer requires the shape of the feature maps to be identical, and the blob output of the eltwise layer can be expressed as:

$$blob_{Eltwise} = k \times C \times H \times W \tag{2}$$

In formula (2), $k = k_1 = k_2$. The eltwise layer combines two or more layers into one layer, which increases the saliency and effectiveness of the feature.

The differences between Concat layer and Eltwise layer are as follow. First of all, the form of operation is different, the Concat layer splicing two or more feature maps in the channel or number dimension, but the Eltwise layer is to operate on the corresponding elements on the feature map. Secondly, the shape of the feature maps are different, the Concat layer does not require the shape of the feature map to be consistent in operation (for example, the dimensions of channel or number can be inconsistent), but the shape of the feature maps must be consistent on the Eltwise layer. Finally, the function is different, the Concat layer can get the target architecture information through the fusion of learning weights, which can reduce the influence of background noise on detection performance. The Eltwise layer uses equivalent weights set manually and fuses the multi-level features, which can improve the utilization of contextual information [23].

In addition, the related research shows that deeper convolutional layers can get higher recall, and lower convolutional layers can get more accurate localization [47]. Therefore, we combine the concatenation module and eltwise module from shallow features and deep features, which can further enhance the effectiveness of the object detection.

3.1.2 Overall Architecture

The overall structure of our model is shown in **Fig. 2**. The first convolutional layer (conv1_1) takes the training images as input and performs a convolution operation with number output equal 64, pad equal 1, kernel size equal 3. The conv1_2 executes the same operation as conv1_1 with number output equal 64, pad equal 1, kernel equal 3. Then the conv1_2 is used for ReLu operation to export relu1_2, then the relu1_2 performs MAX pooling operation with kernel size equal 2 and with a stride of two pixels to output pool1. After a series of operations of the first layer convolution, the output of pool1 is the result of feature calculation, and the size is 351×312×64. Similar to the operation of the first layer, the output characteristics of the second layer, the third layer, and the fourth layer are pool2, pool3 and pool4, the sizes are 175×156×128, 87×78×256,and 43×39×512, respectively. Conv5 does not perform MAX pooling operations. It performs only three times convolutions and ReLu operations similar to the first layer. The output of this layer is conv5_3, so the size of conv5_3 is same as the size of pool4 (43×39×512). In order to make the scale of the conv3 same as the last two layers, we make pool3 perform the MAX pooling operation with kernel_size equal 2 and the stride equal 2, the output named pool_out and the size is 43×39×256. Then we perform a convolution operation on pool_out with 1×1×512, the output named pool_out_1 and the size is 43×39×512.

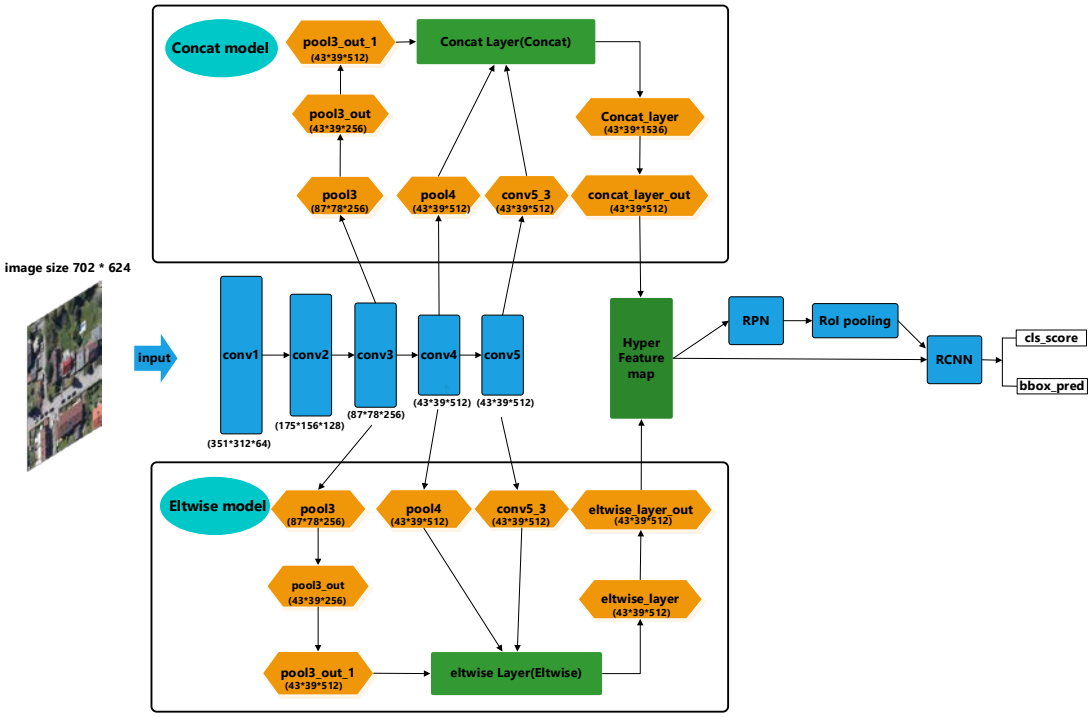


Fig. 2. The overall architecture of Hyper Feature Map network

Through the above series of operations, we have made the output of the last three layers consistent in size. We establish hyper feature maps from three convolutional layers (namely pool3_out_1, pool4 and Conv5_3), which have the same size but different levels of detail information. In addition, because the Concat layer connects the last three layers to a size of

43×39×1536, the Concat layer need to perform a convolution operation with 1×1×512, which will make the size of output same as Eltwise layer. Then perform relu operations on Eltwise layer and Concat layer and transmit the output to hyper feature map. We performed eltwise sum operation with Eltwise layer and Concat layer in the hyper feature map. At this point, the Eltwise module and Concat module complete the fusion of features. As shallower layers are more suitable for location and deeper layers are more suitable for classification, the fused hyper feature map is complementary for small-size vehicle detection.

We perform a series of operations on the characteristics extracted from the RPN layer through "Reshape", "SoftmaxWithLoss", etc., and pass the corresponding feature parameters to the ROI proposal layer and generate the ROI region. The ROI region features are passed to the RCNN, then the cls_score layer and bbox_pred layer are generated through the computation of two fully connected layers (namely fc6 and fc7). The cls_score layer is used to output the predicted score for each category, the size of output is the number of all categories. The bbox_pred layer is used to predict the coordinates of bounding boxes, there are four output values for each category, the corresponding feature vector is loc = (x, y, w, h), where x and y represent the top-left coordinates of the predicted region, whereas, w and h denote the width and height of the predicted region.

3.1.3 Training Stage

In order to cope with the impact of insufficient training data on the results, we initialize our model with a pre-trained VGG16 model on ImageNet, and then fine tune it with a smaller learning rate. We perform 80k iterations on our model and set the batch size on 256. In each iteration, our model predicts the category and bounding boxes of the image blocks. The Intersection-over-Union (IoU) indicates that the ratio of the overlapping of the prediction area and the ground-truth box. If the value of IoU is greater than 0.5, we assign a positive label to it. However, if the IoU is lower than 0.3 for all ground-truth boxes, we assign a negative label to it. Then the rest of the region does not need to be considered. The IoU ratio is defined as follows:

$$IoU = \frac{area(C) \bigcap area(G)}{area(C) \bigcup area(G)} \tag{3}$$

Where $area(C) \bigcap area(G)$ represents the intersection of the vehicle proposal box and ground truth box, and $area(C) \bigcup area(G)$ represents their union.

All labeled positive and negative samples and related region proposals features are fed to the loss function, and a robust method of object classification and detection is established by iteration. In addition, the multitasking loss function updates network parameters iteratively, the purpose is to minimize the error rate of classification and localization. The L_{cls} is a loss function for the classification of vehicles and backgrounds in each region by a softmax function, and the L_{bbr} is used for box-regression. Similar to [20], the loss function is defined as shown in formula (4):

$$L(p_i, loc_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{bbr}} \sum_i p_i^* L_{bbr}(loc_i, loc_i^*) \tag{4}$$

Where i is the index of an anchor in the mini-batch. p_i is the score that predicted the probability of anchor i being an object in each region. p_i^* is the label of ground-truth, which equal 1 if the anchor is positive, and equal 0 if the anchor is negative. t_i is a vector representing the 4 parameterized coordinates of the predicted bounding box, and t_i^* is the ground-truth box associated with a positive anchor. The two terms are normalized by N_{cls} and N_{bbr}, and the weighted by a balancing parameter λ. In each iteration, the number of positive and negative region boxes are almost the same. Therefore, we set λ=2 to make L_{cls} and L_{bbr} have the same weight. Moreover, L_{bbr} denotes a smooth L_1 loss, which same as in Fast R-CNN [19]. It is defined as Equation (5):

$$L_{bbr}(loc, loc^*) = f_{L1}(loc_i - loc_i^*),$$
$$f_{L1}(x) = \begin{cases} 0.5x^2, if \mid x \mid < 1, \\ \mid x \mid -0.5, otherwise \end{cases} \tag{5}$$

In Equation 5, *loc* represents the predicted value of the vector *loc* = (x, y, w, h) in each region, and loc^* represents the ground-truth box of the related object vector. The *loc* and loc^* represent the predicted bounding box and ground-truth bounding box respectively. The $f_{L1}(x)$ is a robust smooth L_1 loss that less sensitive to outliers. In addition, the parameters of weight are initialized in the new layer from a zero-mean Gaussian distribution with standard deviation 0.01. In order to suppress those redundant boxes, we use a non-maximal suppression algorithm, which is an iterative-ergodic-elimination process. First, it sorts all the boxes and select the highest score box. Next, it traverses the rest of the box, if the overlap area (IoU) of the current maximum framing is greater than a certain threshold, it will delete the box. Finally, it continues to select the highest score in the box that has never been processed and repeat the above process. In our model, we set NMS=0.4. The vehicle-like features are passed to the ROI propossal layer and RCNN, and then the bounding boxes and categories of the object are predicted by two fully connected layers.

3.2 Anchor Box

The essence of anchor is the reverse of the idea of SPP (spatial pyramid pooling). The basic idea is to push the output of the same size back to different sizes of inputs. In Faster R-CNN the anchor box have three area sizes with three different aspect ratios. In this way, it obtains a total of nine kinds of anchors of different sizes. The schematic diagram of anchor box is shown in **Fig. 3**.

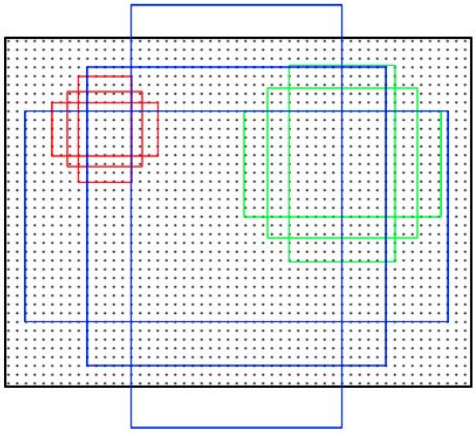


Fig. 3. The schematic diagram of Anchor Box

Based on the obtained characteristic parameters, the coordinates of the center point of the corresponding original picture are calculated through a 3×3 sliding window. At each window position, it can reverse a region in the original image according to the different anchor, then the size and coordinates of the region are obtained. Next, each proposal outputs the category of the predicted object and the coordinates of bounding box.

Table 1. The size of anchor box in Faster R-CNN and our model

Method	Ratio=0.5	Ratio=1.0	Ratio=2.0
Faster R-CNN	184×96 368×192 736×384	128×128 256×256 512×512	88×176 176×352 352×704
Our model	60×30 80×40 100×50	45×45 60×60 75×75	30×60 40×80 50×100

However, the size of the anchor boxes set in the Faster R-CNN is used to detect the object in the natural scene, which is not suitable for detect a specific small object, especially on the aerial image. Therefore, we need to set the appropriate size of the anchor box for a specific detection task. In the Faster R-CNN, the anchor size is set to: base size is 16, ratios is (0.5, 1, 2), and scales is (8, 16, 32). After a series of calculations, it finally produces nine anchors with three scales (128, 256, 512) and three ratios (0.5, 1, 2), as shown in **Table 1**. The size of the anchor box generated in Faster R-CNN is not suitable for detection of small objects. As to how to set up suitable anchor box, yolo9000 [48] proposes to use k-means clustering algorithm to find suitable anchor box size. However, this algorithm is suitable for the objects with various sizes and scales, it is not suitable for specific object detection. In addition, the algorithm also significantly increases the computation and consumption. In this research, we can detect that the size of vehicles in the aerial image dataset is basically around 30×60, so we set the appropriate anchor box in our model according to this size. The anchor size is set to: base size is 3, and ratios is (0.5, 1, 2), scales is (15, 20, 25). After a series of calculations, 9 anchors of three scale (45, 60, 75) with three ratios (0.5, 1.0, 2.0) are finally generated (as shown in **Table 1**). This size basically covers the size of most types of vehicles. Compared with the anchor box set in the Faster R-CNN, the introduction of a specific anchor box increased the mAP by about 3%, at the same time the detection time is reduced by approximately 20%.

4. Experimental Results

In this section, we show the results of our method in vehicle detection, and analyze the results in detail. The experimental procedure is based on the deep learning framework Caffe. The configuration of the computer is as follows: CPU is Inter core i7-7700, GPU is NVIDIA GTX-1060 (6 GB video memory), and the memory is 8GB. The operating system is Ubuntu 14.04 (Canonical, London, UK).

4.1 Dataset Description

Two datasets are used in these experiments. The Munich Vehicle dataset is collected over the city Munich, Germany. Our Collected Vehicle dataset is captured over the city Nanjing, China. They are both high-resolution aerial vehicle datasets.

4.1.1 Munich Vehicle Dataset

As described in [49], the Munich Vehicle dataset is used in the paper "K.Liu and G.Mattyus: Fast Multiclass Vehicle Detection on Aerial Images, Geoscience and Remote Sensing Letters, IEEE, Volume: 12, Year 2015". Anyone can download the dataset from the link in [44]. The images are captured from an airplane by a Canon Eos 1Ds Mark III camera with a resolution of 5616×3744 pixels, 50 mm focal length and they are stored in JPEG format. The optical image is taken at a height of 1000 meters above ground, and the ground sampling distance is approximately 13cm. The Munich vehicle dataset annotated eight types of vehicle information, and most of the vehicle types is car. Following [44], we combine "ca" and "van" as car. In order to ensure the reliability of the data, in our research, we only detect car types. Due to the limited size of the training set and video memory, following [5], each original aerial image (5616×3744 pixels) is cropped into11×10 image blocks (702×624 pixels) with overlap. The blocks without vehicles are discarded and the remaining image blocks are rotated with four angles.

4.1.2 Our Collected Vehicle Dataset

The collected vehicle dataset contains 615 aerial images with a resolution of 1368×770 pixels. We have uploaded the dataset to the public repository, the readers can download the dataset from the link in [50]. The UAV captured the images at a height of about 60 meters. The car type vehicle is annotated on each image. An average of 30 car type samples are annotated in each image, so there are approximately 18450 samples with the ground truth in our dataset. We select eighty percent of the dataset as training samples and the remaining data as test samples. During training, the original image is rotated vertically, horizontally, and mirrored in three ways to expand the dataset. Most of these aerial data sets are obtained from the road.

4.2 Evaluation Index

In our model, we use four typical indicators to evaluate the detection performance, namely precision rate, recall rate, mAP and F1-score.

The definition of precision rate is as follow:

$$precision = \frac{TP}{TP + FP} \tag{6}$$

Where TP (True Positive) indicates the number of positive samples predicted to be positive, and FP (False Positive) indicates the number of negative samples predicted to be positive.

The recall rate is also an important index to measure the detection performance. It is defined as follows:

$$recall = \frac{TP}{TP + FN} \tag{7}$$

Where FN (False Negative) represents the number of positive samples predicted to be negative.

The precision rate and recall rate affect each other. Ideally, both are high, but in general, the precision rate is high and the recall rate is low, and vice versa. At this time, a comprehensive recall rate and precision rate need to be evaluated. F1-Measure is a weighted harmonic mean of precision and recall. The definition is as follows:

$$F_1 = \frac{2 \times precision \times reacll}{precision + reacll} \tag{8}$$

The mAP is designed to solve the single-point value limitations of precision rate, recall rate, and F1-score. Its purpose is to obtain an index that reflects global performance. The definition of the mean average precision (mAP) is as follows (where P and R represent the precision rate and recall rate respectively):

$$mAP = \int_0^1 P(R)dR \tag{9}$$

4.3 Results for Munich Vehicle Dataset

The results of our experiments are shown in **Table 2**. In **Table 2**, we can see that the performance of the VGG16 model is significantly better than the ZF model in the Faster R-CNN, which is the reason why we chose the VGG16 model. Compared with the ZF model, the VGG16 model has improved by nearly 0.1 in F1-score and 12.8 points in mAP. This fully demonstrates the superior performance of the VGG16 model in the field of object detection. In the detection model, we introduce the specific anchor box method to improve the performance of the detection. Compared with the non specific anchor box, the recall and precision rate are increased by 1.1 points and 3.0 points respectively. At the same time, F1-score and mAP are also improved significantly. More details, the detection speed is increased by approximately 20%. These can fully demonstrate the importance of setting a specific anchor box based on the size of the object in object detection.

Table 2. The results of method under different indicators

Method	Reacll Rate	Precision Rate	F1-Score	mAP	Time/per image
Faster-RCNN(ZF)	69.7%	68.6%	0.691	56.7%	4.14
Faster-RCNN(VGG16)	78.5%	78.7%	0.786	69.5%	4.92
Faster-RCNN(Our Anchor)	79.6%	81.7%	0.806	73.2%	**3.93**
Faster-RCNN(Eltwsie model)	81.2%	88.3%	0.846	82.6%	4.27
Faster-RCNN(Concat model)	80.5%	89.1%	0.846	82.8%	4.31
Our method	**82.3%**	**90.2%**	**0.861**	**85.5%**	4.56

In order to further demonstrate the role of Eltwsie model and Concat model in our experiments, we test Eltwsie model and Concat model respectively. The results show that the introduction of Eltwsie model and Concat model can further improve the detection performance. More details, we find that the Eltwsie model is more obvious in increasing the recall rate, and the Concat model is more prominent in improving the precision rate. In addition, our model achieved the most advances performance after the fusion of the Eltwsie model and the Concat model, and all performance indicators have been greatly improved. Our method eventually achieves a recall rate of 82.3% on the Munich dataset. At the same time the precision is 90.2%, the F1-score is 0.861 and the mAP is 85.5%. These indicators are currently reaching the leading level. In addition, our model has a training time of 0.578 seconds per iteration on the Munich dataset. The model was iterated 80000 times, so the total training time is about 13 hours.

Table 3. Performance comparison between different methods

Method	Reacll Rate	Precision Rate	F1-Score	Time/per image
Fast multiclass[2]	75.2%	81.3%	0.781	4.27
VPN_VAD[3]	78.2%	79.1%	0.786	4.83
AVPN[4]	77.6%	88.5%	0.827	4.91
HRPN[5]	79.8%	88.9%	0.841	4.85
Our method	**82.3%**	**90.2%**	**0.861**	4.56

We compare our method with state-of-the-art detection methods, as shown in **Table 3**. To be fair, we do not copy the results of other algorithms directly, and all algorithms were reproduced on the Munich dataset. The best performances are highlighted in bold. It can be observed our proposed method achieves the best performance. Our results have reached the leading level in terms of recall rate, precision rate and F1-score. In our model we use a specific anchor box which allows the model to speed up the convergence and reduce the time of detection. Therefore, our model is also considerable in detection time compared to other methods. In our method, the Etlwise model and Concat model are introduced, and the appropriate anchor box is set according to the size of the object, which make our results reach the leading level in terms of recall rate and precision rate.

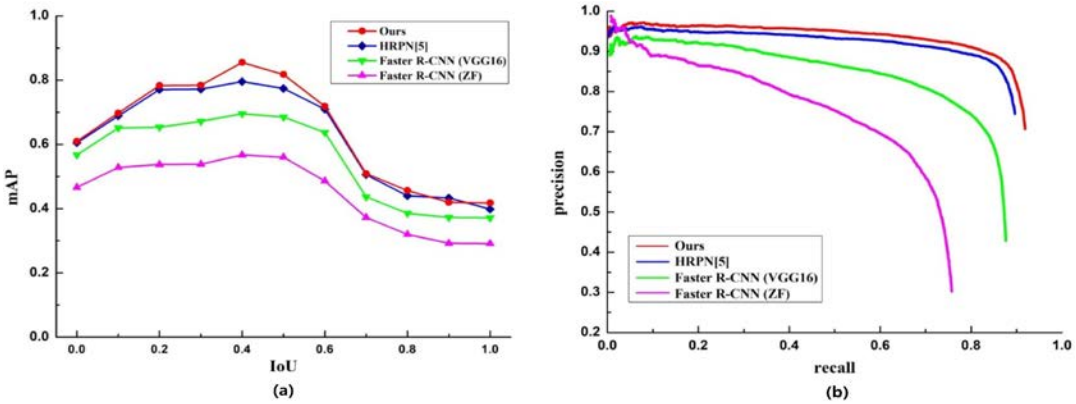


Fig. 4. Comparisons of four detection models (a) mAP vs. IoU curve , (b) precision-recall curve

In addition, the mAP-IoU curve and precision-recall curve are showed in **Fig. 4**. **Fig. 4(a)** shows the mAP changes with IoU. Compare with other methods, the result of ours has a higher mAP value when the IoU at different values. With the increase of IoU (from 0 to 1), the mAP appears increase first and then decrease. In particular, the mAP reaches the highest value when IoU is around 0.4, so in our model we choose IoU=0.4 in test. Accuracy rate and recall rate are important basis for evaluating the detection performance of the model. These two indicators interact with each other. If the accuracy rate is high, the recall rate is low, and vice versa. **Fig. 4(b)** shows the performance of our model and several other methods on precision-recall. Obviously, our results are also superior to other methods in terms of recall-precision. In more detail, our model has obvious advantages over Faster R-CNN on precision-recall. At the same time, our model has some advantages compared with the HRPN method.

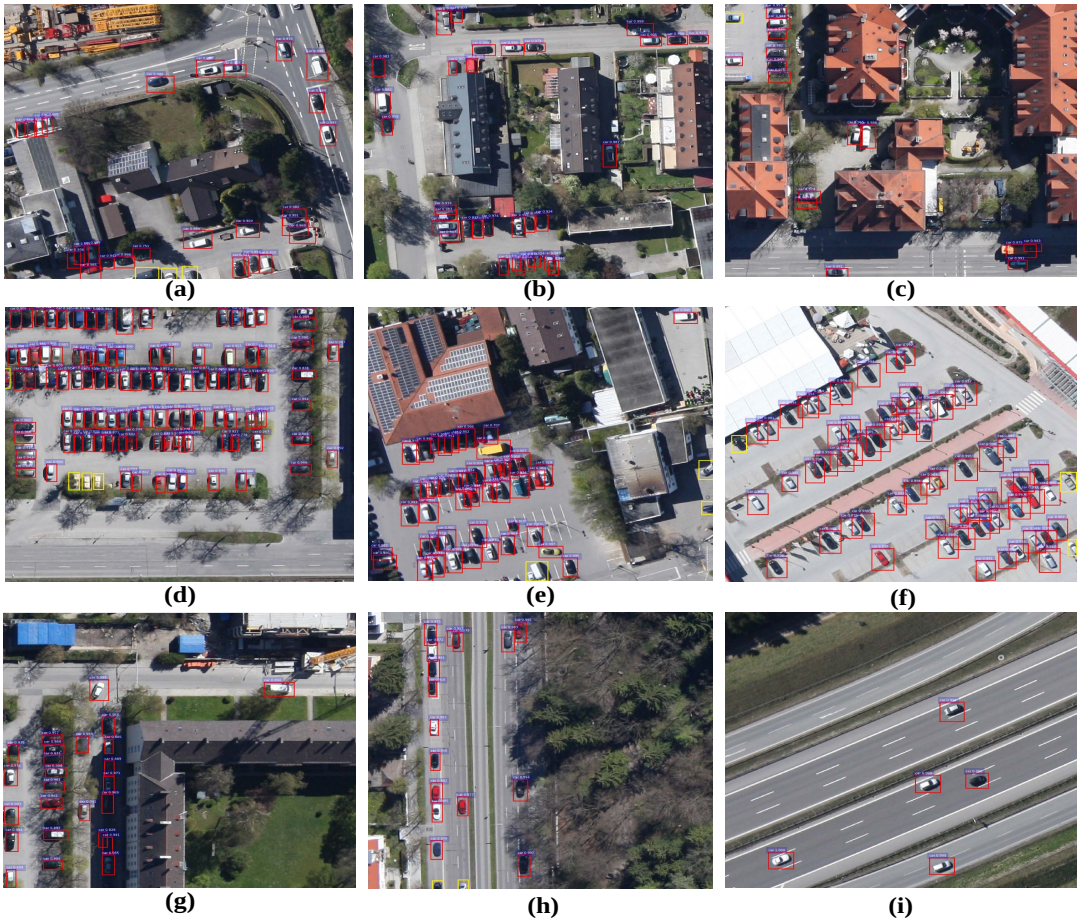


Fig. 5. Detection results for the Munich test aerial images. Red boxes denote correct localization, yellow boxes denote missing detection

Fig. 5 shows several results of test image blocks on Munich dataset with the method which we proposed. The red box represents the correct localization, and the yellow box represents miss detection. Each red box gives a score, which indicates the reliability of the object. The higher the score, the higher the credibility of the prediction. **Fig. 5** shows that our method can detect most of the vehicles in various scenarios, which indicates that our method is effective. **Fig. 5(a, b, c)** shows that our method exhibits good detection performance in a complex

background. As shown in **Fig. 5(d, e, f)**, when the cars are in a dense scene our method can also show good detection results. When the vehicle is covered by shadows, such as trees or buildings in the shadow of light, as shown in **Fig. 5(b, d, g)**, the results show that our method still have superior performance. This shows that our method can successfully detect the object vehicle in the shadowed area. **Fig. 5(h, i)** shows the detection results in a single background. In this case, the detected objects have high credibility, and most of the predicted scores are above 0.95.

As shown in **Fig. 6**, all the detection results of blocks are stitched together to recombine the original image. There are 1158 car type samples in **Fig. 6**. It is shown that only 30 samples are miss detection with our method. In this case, the accuracy of our method is about 97.4%.

Fig. 6. Detection results in original images. Red boxes denote correct localization, yellow boxes denote missing detection and incorrect detection

4.4 Results of Our Collected Dataset

In order to demonstrate the effectiveness of our method, we also implemented our method on the collected dataset. As shown in **Table 4**, our method achieves optimal performance in terms of recall rate, precision, and F1-score. Compared with the Munich vehicle data set, the aerial image in our dataset has a higher resolution, so the detection indexes on the collected datasets are higher generally. Our method get a recall rate of 91.5% in our collected dataset, at the same time the precision is 92.9%, the F1-score is 0.92. This shows that our method still has superior detection performance in our collected dataset. The model spends 0.752 seconds per training session on our collected dataset, and iterates over 80,000 times, so the total training time isabout 17 hours.

Table 4. Results of different methods for the collected vehicle images

Method	Reacll Rate	Precision Rate	F1-Score	Time/per image
Fast multiclass [2]	77.5%	83.4%	0.80	**0.212**
VPN_VAD [3]	87.2%	80.7%	0.84	0.241
AVPN [4]	86.7%	84.9%	0.86	0.245
HRPN [5]	85.5%	86.2%	0.86	0.242
Our method	**91.5%**	**92.9%**	**0.92**	0.221

The F1 Score is an indicator used in statistics to measure the accuracy of a binary model. It also takes into account the accuracy and recall rate of the classification model. The F1 score can be seen as a weighted average of the model accuracy and recall rate, with a maximum of 1 and a minimum of 0. The mAP indicator is the average precision, which is aimed to address the single point value limitations of Precision, Recall, and F-measure. Both MAP and F1-score can reflect the performance of the model globally. **Fig. 7(a)** shows that the value of F1-score and mAP in different IoU. With the increase of IoU (from 0 to 1), the value of F1-score and mAP present a trend of increasing first and then decreasing. In particular, in our method the F1-score and mAP have the best detection performance when IoU=0.4. This phenomenon is similar to the results on the Munich dataset. In the detection of vehicle objects based on aerial images, the objects are dense and numerous, and the predicted bounding boxes easily form overlapping areas, so it is very important to set an appropriate IoU. In addition, we compare the results of precision-recall on four different methods. **Fig. 7(b)** shows the results of our results and the other four methods on precision-recall. As shown in **Fig. 7(b)**, the results of our method have significant advantages in detection performance compared to other methods. In particular, the results of our model show a more significant advantage in recall rate greater than 30%.

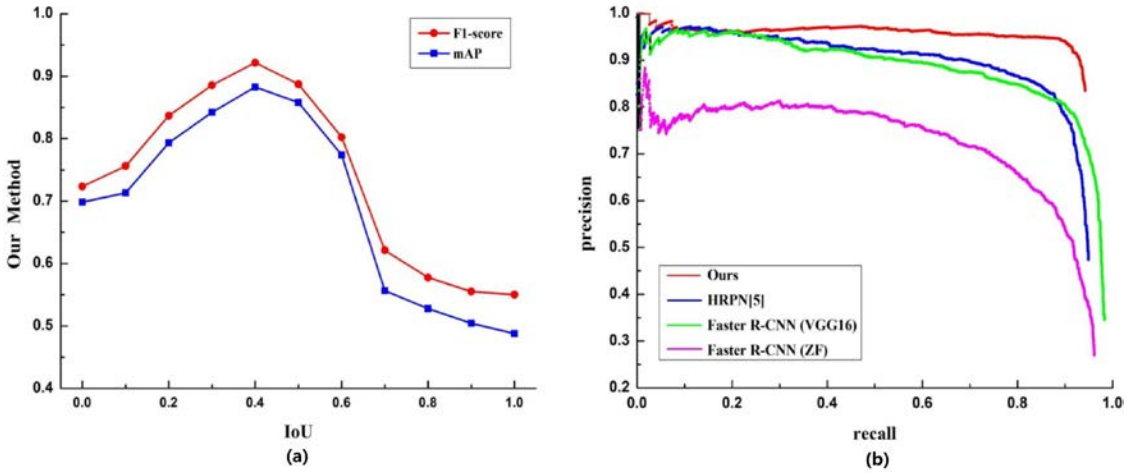


Fig. 7. Results for the collected vehicle images. (a)The value of F1-score and mAP in different IoU ,(b) Comparisons of four detection models for precision-recall curve

Fig. 8 shows the results of our method on the collected dataset. As shown in **Fig. 8**, our method can successfully detect most of the object vehicles in various backgrounds. **Fig. 8(a, b, c)** show that the detection object is in a simple scene, and the object is not located in the boundary area, in this case, our method can detect the object without any miss detection. **Fig. 8 (d, e, f)** show that when the background is relatively complex and the detection objects are

dense, our method can also complete the detection of most of the objects. As shown in **Fig. 8 (e)**, most of the missed objects are concentrated in the boundary area of the image. These objects can only extract part of the features, so it is easy to cause missed detection.

Fig. 8. Detection results for the Collected test aerial images. Red boxes denote correct localization of car, yellow boxes denote missing detection

5. Co nclusions

In this paper, we proposed an accurate and effective vehicle detection method in aerial image. In our method, we have established a hyper feature map network to extract the characteristics of the object vehicle. This network was created through fusing Eltwsie model and Concat model, which is more suitable for the detection of small objects. Moreover, we designed the appropriate anchor box according to the size of the object, which further improved the performance of the detection. We evaluated our proposed method on the Munich vehicle image dataset and our collected dataset. Compared with the most advanced detection methods, our method has the best performance. The results of our model reached 82.3% in recall rate

and 90.2% in accuracy on the Munich vehicle dataset. It has increased by 2.5 and 1.3 percentage points respectively over the state-of-the-art methods. The method which we proposed can successfully detect objects in a variety of complex backgrounds.

However, our method still has some miss detection. The extraction of excellent vehicle characteristics is still a critical task for accurate vehicle detection. Apart from this, the training of the model takes about 13 hours in our experiments on Munich dataset, so it is currently unable to perform real-time detection. For future work, we will pay attention to the feature extraction of the missing targets for further improvement of the detection performance. In addition, we will optimize the structure of the network to reduce the computation time.

Author Contributions

Jiaquan Shen and Ningzhong Liu conceived and designed the experiments; Jiaquan Shen and Han Sun performed the experiments and analyzed the data; Jiaquan Shen developed the algorithm and wrote this paper. Xiaoli Tao and Qiangyi Li contributed experiment tools. All authors contributed in reviewing the article.

Funding

This research is supported in part by National Natural Science Foundation of China (No. 61375021) and the Fundamental Research Funds for the Central Universities (No. NS2016091).

Acknowledgments

The authors would like to thank the editor and anonymous reviewers for their valuable comments and suggestions, which are helpful in improving this paper. Thanks to those who contributed to the collection and annotation of our dataset.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Yi Meng, Baolong Guo and Chunman Yan, “Improved image alignment algorithm based on projective invariant for aerial video stabilization,” *KSII Transactions on Internet & Information Systems*, vol. 8, no. 9, pp. 3177-3195, September, 2014. Article (CrossRef Link).

[2] Liu Kang and Gellert Mattyus, “Fast Multiclass Vehicle Detection on Aerial Images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 9, pp. 1938-1942, September, 2015. Article (CrossRef Link).

[3] Zhong Jiandan, Tao Lei and Guangle Yao, “Robust Vehicle Detection in Aerial Images Based on Cascaded Convolutional Neural Networks,” *Sensors*, vol. 17, no. 12, pp. 2720-2737, November, 2017. Article (CrossRef Link).

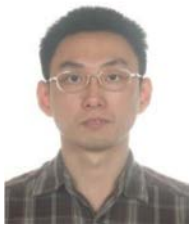
[4] Zhipeng Deng, Hao Sun, Shilin Zhou, Juanping Zhao and Huanxin Zou, “Toward Fast and Accurate Vehicle Detection in Aerial Images Using Coupled Region-Based Convolutional Neural Networks,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no 8. pp. 3652-3664, August, 2017. Article (CrossRef Link).

[5] Tianyu Tang, Shilin Zhou, Zhipeng Deng, Huanxin Zou and Lin Lei, "Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining," *Sensors*, vol. 17, no. 2, pp. 336-352, February, 2017. Article (CrossRef Link).
[6] Yanjun Liu, Na Liu, Hong Huo, Tao Fang, "Vehicle detection in high resolution satellite images with joint-layer deep convolutional neural networks," in *Proc. of International conference on mechatronics and machine vision in practice*, pp. 1-6, November, 2016. Article (CrossRef Link).
[7] Juanjuan Zhu, Wei Sun, Baolong Guo and Cheng Li, "Surf points based Moving Target Detection and Long-term Tracking in Aerial Videos," *Ksii Transactions on Internet & Information Systems*, vol. 10, no. 11, pp. 5624-5638, November, 2016. Article (CrossRef Link).
[8] Thomas Moranduzzo and Farid Melgani, "Detecting cars in UAV images with a catalog-based approach," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 52, no. 10. pp. 6356-6367, January, 2014. Article (CrossRef Link).
[9] Thomas Moranduzzo and Farid Melgani, "Automatic car counting method for unmanned aerial vehicle images," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 52, no. 3, pp. 1635-1647, May, 2014. Article (CrossRef Link).
[10] Yongzheng Xu, Guizhen Yu and Xinkai Wu, "An Enhanced Viola-Jones Vehicle Detection Method from Unmanned Aerial Vehicles Imagery," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1-12, July, 2017. Article (CrossRef Link).
[11] Gong Cheng and Junwei Han, "A survey on object detection in optical remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 11-28, July, 2016. Article (CrossRef Link).
[12] PDA Kraaijenbrink, JM Shea, F Pellicciotti, SMD Jong and WW Immerzeel, "Object-based analysis of unmanned aerial vehicle imagery to map and characterise surface features on a debris-covered glacier," *Remote Sensing of Environment*, vol, 186, no. 1, pp. 581-595, December, 2016. Article (CrossRef Link).
[13] Hailing Zhou, Hui Kong, Lei Wei, Douglas Creighton, Saeid Nahavandi, "Efficient Road Detection and Tracking for Unmanned Aerial Vehicle," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 297-309, February, 2015. Article (CrossRef Link).
[14] Hsu-Yung Cheng, Chih-Chia Weng and Yi-Ying Chen, "Vehicle Detection in Aerial Surveillance Using Dynamic Bayesian Networks," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2152-2159, April, 2012. Article (CrossRef Link).
[15] Wen Shao, Wen Yang, Gang Liu, Jie Liu, "Car detection from high-resolution aerial imagery using multiple features," in *Proc. of 2012 IEEE International Geoscience and Remote Sensing Symposium*, November, pp. 4379-4382, 2012. Article (CrossRef Link).
[16] Ziyi Chen, Cheng Wang, Chenglu Wen, Xiuhua Teng, Yiping Chen, Haiyan Guan and Huan Luo, "Vehicle detection in high-resolution aerial images via sparse representation and superpixels," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 1, pp. 103-116, January, 2015. Article (CrossRef Link).
[17] Aniruddha Kembhavi, David Harwood and Larry S. Davis, "Vehicle detection using partial least squares," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1250-1265, June, 2011. Article (CrossRef Link).
[18] Ross B Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142-158, May, 2016. Article (CrossRef Link).
[19] Ross Girshick, "Fast R-CNN," in *Proc. of 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440-1448, 2015. Article (CrossRef Link).
[20] Shaoqing Ren, Kaiming He, Ross B Girshick and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, June, 2017. Article (CrossRef Link).
[21] Joseph Redmon, Santosh Kumar Divvala, Ross B Girshick and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788, 2016. Article (CrossRef Link).

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E Reed, Chengyang Fu and Alexander C Berg, "SSD: Single shot multibox detector," in *Proc. of European conference on computer vision*, pp. 21-37, 2016. Article (CrossRef Link).
[23] Guimei Cao, Xuemei Xie, Wenzhe Yang, Quan Liao, Guangming Shi and Jinjian Wu, "Feature-Fused SSD: Fast Detection for Small Objects," in *Proc. of SPIE 10615, Ninth International Conference on Graphic and Image Processing (ICGIP 2017)*, 2018. Article (CrossRef Link).
[24] Paul Viola, John C. Platt and Cha Zhang, "Multiple instance boosting for object detection," *NIPS'05 Proceedings of the 18th International Conference on Neural Information Processing Systems*, pp. 1417-1424, December, 2005.
[25] D.N. Chandrappa, G. Akshay and M. Ravishankar, "Face Detection Using a Boosted Cascade of Features Using OpenCV," in *Proc. of Wireless Networks and Computational Intelligence , ICIP 2012*, pp. 399-404, 2012. Article (CrossRef Link).
[26] Timo Ojala, Matti Pietikainen and Topi Maenpaa, "Gray-scale and rotation invariant texture classification with local binary patterns," in *Proc. of IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 404-420, 2000. Article (CrossRef Link).
[27] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, November, 2004. Article (CrossRef Link).
[28] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Proc. of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886-893, 2005. Article (CrossRef Link).
[29] Pedro F Felzenszwalb, Ross B Girshick, David A Mcallester and Deva Ramanan, "Object Detection with Discriminative Trained Part Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, September, 2010. Article (CrossRef Link).
[30] Bushra Zafar, Rehan Ashraf, Nouman Ali, Mudassar Ahmed, Sohail Jabbar, Savvas A Chatzichristofis, "Image classification by addition of spatial information based on histograms of orthogonal vectors," *Plos One*, vol. 13, no. 6, June, 2018. Article (CrossRef Link).
[31] Nouman Ali, Khalid Bashir Bajwa, Robert Sablatnig and Zahid Mehmood, "Image retrieval by addition of spatial information based on histograms of triangular regions," *Computers & Electrical Engineering*, vol. 54, pp. 539-550, August, 2016. Article (CrossRef Link).
[32] Nouman Ali, Khalid Bashir Bajwa, Robert Sablatnig, Savvas A Chatzichristofis, Zeshan Iqbal, Muhammad Rashid, Hafiz Adnan Habib, "A Novel Image Retrieval Based on Visual Words Integration of SIFT and SURF," *Plos One*, vol. 11, no. 6, June, 2016. Article (CrossRef Link).
[33] Chia-Feng Juang and Guo-Cyuan Chen, "Fuzzy Classifiers Learned Through SVMs with Application to Specific Object Detection and Shape Extraction Using an RGB-D Camera," *Computational Intelligence for Pattern Recognition*, vol. 777, pp. 253-274, 2018. Article (CrossRef Link).
[34] Yanxiang Chen, Gang Tao, Hongmei Ren, Xinyu Lin and Luming Zhang, "Accurate seat belt detection in road surveillance images based on CNN and SVM," *Neurocomputing*, vol. 274, pp. 80-87, January, 2018. Article (CrossRef Link).
[35] Xiangbo Shu, Jinhui Tang, Guojun Qi, Yan Song, Zechao Li, Liyan Zhang, "Concurrence-Aware Long Short-Term Sub-Memories for Person-Person Action Recognition," in *Proc. of Computer vision and pattern recognition*, pp. 2176-2183, 2017. Article (CrossRef Link).
[36] Xiangbo Shu, Guojun Qi, Jinhui Tang and Jingdong Wang, "Weakly-Shared Deep Transfer Networks for Heterogeneous-Domain Knowledge Propagation," in *Proc. of the 23rd ACM international conference on Multimedia*, pp. 35-44, 2015. Article (CrossRef Link).
[37] Yongzheng Xu, Guizhen Yu, Yunpeng Wang, Xinkai Wu and Yalong Ma, "Car detection from low-altitude UAV imagery with the faster R-CNN," *Journal of Advanced Transportation*, vol. 2017, pp. 1-10, August, 2017. Article (CrossRef Link).

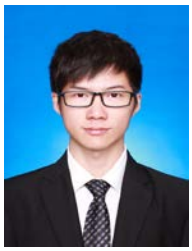
[38] Yakoub Bazi and Farid Melgani, "Convolutional SVM Networks for Object Detection in UAV Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3107-3118, June, 2018. Article (CrossRef Link).
[39] Mesay Belete Bejiga, Abdallah Zeggada, Abdelhamid Nouffidj and Farid Melgani, "A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery," *Remote Sensing*, vol. 9, no. 2, pp. 100-121, January, 2017. Article (CrossRef Link).
[40] Faisal Riaz, Sohail Jabbar, Muhammad Sajid, Mudassar Ahmad, Kashif Naseer and Nouman Ali, "A collision avoidance scheme for autonomous vehicles inspired by human social norms," *Computers & Electrical Engineering*, vol. 69, pp.690-704, 2018. Article (CrossRef Link).
[41] Lars Wilko Sommer, Tobias Schuchert and Jurgen Beyerer, "Deep learning based multi-category object detection in aerial images," in *Proc. of SPIE*, vol. 10202, 2017. Article (CrossRef Link).
[42] Igor Sevo and Aleksej Avramovic, "Convolutional Neural Network Based Automatic Object Detection on Aerial Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 5, pp. 740-744, May, 2016. Article (CrossRef Link).
[43] Nassim Ammour, Haikel Salem Alhichri, Yakoub Bazi, Bilel Benjdira, Naif Alajlan and Mansour Zuair, "Deep learning approach for car detection in UAV imagery," *Remote Sensing*, vol. 9, no. 4, pp. 1-15, March, 2017. Article (CrossRef Link).
[44] Gong Cheng, Peicheng Zhou, Junwei Han, "Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7405-7415, December, 2016. Article (CrossRef Link).
[45] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *Proc. of European conference on computer vision*, pp. 818-833, 2014. Article (CrossRef Link).
[46] Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proc. of International conference on learning representations*, 2015. Article (CrossRef Link).
[47] Amir Ghodrati, Ali Diba, Marco Pedersoli, Tinne Tuytelaars and Luc Van Gool, "Deepproposal: Hunting objects by cascading deep convolutional layers," in *Proc. of international conference on computer vision*, pp. 2578-2586, December, 2015. Article (CrossRef Link).
[48] Joseph Redmon and Ali Farhadi, "YOLO9000: Better , Faster , Stronger," in *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517-6525, 2017. Article (CrossRef Link).
[49] K. Liu and G. Mattyus, "DLR 3k Munich Vehicle Aerial Image Dataset," *Available online: http://pba-freesoftware.eoc.dlr.de/3K_VehicleDetection_dataset.zip*, 2015.
[50] https://pan.baidu.com/s/1mz-phfgwG3VdF0OASAI14g. (Datasets Link)

Jiaquan Shen was born in 1992. He received the M.S. degree in Computer Science and Technology from Wenzhou University, Wenzhou, China in 2017. He is currently pursuing the PH.D. Degree in Nanjing University of Aeronautics and Astronautics. His research interests include deep learning and computer vision.



Ningzhong Liu was born in 1975. He received the B.S. degree in Computer Engineering from Nanjing University of Science and Technology, Nanjing, China in 1998, and received the Ph.D. degree in Pattern Recognition and Intelligent Systems from Nanjing University of Science and Technology in 2003. He is currently a professor of the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include artificial intelligence and machine learning.

Han Sun received the B.S. degree in Computer Engineering from Nanjing University of Science and Technology, Nanjing, China in 2000, and received the Ph.D. degree in Pattern Recognition and Intelligent Systems from Nanjing University of Science and Technology in 2005. He is currently an associate professor of the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include machine learning and image processing.



Xiaoli Tao received the B.S. in Nanjing University of Aeronautics and Astronautics, Nanjing, China in 2016. He is currently pursuing the M.S. degree in Nanjing University of Aeronautics and Astronautics. His research interests machine learning and computer vision.

Qiangyi Li is currently working toward the Ph.D. at the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Jiangsu Nanjing, China. His research interests include digital image processing and pattern recognition.