A feature fusion deep-projection convolution neural network for vehicle detection in aerial images

Bin Wang¹,², Bin Xu¹,²*

¹ School of Communication and Information Engineering, Shanghai University, Shanghai, China,
² Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai, China

* bingoxushu@163.com

Abstract

With the rapid development of Unmanned Aerial Vehicles, vehicle detection in aerial images plays an important role in different applications. Comparing with general object detection problems, vehicle detection in aerial images is still a challenging research topic since it is plagued by various unique factors, e.g. different camera angle, small vehicle size and complex background. In this paper, a Feature Fusion Deep-Projection Convolution Neural Network is proposed to enhance the ability to detect small vehicles in aerial images. The backbone of the proposed framework utilizes a novel residual block named stepwise residual block to explore high-level semantic features as well as conserve low-level detail features at the same time. A specially designed feature fusion module is adopted in the proposed framework to further balance the features obtained from different levels of the backbone. A deep-projection deconvolution module is used to minimize the impact of the information contamination introduced by down-sampling/up-sampling processes. The proposed framework has been evaluated by UCAS-AOD, VEDAI, and DOTA datasets. According to the evaluation results, the proposed framework outperforms other state-of-the-art vehicle detection algorithms for aerial images.

1. Introduction

As one of the core research topic of computer vision, object detection is widely used in automatic driving, crowd flow counting, topographic exploration, environmental pollution monitoring, etc. The task of object detection is to find out the various targets in the images, and determine the locations and categories of these targets. Because the appearance of objects changes significantly according to various factors [1–4], object detection is commonly regarded as one of the most challenging tasks in the field of computer vision.

The traditional object detection algorithms are normally based on hand-crafted features or textures. R.M. Haralick et al. proposed textural features for image classification in 1973 [5]. D.G. Lowe proposed scale-invariant feature transform (SIFT) in 1999 [6]. In 2014, T. Moranduzzo and F. Melgani used SIFT to count the number of vehicles and trained a support vector
machine (SVM) to classify vehicle and non-vehicle objects based on the SIFT descriptors [7]. S. Qi et al. proposed a ship detector for optical satellite images based on saliency and S-HOG descriptor in 2015 [8]. In recent years, due to the significant development of the neural network, object detection algorithms based on convolutional neural network (CNN) outperform traditional object detection algorithms in almost every aspect [9, 10]. According to [11, 12], object detection algorithms based on CNN could be roughly divided into two categories: two-stage and one-stage object detection algorithms. Two-stage object detection algorithms consist of two models. The first model is designed for distinguishing interesting targets from the background; the second model is responsible for classifying these interesting targets into various categories. Typical two-stage object detection algorithms are Faster R-CNN [13], Mask R-CNN [14], and so on. One-stage object detection algorithms integrate the region proposal model and the classification network into a single neural network to improve feature reusability and detection speed. Typical one-stage object detection algorithms are YOLOv3 [15], Single-Shot Multibox Detector (SSD) [16], RetinaNet [17] and so on.

Vehicle detection in aerial images, as an important branch of object detection, has been intensively researched recently. Along with the development of Unmanned Aerial Vehicles (UAVs), vehicle detection in aerial images plays an important role in various kinds of fields. Comparing with general object detection problems, vehicle detection in aerial images has its unique characteristics. According to [3, 18–21], the main difficulties of object detection in aerial images are as follows:

- **The particularity of perspective.** Aerial images are usually taken from the top of the sky, but most images of the conventional datasets were shot from the horizontal perspective at the ground. Therefore, the patterns of the objects are different in aerial images comparing with the same objects in conventional datasets. From the top view, the objects usually contain fewer texture features, and it is easier to confuse the targets with other objects.

- **The size of the objects.** Comparing to the objects in images taken on the ground, the objects in aerial images are generally small (composed by only dozens or even fewer pixels), thus these objects contain only a little information. Furthermore, the down-sampling layers in CNN will reduce the amount of information contained in each object, e.g. after four down-sampling layers, a 24 × 24 object retains only about one pixel in feature maps, which makes it difficult to distinguish small objects from the background.

- **The complexity of the background.** The receptive field of aerial images is relatively large (covering an area of several square kilometers). This receptive field may contain various backgrounds, e.g. farmland, mountains, urban area and so on, which will lead to strong and diversified interference to the object detection process.

In general, object detection in aerial images requires detecting small objects at specific angles under complex backgrounds. As a result, making full use of limited information is the key to accurate detection. The low-level features contain more detailed information, such as edges, corners, lines, color, and so on, while the high-level information contains more abstract semantic information. To get high-level semantic features, the input images need to be processed by a large number of convolutional layers, which will blur the low-level features. For small object detection, it even causes the loss of key features of objects, which usually leads to detection failures. Therefore, the state-of-the-art general object detectors usually do not perform well when processing aerial images.

Considering the unique requirements of vehicle detection in aerial images, a Feature Fusion Deep-Projection Convolution Neural Network (FFDP-CNN) is proposed in this paper. The main contributions of this paper are as follow:
1. A novel residual block named stepwise res-block is proposed. To explore high-level semantic features, a single stepwise res-block contains three 3x3 convolutional layers. These three layers form a hierarchical structure that produces features processed by various numbers of convolution layers. Comparing to the original res-block, the proposed stepwise res-block increases the depth of the network, and keeps the parameter number at a relatively low level. In addition to the shortcut connection adding to the output, the proposed stepwise res-block introduces another shortcut to conserve the low-level features.

2. Based on the proposed stepwise res-block, a backbone for small object detection is designed. This backbone is composed of 34 stepwise res-blocks. Comparing with the original ResNet or Res2Net with roughly the same parameter number, the proposed backbone provides more convolutional layers which enhance its capacities of extracting complex high-level semantic features. Meanwhile, the combination of multiple stepwise res-block enables the output features to be processed by various numbers of convolutional layers, as well as conserves the low-level features which are important in small object detection. Thus the feature generated by the proposed backbone contains both low-level detailed information and high-level semantic information at the same time.

3. A new feature fusion and detection network is proposed. To make full use of the high-level features and low-level features generated by the backbone, a feature fusion and multi-scale detection network is designed. The outputs from different levels of the proposed backbone are collected into the feature library through the spatial pyramid structure, and the features suitable for detection are obtained through an automatic selection process of the network. The selected features are then fed into the multi-scale detection module for detection. To reduce the information contamination introduced by up-sampling/down-sampling processes, a deep-projection deconvolution module is adopted in the proposed feature fusion network.

The rest of this paper is organized as follows. Section 2 reviews the state-of-the-art vehicle detection algorithms for aerial images; The implementation detail of the proposed Feature Fusion Deep-Projection Convolution Neural Network is described in section 3; Section 4 contains the evaluation results of the proposed framework using three different datasets; section 5 concludes this paper.

2. Related work

Due to the significant performance advantages of Convolution Neural Network (CNN), the most recent proposed vehicle detection algorithms for aerial images are based on various CNNs. In 2017, T. Tang et al. designed an improved Faster-RCNN to solve the difficulties of locating the positions of small vehicles and classifying the vehicle from the background [22]. The authors use a hyper region proposal network (HRPN) combined with different levels of feature maps to extract vehicles, and an enhanced classifier to reduce false detection.

In 2018, S. Liu et al. proposed a Path Aggregation Network (PANet) which aims at solving the problem of boosting information flow in a proposal-based instance segmentation framework [23]. By extending the bottom-up path, an accurate positioning signal can be obtained at the bottom layer. Thereby the entire feature hierarchy is enhanced, and the information path between the bottom layer and the topmost feature is shortened. The authors also proposed an adaptive feature pool to link all features, so that the useful information of each layer can be directly transmitted to the proposal subnetwork.

M.Y. Yang et al. presented a novel double focal loss convolutional neural network framework (DFL-CNN) in 2018 [24]. The proposed framework uses skip connection and focal loss
functions to improve detection performance. At the same time, a new large-scale vehicle detection dataset named ITCVD was proposed in the same paper. According to the authors, DFL-CNN obtains good performance on the ITCVD dataset.

In 2018, Y. Koga et al applied Hard Example Mining (HEM) to Stochastic Gradient Descent (SGD) in the process of training neural networks [12]. According to the authors, the application of HEM can improve detection accuracy by training the networks with more informative samples.

G. Cheng et al tried to solve the detecting problem related to object rotation, within-class diversity and inter-class similarity by adding a rotation-invariant regularizer and a fisher discrimination regularizer to the existing neural networks in 2018 [25]. Compared with the other state-of-the-art algorithms, the authors claimed that their method achieved good performance on various datasets.

In 2019, Murari Mandal et al. designed a one-stage vehicle detection network (AVDNet) [11]. The proposed network used ConvRes to hold the features of small objects on multiple scales. The author also proposed a recurrent-feature aware visualization (RFAV) technique to analyze the layers in the network.

W. Liu et al. also introduced an object detection algorithm for aerial images in 2019 [26]. A feature introducing strategy based on oriented response dilated convolution to make the model adaptable to multiscale objection detection.

Qiu H et al. proposed a novel end-to-end Adaptively Aspect Ratio multi-scale Network (A2RMNet) which focuses on detecting objects with various sizes and aspect ratios in 2019 [27]. The authors designed a feature gate fusion network to integrate multi-scale feature maps, and an aspect ratio attention network to prevent the changes of objects’ aspect ratios.

Artacho, B et al. proposed a new semantic segmentation network in 2019 [28]. The main structure of the network is named “Waterfall” Atrous Spatial Pooling (WASP) architecture. Compared with the traditional spatial pyramid structure, WASP can reduce the number of parameters used in the network, while increasing the receptive field of the network.

W. Li et al. proposed a network to detect and count vehicles simultaneously in aerial images in 2019 [4]. The authors utilized the combination of bottom-up cues and top-down attention mechanisms to maximize the use of mutual information between object categories and features. An effective loss function is used in the proposed network to promote the ability to push the anchors toward matching the ground-truth boxes as much as possible.

In 2020, Y. Yang et al. proposed a framework to solve the perspective distortion problem of aerial images [29]. The proposed framework utilized a reverse perspective network to evaluate perspective distortion, and evenly distort the image to obtain similar example scales. At the same time, the framework forces the regressor to learn from the augmented density maps via an adversarial network to further solve the problem of scale distortion in dense regions.

Rabbi, J et al. proposed a novel small object detection framework by combining a generative adversarial network (GAN)-based model called enhanced super-resolution GAN (ESRGAN), an Edge-Enhancement Network EEN and a detection network in 2020 [30]. The proposed framework achieved promising performance on both public datasets and self-assembled dataset.

A brief summary of some related works and their contributions is shown in Table 1.

### 3. Method

#### 3.1. Introduction of proposed framework

As shown in Fig 1, the proposed Feature Fusion Deep-Projection Convolution Neural Network (FFDP-CNN) can be roughly divided into two parts: a backbone network composed of
Thanks to the stepwise res-block, the proposed backbone produces features processed by various numbers of convolution layers, which helps to conserve low-level detailed information and produce high-level semantic information. The feature fusion network fuses the low-level and high-level feature maps and selects the information that is effective to vehicle detection in the different feature maps.

3.2. Introduction of stepwise res-block

The original residual block [31] is proposed to solve the vanishing gradient problem which appears as the depth of the network increases. By stacking up multiple residual blocks, networks with various depth have been proposed to explore high-level semantic features. However, in small object detection, low-level detailed information is considered as important as high-level semantic information. Although the side connection in the original residual block helps to increase the depth of the network, it cannot prevent the disappearance of low-level detailed information caused by convolutions. To obtain the low-level detailed information and high-level semantic information at the same time, a stepwise hierarchical convolution structure is created to replace the 3x3 convolution layer in the original residual block. Because the input feature is divided and processed hierarchically, like climbing stairs, the proposed res-block is named as stepwise res-block.

The structure of the proposed stepwise res-block is shown in Fig 2. \( S_i \) indicates a feature subset (i indicates the number of convolution layers the subset has been processed by, \( i \in (0,3) \)), \( j \) indicates the subset number, \( j \in (0,3) \). \( F_i \) indicates a feature map assembled by feature subsets (i indicate the number of this feature map). \( C_j \) indicates a 1x1 convolution layer. The input of stepwise res-block is first processed by a \( C_0 \) to decrease the channel number (to 1/4 of the input’s original channel number), and then split into even quarters \( S_0^{(0,3)} \) by the number of channel. \( S_0^{(0,3)} \) is directly superimposed onto \( F_4 \) without processing, while \( S_1^{(0,3)} \), \( S_2^{(0,3)} \), \( S_3^{(0,3)} \) are processed by 1x1 convolution layers before being superimposed onto \( F_4 \).
pass through $C_0$ to obtain $S_1^0$, $S_2^0$, $S_3^0$ (collectively called $F_1$). Then, $S_1^1$ is directly superimposed onto $F_4$ without processing, while $S_2^1$, $S_3^1$ pass through $C_1^3$ to obtain $S_2^1$, $S_3^1$ (collectively called $F_2$). After that, $S_2^2$ is directly superimposed onto $F_4$ without processing, while $S_3^2$ passes through $C_3^3$ to obtain $S_3^3$ (also named $F_2$) which is superimposed onto $F_4$. It can be seen that $S_0^0$, $S_1^0$, $S_2^0$ and $S_3^0$ are concatenated together to form feature map $F_4$. At the end of this process, $F_4$ is processed by another $1 \times 1$ convolution layer $C_1^4$ to restore the channel number to the input’s original channel number. Side connection is also used in stepwise res-block after $F_4$ passing through a $C_1^4$.

Different subsets of output processed by different numbers of convolution layers, resulting in different subsets containing features with different receptive fields. Among them, small receptive field subsets, such as $S_0^0$, $S_1^0$, pass through convolution layers fewer times, contain more detail information which is important for small object detection. Subsets with large receptive field, such as $S_2^2$, $S_3^3$, blur the detail information but can explore deep semantic information which is also important for detection [21]. Two $C_1^4$ are used at the beginning and end of the proposed block to automatically select the suitable features and implement a bottleneck to reduce the parameter number.

Comparing with the original res-block, stepwise res-block can provide more maximum depth with fewer parameters. Assuming that the dimension of the input and the outputs is $n \times n \times m$, the comparison of parameter number between the original res-block and stepwise res-block is shown in Table 2.
Compared with the original residual module (as shown in Table 2), the stepwise res-block can add two more convolution layer to the network (ignoring the 1 x 1 convolution layers), but the number of parameters has increased by only 30%, which is more suitable for composing deep network and exploring the underlying semantic information. The effectiveness and efficiency of the proposed stepwise res-block are validated in the experiments shown in section 4.4.

3.3. Introduction of backbone based on stepwise res-block

The stepwise res-block can be used in the mainstream networks to replace the convolution layer or the residual module, but to fully explore its advantages, a backbone based on the

Table 2. Parameter comparison between original res-block and stepwise res-block.

| Block name       | Original res-block | Stepwise res-block |
|------------------|--------------------|--------------------|
| Parameter of $C_0^0$ | $1 \times 1 \times m$ | $1 \times 1 \times m$ |
| Parameter of $C_0^1$ | $3 \times 3 \times m$ | $3 \times 3 \times \frac{m \times 3}{2}$ |
| Parameter of $C_1^0$ | $- \times 4$ | $3 \times 3 \times \frac{m \times 4}{2}$ |
| Parameter of $C_1^1$ | $- \times 4$ | $3 \times 3 \times \frac{m \times 4}{2} \times \frac{1}{2}$ |
| Parameter of $C_1^2$ | $1 \times 1 \times m$ | $1 \times 1 \times m$ |
| Total number of parameter | $3 \times 3 \times \frac{m \times 3}{2}$ | $3 \times 3 \times \frac{m \times 12}{7}$ |
| Max convolution Number (ignoring $C_1$) | 1 | 3 |
| Average parameter per convolution | $3 \times 3 \times \frac{m \times 3}{16}$ | $3 \times 3 \times \frac{m \times 13}{16}$ |
proposed stepwise res-block is designed in this section. The network structure is shown in Table 3.

Taking full advantage of stepwise res-block, the proposed backbone can explore high-level semantic information and conserve low-level details information at the same time. To explore high-level semantic information, 34 stepwise res-blocks (each stepwise res-block contains three 3x3 convolutional layers) is used in the proposed backbone. As a result, the maximum depth of the proposed backbone is 102 3x3 convolutional layers. As a comparison, widely used ResNet101 only contains 33 3x3 convolutional layers. A deeper network can generate better nonlinear expressions and semantic features. At the same time, the task of each convolutional layer becomes clearer while the depth of the networks goes deeper [21].

On the other hand, thanks to the stepwise res-block, the output of the proposed backbone has hierarchical receptive fields. Given the fact that the amount of input processed by 0–3 convolutional layers are divided equally in each stepwise res-block, the output of the proposed backbone contains feature generated by various numbers of convolutional layers (0–102, distributed equally if 1x1 convolutional layers are ignored), which increase the diversity of the features contained in the output. As a result, as well as high-level semantic features, there are plentiful low-level or mid-level features in the output.

### 3.4. Introduction of feature fusion deep-projection module and multi-scale detection network

Since the feature maps obtained from different levels of the proposed backbone have different dimensions, a feature fusion module is required to mix features generated by different levels of the proposed backbone together to ensure each detector receives feature maps containing full information of targets. To avoid feature loss, deep-projection deconvolution is used in the feature maps down/up-sampling process. As shown in Fig 3, feature maps with different scales go through the feature fusion deep-projection module and multi-scale detection network to generate detection results. The feature fusion deep-projection module is responsible for mixing Feature images of different levels and restoring them to the specified resolution. On the one hand, low-level and high-level features can be fused together to generate feature maps containing both semantic and detail information for the detectors. On the other hand, the deep-projection deconvolution module implemented in the down/up projection unit prevents the information contamination introduced by up-sampling/down-sampling steps. The detailed process of the proposed feature fusion deep-projection module is represented in section 3.4.1.
The multi-scale detection network of the proposed framework remains the same as YOLO V3, while the loss function is changed to mine hard samples. The detailed introduction of the multi-scale detection network can be found in section 3.4.2.

**Introduction of feature fusion deep-projection module.** In general, the output from different levels of the proposed backbone is sampled and concatenated into a collection of feature maps. Multiple $1 \times 1$ convolutions are implemented before and after the obtained collection of feature maps to implement a feature selection process. The selected feature is then restored to feature maps with different resolutions which finally sent to various detectors. Because of the proposed feature fusion network, each detector can use both the high-level and low-level information from the backbone.

Because of the down-sampling and up-sampling steps used in the proposed feature fusion network, information contamination is unavoidable. To reduce the impact of information contamination, inspired by the deep-projection unit used in super-resolution image reconstruction algorithms [32], a deep-projection deconvolution module is implemented in the proposed feature fusion network. The detailed structure of the proposed deep-projection deconvolution module is shown in Fig 4. It contains two different components: the up projection unit (for up-sampling) and the down projection unit (for down-sampling). In the up projection unit, as shown in Fig 4(A), the low-resolution feature maps $LR_1$ go through a deconvolution layer to get the high-resolution feature maps $HR_1$ firstly. Then the high-resolution feature maps $HR_1$ pass through a convolution layer to obtain a low-resolution feature maps $LR_2$. The residual result of the obtained low-resolution feature maps $LR_2$ and the initial low-resolution feature maps $LR_1$ goes through another deconvolution layer to obtain high-resolution feature maps $HR_2$. Finally, the high-resolution feature maps $HR_2$ (after passing through a convolution layer) and the high-resolution feature maps $HR_1$ obtained initially are added together to obtain the final high-resolution feature map $HR_3$.

In the down projection unit, as shown in Fig 4(B), the high-resolution feature maps $HR_1$ go through convolution layer to get the low-resolution feature maps $LR_1$ firstly. Then the low-resolution $LR_1$ passes through a deconvolution layer to obtain the high-resolution feature maps $HR_2$. The residual result of the obtained high-resolution feature maps $HR_2$ and the initial high-resolution feature maps $HR_1$ goes through a convolution layer to obtain the low-resolution feature maps $LR_2$. Finally, the low-resolution feature maps $LR_2$ (after passing through a convolution layer) and the low-
resolution feature maps $LR_1$ obtained initially are added together to obtain the final low-resolution feature map $LR_3$. These projection units continuously correct themselves and produce accurate feature maps which will be fed into 3 detectors later. The effectiveness of the proposed feature fusion deep-projection module is validated in the experiments shown in section 4.4.

3.4.2. Introduction of improved multi-scale detection network. The detection network used in the proposed framework is improved from the YOLOv3 framework. The anchor boxes used in the proposed detection network are optimized according to the target sizes of the evaluation datasets since the original anchor boxes used in YOLOv3 are obtained by clustering the target sizes in the COCO dataset and not suitable for vehicle detection in aerial images. Multi-scale detection is kept in the proposed detection network. Each scale uses different anchors, which is shown in Fig 5, and is responsible for detecting different groups of targets. The 80×80 scale detector, which is responsible for the mainstream targets detection tasks, is assigned to anchors with medium sizes, e.g. (65×40), (44×75), (40×65). The 20×20 scale detector, which is responsible for the relatively large targets detection tasks, is assigned to anchors with relatively large size, e.g. (75×42), (50×92), (108×40). The 160×160 scale detector, which is responsible for relatively small targets detection tasks, is assigned to anchors with relatively small size, e.g. (38×39), (34×59), (59×35).

The loss function is used to compute the error between the predicted value and true value. The loss function in our framework is improved from the loss function used in YOLOv3. The original loss function is consist of three parts: location prediction box loss, IOU loss, classification loss. The detailed description is introduced in [33].

$$\text{Loss} = \text{Loss}_{\text{location}} + \text{Loss}_{\text{class}} + \text{Loss}_{\text{confidence}},$$

$$\text{Loss}_{\text{location}} = \lambda_{\text{coord}} \sum_{i=0}^{B} \sum_{j=0}^{n} \rho_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{\text{coord}} \sum_{i=0}^{B} \sum_{j=0}^{n} \rho_{ij}^{\text{obj}} [(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2],$$

$$\text{Loss}_{\text{class}} = \sum_{i=0}^{B} \rho_{ij}^{\text{obj}} \sum_{j=0}^{n} [(p_i(c) - \hat{p}_i(c))^2],$$
\[ \text{Loss}_{\text{IOU}} = \sum_{i=0}^{s^2} \sum_{j=0}^{B} \rho_{ij} \left[ \left( c_i - c_j \right) \right] \]

\[ + \lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \rho_{ij} \left[ \left( c_i - c_j \right)^2 \right] \]

\[ \text{Loss}_{\text{cls}} = \text{M}_{\text{HSM}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \rho_{ij} \left[ \left( p_i(c) - \hat{p}_i(c) \right)^2 \right] \]

\[ \text{M}_{\text{HSM}}(p_i(c)) = \begin{cases} 10^{-x} & p_i(c) > \alpha \\ -\log(1-x) & p_i(c) \leq \alpha \end{cases} \]

\( \lambda_{\text{coord}}, \lambda_{\text{noord}} \) represent the weight of the corresponding loss, \( s^2 \) represents the total number of grids in the image, \( B \) represents the total number of predicted bounding box in each grid, \( \rho_{ij} = 1 \) represents that the \( j \)th predicted bounding box is effective detection in \( i \)th grid. \( \lambda_{\text{coord}}, \lambda_{\text{noord}} \) is assigned 1 in our network.

Since one stage network without region proposal network (RPN) is usually plagued by overwhelming negative samples, a hard sample mining strategy is added in our classification loss. The improved classification loss is shown below.

\( \alpha \) represents a constant that is usually equal to the score of vehicle determination. When a sample is an emblematic negative sample, \( p_i(c) \) is close to 0, and this emblematic negative sample leads to a small effect on the back propagation process. When a sample is a hard sample, \( p_i(c) \) is close to the score of vehicle determination. \( \text{M}_{\text{HSM}} \) obtains a high value, which improves the effect on the back propagation process. Therefore, the hard sample mining strategy filters the impact of a large number of negative samples on classification, and at the same time strengthens the impact of hard samples on classification.
4. Experiments and results

4.1. Dataset

4.1.1. Brief introduction. The increase of datasets has been one of the driving forces for the rapid development of object detection algorithms in recent years. Commonly used datasets, such as VOC [34] and COCO [35], not only provide data for various algorithms, but also a baseline for performance comparisons.

Comparing with the images contained in datasets designed for general object detection or image classification, aerial images have special characteristics. It is unlikely to train vehicle detection algorithms for aerial images only based on datasets designed for general object detection algorithms or image classification algorithms. As a result, several aerial images datasets, e.g. RSOD [36], INRIA [37], UCAS-AOD [19], VEDAI [18], DOTA [20] are announced as vehicle detection in aerial images has developed significantly recently. In this paper, three commonly used datasets designed for vehicle detection in aerial images, i.e. UCAS-AOD, VEDAI and DOTA, are used in the evaluation section. Since Images form theses three datasets are taken by UAVs or satellites from high altitudes, the sizes of the targets in these three datasets vary while the imaging angle of these targets remains the same.

4.1.2. UCAS-AOD datasets. The UCAS-AOD Dataset is proposed in [19] in 2015. The images contained in UCAS-AOD are selected from Google earth, and labelled by Patterns and Intelligent System Development Laboratory in the University of China Academy of Sciences. This dataset contains two categories of targets: vehicles, aircraft. Some details of the dataset are shown in Table 4. All the targets labelled as vehicles are used in the evaluation process.

4.1.3. VEDAI datasets. The VEDAI dataset is proposed in [18] in 2015. The images of the VEDAI dataset were taken from the satellite in the spring of 2012. The authors manually selected 1210 1024 × 1024 images from immense original images. All of these images were taken at the same height and the same shooting angle from the sky. Selected images have a variety of backgrounds, e.g. fields, grass, mountains, urban area, etc. As shown in Table 5, the VEDAI dataset contains nine different categories of vehicles: car, pick-up, truck, plane, boat, camping car, tractor, van, and others. These categories are divided into two meta-classes. The “small land vehicles” class consists of the “car”, “pick-up”, “tractor” and “van” categories.

| Class name | Total | Meta-class name | Total |
|------------|-------|----------------|-------|
| Pickup     | 950   | Small Land Vehicles | 2950   |
| Tractor    | 190   |                |       |
| Vans       | 100   | Large Land Vehicles | 690   |
| Car        | 1340  |                |       |
| Truck      | 300   |                |       |
| Camping Car| 390   |                |       |
| Plane      | 47    |                |       |
| Others     | 200   |                |       |
| Boat       | 170   |                |       |

Table 4. UCAS-AOD dataset categories information.

Table 5. VEDAI dataset categories information.

https://doi.org/10.1371/journal.pone.0250782.t004

https://doi.org/10.1371/journal.pone.0250782.t005
“large land vehicles” class consists of the “truck” and “camping car” categories. In this paper, all the objects from the “small land vehicles” meta-category are used in the evaluation process.

4.1.4. DOTA datasets. The DOTA dataset is proposed in [20] in 2017. Most images in DOTA-v1.5 are collected from Google Earth, while the others are taken by Satellite JL-1 and Satellite GF-2. The resolutions of images contained in the DOTA dataset various from 800×800 to 4000×4000. As shown in Table 6, the dataset is labelled with 16 categories, namely “plan”, “ship”, “storage tank”, “baseball diamond”, “tennis court”, “basketball court”, “ground track field”, “harbor”, “bridge”, “large, small vehicle”, “helicopter”, “roundabout”, “soccer ball field” and “swimming pool”. In previously published researches of vehicle detection in aerial images, the DOTA dataset is mainly used in three different ways. Some algorithms, e.g. [21, 38], are evaluated using targets form “small vehicle” only. Some algorithms, e.g. [39, 40], are evaluated using targets from both “small vehicle” and “large vehicle”. Others, e.g. [11, 41], are evaluated using targets from “small vehicle”, “large vehicle”, “plane”, and “boat”.

Since targets in some categories, e.g. “baseball diamond”, “tennis court”, “basketball court”, “harbor”, “bridge”, are irrelevant to the theme of this paper, only the targets labelled as “small vehicle”, “ship” and “plane” are used in this partition to evaluate the performance of the proposed algorithm on targets with various scales. In these three categories, small vehicles have relatively stable scales (from 2×9 to 89×124), planes have various scales (from 9×8 to 856×842) but distinctive shapes, while the scales of ships various rapidly (from 5×12 to 939×1750).

4.2. Training detail and preprocessing on dataset

The proposed FFDP-CNN is an end to end one stage algorithm. It is implemented on the TensorFlow 2.0 deep learning framework, trained and evaluated using NVIDIA GeForce RTX 2080Ti with 11G memory. The weight decay and the momentum are set to 1e⁻⁴ and 0.9 respectively. In the experiments, the learning rate is set to 1e⁻³ initially, and gradually decades to 1e⁻⁴ and 1e⁻⁵.

Images from the three datasets are pre-processed before fed into the network. Through the sliding window, padding, and stride mechanism, all datasets are processed into images with a resolution of 640×640. Besides, the annotations of all datasets are optimized. The optimization

| Class name               | Total |
|-------------------------|-------|
| Large-vehicle           | 11552 |
| Small-vehicle           | 12841 |
| Harbor                  | 5723  |
| Ship                    | 23467 |
| Ground track field      | 398   |
| Soccer ball field       | 407   |
| Baseball diamond        | 495   |
| Swimming pool           | 986   |
| Roundabout              | 417   |
| Tennis court            | 2044  |
| Basketball              | 353   |
| Plane                   | 6629  |
| Helicopter              | 196   |
| Bridge                  | 1227  |
| Storage tank            | 7636  |

https://doi.org/10.1371/journal.pone.0250782.t006
process corrects the vehicle labels if targets are obviously missing. This is a common dataset optimization method that is also used in [21, 27].

4.3. Evaluation method

To evaluate the performance of the proposed framework, Average Precision (AP), mean Average Precision (mAP) and F1 are used in the experiments.

4.3.1 P-R curve, AP and mAP. The main purpose of object detection is to find out targets and classify them into various categories. The evaluation indicators for these two tasks are "the proportion of the correct targets detected to all targets" and "the correct rate of the classification of detected targets", which can be represented by recall(abbreviated as R) and precision (abbreviated as P). The these two measurements are defined using true positive(TP), false positive (FP), true negative (TN), and false negative (FN).

The precision is calculated as shown in formula (7):

\[
P = \frac{TP}{TP + FP},
\]

(7)

The recall is calculated as shown in formula (8):

\[
R = \frac{TP}{TP + FN}.
\]

(8)

When comparing the performance of different frameworks, the precision and recall indicators sometimes appear contradictory. Moreover, it is necessary to measure the accuracy of a detector with a single index. Therefore, AP(average precision) is proposed in [34]. P and R are sorted according to score, and draw a rectangular coordinate graph with P and R as the coordinates. This rectangular graph is called P-R curve. The area under the P-R curve is AP, and mAP (mean average precision) is the average of APs of multiple categories. AP measures the quality of the detector in each category, while mAP measures the quality of the detector in all categories. As a result, in this paper, AP is used when targets are from only one category (i.e. evaluations using UCAS-AOD and VEDIA datasets), and mAP is used when targets are from multiple categories (i.e. evaluation using DOTA dataset).

4.3.2 F-Measure. F-Measure (also known as F-Score) is another commonly used indicator for object detectors. The F-Measure is a weighted average of Precision(P) and Recall(R):

\[
F = \frac{(\alpha^2 + 1)P \times R}{\alpha^2(P + R)},
\]

(9)

When \(\alpha\) equal 1, F-Measure can be called F1:

\[
F = \frac{2 \times P \times R}{P + R},
\]

(10)

F1 combines the results of P and R. when F1 is higher, it means that the detector is more effective. The P and R values of the proposed framework on various datasets are also provided for comprehensive evaluation.

4.4. Results on UCAS-AOD dataset

The effectiveness of the proposed framework has been validated on the UCAS-AOD dataset. The APs of the proposed FFDP-CNN detector and other 18 state-of-the-art vehicle detection algorithms for aerial images are shown in Table 7. It can be seen that the proposed FFDP-CNN detector achieves an AP of 97.34% which is roughly 1.2% higher than the APs achieved by
The P-R curve and the detailed performance of the proposed FFDP-CNN detector are shown in Fig 6 and Tables 7, 8 respectively.

To demonstrate the effectiveness of the proposed stepwise res-block and feature fusion deep-projection module, two other algorithms (FFDP-CNN with Res2Net and FPN with Stepwise Res-block) have been evaluated, and the evaluation results are shown in Table 8.

### 4.4.1 Effectiveness of stepwise res-block.
Compared with the proposed FFDP-CNN (which utilizes Stepwise Res-block and FFDP), FFDP-CNN with Res2Net utilizes Res2Net blocks to build its backbone while other parts remain the same with FFDP-CNN. In other words, the only difference between FFDP-CNN and FFDP-CNN with Res2Net is the res-block used in their backbone. As a result, the effectiveness of the proposed Stepwise Res-block can be demonstrated by comparing the evaluation results of FFDP-CNN and FFDP-CNN with Res2Net. According to the evaluation results shown in Table 8, FFDP-CNN with Res2Net achieves an AP of 95.38% which is 1.96% lower than the AP obtained by FFDP-CNN. In other words, comparing with Res2Net, Stepwise res-block gains an advantage of 1.96% AP in the evaluation.

### 4.4.2 Effectiveness of feature fusion deep projection module.
Comparing with FFDP-CNN, FPN with Stepwise Res-block utilizes FPN instead of feature fusion and deep-projection module (FFDP), while other parts remain the same. As a result, the effectiveness of the proposed FFDP can be demonstrated by comparing the evaluation results of FFDP-CNN and FFDP-CNN with Res2Net. From Table 8, it can be seen that compared with standard FPN, the feature fusion deep-projection module (FFDP) achieves a solid performance advantage of 2.48% AP.

### 4.4.3 Parameter number and processing speed.
To demonstrate the efficiency of the proposed stepwise res-block, the parameter number of the proposed FFDP-CNN and other algorithms are shown in Table 9. It can be seen that FFDP-CNN has less parameter comparing...
with other state-of-the-art algorithms (i.e. YOLO, Faster RCNN and An Improved FBPN). Because of the complexity of the proposed feature fusion and deep-projection module, the processing speed of the proposed FFDP-CNN for 640×640 color images is 14.01 FPS on an NVIDIA 2080Ti GPU.

4.4.4 Performance analysis. The detection results of the proposed FFDP-CNN are analyzed manually to get a comprehensive understanding of its performance. In the detection results, most targets can be found correctly regardless of their orientations, colours and scales, which also demonstrated the effectiveness of the proposed framework. However, the proposed algorithm sometimes fails when parts of the targets are blocked by trees or building roofs in the input images, especially when the blocked parts are more than half of the original targets.

4.5. Results on VEDAI dataset

The proposed FFDP-CNN is trained and evaluated on objects labelled as “small land vehicles” in VEDAI dataset. The APs of the proposed detector and other 13 state-of-the-art algorithms

| Method                  | R  | P  | AP(%)  | F1-Score |
|-------------------------|----|----|--------|----------|
| FPN with Stepwise Res-block | 91.9 | 91.0 | 94.86  | 91.48    |
| FFDP-CNN with Res2Net   | 93.4 | 91.1 | 95.38  | 92.27    |
| FFDP-CNN                | 93.8 | 93.1 | 97.34  | 93.42    |

Table 8. Comparison between proposed framework and other methods on UCAS-AOD.

https://doi.org/10.1371/journal.pone.0250782.g006

https://doi.org/10.1371/journal.pone.0250782.t008
are shown in Table 10. The proposed FFDP-CNN detector achieves an AP of 93.96%, outperforms other state-of-the-art networks designed for vehicle detection. Compared with DFL [11], the AP has been improved by 3.42%. Compared with Improved FBPN-based Detection Network [38], the AP has been improved by 2.69%. The P-R curve and the detailed performance of the proposed FFDP-CNN detector are shown in Fig 7 and Table 11 respectively.

4.6. Results on DOTA dataset

The proposed framework has also been evaluated by the DOTA dataset. The evaluation results of the proposed FFDP-CNN detector and other 16 state-of-the-art algorithms are shown in Table 12. In general, it can be seen that the proposed framework achieves an mAP of 89.86% which is higher than the mAPs obtained by other state-of-the-art algorithms. The P-R curves of the proposed FFDP-CNN algorithm based on the DOTA dataset are shown in Fig 8. Table 13 demonstrates the detailed evaluation results of the proposed framework. Table 12 also demonstrate that the proposed FFDP-CNN achieves the best performance when detecting small vehicles (with an AP of 89.78%), and the second-best performance when detecting planes and ships (with APs of 91.33% and 88.48% respectively). As a result, comparing to other algorithms, the performance of FFDP-CNN can maintain at a relatively stable level when the scales of the targets various rapidly.

### Table 9. Comparison of parameter numbers and other information.

| Method             | Input size | Model size | Parameter Number |
|--------------------|------------|------------|------------------|
| YOLO V2 [11]      | 608×608    | 255 MB    | 67 M             |
| YOLO V3 [11]      | 608×608    | 235 MB    | 61 M             |
| Faster R-CNN [11] | 608×608    | 253 MB    | 59 M             |
| RetinaNet [11]    | 608×608    | 146 MB    | 36 M             |
| AVDNet [11]       | 608×608    | 53 MB     | 13 M             |
| An Improved FBPN [38] | 600×600    | 479 MB    | 61 M             |
| Ju.M et.al [21]   | 512×512    | 2.8 MB    | -                |
| FFDP-CNN          | 640×640    | 144 MB    | 37 M             |

https://doi.org/10.1371/journal.pone.0250782.t009

### Table 10. Comparison between proposed framework and other methods on VEDAI.

| Method                  | AP(%)  |
|-------------------------|--------|
| AVDNet 2019 [11]        | 51.95  |
| VDN 2017 [41]           | 54.6   |
| DPM 2015 [55]           | 60.5   |
| R³-Net (R + F) 2019 [56] | 69.0  |
| Faster-RCNN 2017 [55]   | 70.9   |
| Improved Faster RCNN 2017 [43] | 74.30 |
| Ju, et al. 2019 [21]    | 80.16  |
| YOLOv3_Joint-SRVDNet 2020 [57] | 80.4 |
| Faster RER-CNN 2018 [55] | 83.5  |
| YOLOv3_HR [57]          | 85.66  |
| Faster RCNN + Res2Net (resnet101) 2019 [58] | 81.96 |
| Faster RCNN + WaterFall (resnet50) 2019 [28] | 77.36 |
| Improved FBPN-Based Detection Network [38] | 91.27 |
| FFDP-CNN               | 93.96  |

https://doi.org/10.1371/journal.pone.0250782.t010
5. Conclusion and future works

In this paper, a Feature Fusion Deep-Projection Convolution Neural Network (FFDP-CNN) is proposed to detect vehicles in aerial images. The main contribution of this research has three aspects. Firstly, a novel residual block named stepwise res-block is designed in this paper.

![Fig 7. The P-R curve of our framework on VEDAI.](https://doi.org/10.1371/journal.pone.0250782.g007)

| Method               | R (%) | P (%) | AP(%) | F1-Score |
|----------------------|-------|-------|-------|----------|
| FFDP-CNN             | 94.9  | 90.5  | 93.96 | 92.67    |

Table 11. F1-Measure on VEDAI.

![Table 11. F1-Measure on VEDAI.](https://doi.org/10.1371/journal.pone.0250782.t011)

| Method                        | AP(%) (Ship) | AP(%) (Plane) | AP(%) (Small Vehicle) | mAP(%)   |
|-------------------------------|--------------|---------------|-----------------------|----------|
| L-RCNN 2020 [39]             | -            | -             | 56.09                 | -        |
| Yang et al. 2018 [38]        | -            | -             | 61.16                 | -        |
| the Light-Head R-CNN OBB+W/FPN [59] | 83.59        | 88.64         | 68.81                 | 80.35    |
| Faster RCNN Adapted 2018 [60] | 75.77        | 88.02         | 70.15                 | 77.98    |
| DYOLO Module B 2018 [60]     | 88.7         | 88.2          | 76.0                  | 84.3     |
| SSD Adapted2018 [60]         | 85.0         | 88.2          | 76.3                  | 83.17    |
| DFRCNN 2018 [61]             | -            | -             | 76.5                  | -        |
| DSSD 2017 [62]               | 87.5         | 91.1          | 79.0                  | 85.87    |
| DYOLO Module A 2018 [60]     | 87.8         | 86.6          | 79.2                  | 84.53    |
| RefineDet 2018 [60]          | 87.5         | 94.0          | 80.0                  | 87.17    |
| Ju, et al. 2019 [21]         | -            | -             | 88.63                 | -        |
| Improved FBPN-Based Detection Network [38] | -       | -             | 88.76                 | -        |
| FFDP-CNN                     | 88.48        | 91.33         | 89.78                 | 89.86    |

Table 12. Comparison between proposed framework and other methods on DOTA.

https://doi.org/10.1371/journal.pone.0250782.t012
Thanks to its special hierarchical structure, the output of the stepwise res-block contains features processed by 0–3 convolutional layers. Besides, the parameter number keeps at a relatively low level. Secondly, based on the proposed stepwise res-block, the backbone of the proposed FFDP-CNN is designed. By composing 34 stepwise res-blocks, the proposed backbone produce features processed by 0–102 3x3 convolutional layers. It can explore the high-level semantic feature and conserve the low-level detail feature at the same time. Last but not least, a feature fusion module is used to mix features generated from different levels of the backbone, and further balance the low-level and high-level features. A special designed deep-projection deconvolution module is utilized in the proposed feature fusion module to reduce the impact of information contamination introduced by downsampling/upsampling processes. According to the evaluation results, the proposed FFDP-CNN outperforms other state-of-the-art algorithms on UCAS-AOD, VEDAI and DOTA datasets. According to the detection results, although the proposed FFDP-CNN can detect targets regardless of their orientations,

Table 13. F1-Measure on DOTA.

| Method                  | R     | P     | mAP(%) | F1-Score |
|-------------------------|-------|-------|--------|----------|
| FFDP-CNN on Ship        | 87.37 | 91.42 | 88.48  | 89.35    |
| FFDP-CNN on Plane       | 90.94 | 93.07 | 91.33  | 91.99    |
| FFDP-CNN on Small Vehicle | 89.0  | 91.1  | 89.78  | 90.0     |

https://doi.org/10.1371/journal.pone.0250782.t013
colours and scales, the ability to detect partially blocked targets still can be improved. In the future work, GANs will be used to generate more partially blocked training data to train the proposed framework to detect partially blocked targets, and attention mechanisms will be implemented in the stepwise res-block to further increase the detection accuracy.

**Author Contributions**

**Conceptualization:** Bin Xu.

**Data curation:** Bin Xu.

**Formal analysis:** Bin Xu.

**Funding acquisition:** Bin Wang.

**Investigation:** Bin Xu.

**Methodology:** Bin Xu.

**Project administration:** Bin Wang.

**Resources:** Bin Wang.

**Software:** Bin Xu.

**Supervision:** Bin Wang.

**Validation:** Bin Wang.

**Visualization:** Bin Xu.

**Writing – original draft:** Bin Xu.

**Writing – review & editing:** Bin Wang.

**References**

1. Cheng Y T, Patel A, Wen C, et al. Intensity thresholding and deep learning based lane marking extraction and lane width estimation from mobile light detection and ranging (LiDAR) point clouds[J]. Remote Sensing, 2020, 12(9): 1379.

2. Yang M Y, Liao W, Li X, et al. Vehicle detection in aerial images[J]. Photogrammetric Engineering & Remote Sensing, 2019, 85(4): 297–304.

3. Feng R, Fan C, Li Z, et al. Mixed road user trajectory extraction from moving aerial videos based on convolution neural network detection[J]. IEEE Access, 2020, 8: 43508–43519.

4. Li W, Li H, Wu Q, et al. Simultaneously detecting and counting dense vehicles from drone images[J]. IEEE Transactions on Industrial Electronics, 2019, 66(12): 9651–9662.

5. Haralick R M, Shanmugam K, Dinstein I H. Textural features for image classification[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1973 (6): 610–621.

6. Lowe D G. Object recognition from local scale-invariant features[C]/Proceedings of the seventh IEEE International Conference on Computer Vision. IEEE, 1999, 2: 1150–1157.

7. Moranduzzo T, Melgani F. Automatic car counting method for unmanned aerial vehicle images[J]. IEEE Transactions on Geoscience and Remote Sensing, 2013, 52(3): 1635–1647.

8. Qi S, Ma J, Lin J, et al. Unsupervised ship detection based on saliency and S-HOG descriptor from optical satellite images[J]. IEEE Geoscience and Remote Sensing Letters, 2015, 12(7): 1451–1455.

9. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]/Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 580–587.

10. Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]/Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 779–788.

11. Mandal M, Shah M, Meena P, et al. AVDNet: A small-sized vehicle detection network for aerial visual data[J]. IEEE Geoscience and Remote Sensing Letters, 2019, 17(3): 494–498.
12. Koga Y, Miyazaki H, Shibasaki R. A CNN-based method of vehicle detection from aerial images using hard example mining[J]. Remote Sensing, 2018, 10(1): 124.

13. Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 39(6): 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031 PMID: 27295650

14. He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 2961–2969.

15. Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

16. Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European Conference on Computer Vision. Springer, Cham, 2016: 21–37.

17. Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. 2018: 3974–3983.

18. Razakarivony S, Jurie F. Vehicle detection in aerial imagery: A small target detection benchmark[J]. Journal of Visual Communication and Image Representation, 2016, 34: 187–203.

19. Zhu H, Chen X, Dai W, et al. Orientation robust object detection in aerial images using deep convolutional neural network[C]/2015 IEEE International Conference on Image Processing (ICIP). IEEE, 2015: 3735–3739.

20. Xia G S, Bai X, Ding J, et al. DOTA: A large-scale dataset for object detection in aerial images[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 8759–8768.

21. Yang M Y, Liao W, Li X, et al. Deep learning for vehicle detection in aerial images[C]/2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018: 3079–3083.

22. Tang T, Zhou S, Deng Z, et al. Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining[J]. Sensors, 2017, 17(2): 336.

23. Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 8759–8768.

24. Yang M Y, Liao W, Li X, et al. Deep learning for vehicle detection in aerial images[C]/2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018: 3079–3083.

25. Cheng G, Han J, Zhou P, et al. Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection[J]. IEEE Transactions on Image Processing, 2018, 28(1): 265–278.

26. Liu W, Ma L, Wang J. Detection of multiclass objects in optical remote sensing images[J]. IEEE Geoscience and Remote Sensing Letters, 2018, 16(5): 791–795.

27. Qiu H, Li H, Wu Q, et al. A2RMNet: Adaptively aspect ratio multi-scale network for object detection in remote sensing images[J]. Remote Sensing, 2019, 11(13): 1594.

28. Artacho B, Savakis A. Waterfall atrous spatial pooling architecture for efficient semantic segmentation [J]. Sensors, 2019, 19(24): 5361.

29. Yang Y, Li G, Wu Z, et al. Reverse perspective network for perspective-aware object counting[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2020: 4374–4383.

30. Rabbi J, Ray N, Schubert M, et al. Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network[J]. Remote Sensing, 2020, 12(9): 1432. https://doi.org/10.3390/rs12091432

31. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770–778.

32. Haris M, Shakhrnovich G, Ukita N. Deep back-projection networks for super-resolution[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 1664–1673.

33. Tian Y, Yang G, Wang Z, et al. Apple detection during different growth stages in orchards using the improved YOLO-V3 model[J]. Computers and Electronics in Agriculture, 2019, 157: 417–426.

34. Everingham M, Van Gool L, Williams CK I, et al. The pascal visual object classes (voc) challenge[J]. International Journal of Computer Vision, 2010, 88(2): 303–338.

35. Bell S, Zitnick CL, Bala K, et al. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2874–2883.

36. Long Y, Gong Y, Xiao Z, et al. Accurate object localization in remote sensing images based on convolutional neural networks[J]. IEEE Transactions on Geoscience and Remote Sensing, 2017, 55(5): 2486–2498.

37. Maggiori E, Tarabalka Y, Charpiat G, et al. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark[C]/2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2017: 3226–3229.
38. Wang B, Gu Y. An Improved FBPN-Based Detection Network for Vehicles in Aerial Images[J]. Sensors, 2020, 20(17): 4709.
39. Liao W, Chen X, Yang J, et al. LR-CNN: Local-aware Region CNN for Vehicle Detection in Aerial Imagery[J]. arXiv preprint arXiv:2005.14264, 2020.
40. Majid Azimi S, Bahmanyar R, Henry C, et al. EAGLE: Large-scale Vehicle Detection Dataset in Real-World Scenarios using Aerial Imagery[J]. arXiv e-prints, 2020: arXiv:2007.06124.
41. Mandal M, Shah M, Meena P, et al. Sssdet: Simple short and shallow network for resource efficient vehicle detection in aerial scenes[C]/2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019: 3098–3102.
42. Zheng K, Wei M, Sun G, et al. Using vehicle synthesis generative adversarial networks to improve vehicle detection in remote sensing images[J]. ISPRS International Journal of Geo-Information, 2019, 8(9): 390.
43. Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]/Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 7263–7271.
44. Yang X, Sun H, Fu K, et al. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks[J]. Remote Sensing, 2018, 10(1): 132.
45. Liu L, Pan Z, Lei B. Learning a rotation invariant detector with rotatable bounding box[J]. arXiv preprint arXiv:1711.09405, 2017.
46. Wei H, Zhang Y, Chang Z, et al. Oriented objects as pairs of middle lines[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2020, 169: 268–279.
47. Zhou L, Wei H, Li H, et al. Objects detection for remote sensing images based on polar coordinates[J]. arXiv preprint arXiv:2001.02988, 2020.
48. Dai J, Li Y, He K, et al. R-Ifcn: Object detection via region-based fully convolutional networks[J]. arXiv preprint arXiv:1605.06409, 2016.
49. Dai J, Qi H, Xiong Y, et al. Deformable convolutional networks[C]/Proceedings of the IEEE International Conference on Computer Vision. 2017: 764–773.
50. Bao S, Zhong X, Zhu R, et al. Single shot anchor refinement network for oriented object detection in optical remote sensing imagery[J]. IEEE Access, 2019, 7: 87150–87161.
51. Li C, Xu C, Cui Z, et al. Feature-attentioned object detection in remote sensing imagery[C]/2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019: 3886–3890.
52. Yang X, Liu Q, Yan J, et al. R3det: Refined single-stage detector with feature refinement for rotating object[J]. arXiv preprint arXiv:1908.05612, 2019.
53. Yang X, Yan J, Yang X, et al. Scred++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing[J]. arXiv preprint arXiv:2004.13316, 2020.
54. Azimi S M, Vig E, Bahmanyar R, et al. Towards multi-class object detection in unconstrained remote sensing imagery[C]/Asian Conference on Computer Vision. Springer, Cham, 2018: 150–165.
55. Terrail J O, Jurie F. Faster RER-CNN: Application to the detection of vehicles in aerial images[J]. arXiv preprint arXiv:1809.07628, 2018.
56. Li Q, Mou L, Xu Q, et al. R3-net: A deep network for multi-oriented vehicle detection in aerial images and videos. arXiv 2018[8]. arXiv preprint arXiv:1808.05560.
57. Mostofa M, Ferdous S N, Riggan B S, et al. Joint-srsvdnet: Joint super resolution and vehicle detection network[J]. IEEE Access, 2020, 8: 82306–82319.
58. Gao S H, Cheng M M, Zhao K, et al. Res2Net: A New Multi-scale Backbone Architecture[J]. arXiv e-prints, 2019: arXiv:1904.01169.
59. Ding J, Xue N, Long Y, et al. Learning roi transformer for oriented object detection in aerial images[C]/Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 2849–2858.
60. Acatay O, Sommer L, Schumann A, et al. Comprehensive evaluation of deep learning based detection methods for vehicle detection in aerial imagery[C]/2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2018: 1–6.
61. Sommer L, Schumann A, Schuchert T, et al. Multi feature deconvolutional faster r-cnn for precise vehicle detection in aerial imagery[C]/2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2018: 635–642.
62. Fu C Y, Liu W, Ranga A, et al. Dssd: Deconvolutional single shot detector[J]. arXiv preprint arXiv:1701.06659, 2017.