

Article

End-to-End Airplane Detection Using Transfer Learning in Remote Sensing Images

Zhong Chen ^{1,2,3}, Ting Zhang ^{1,2,3} and Chao Ouyang ^{1,2,3,*}

¹ School of Automation, Huazhong University of Science and Technology, Luoyu Road 1037, Wuhan 430074, China; henpacked@hust.edu.cn (Z.C.); douyaer2020@hust.edu.cn (T.Z.)

² National Key Laboratory of Science and Technology on Multi-spectral Information Processing, Luoyu Road 1037, Wuhan 430074, China

³ Key Laboratory of Ministry of Education for Image Processing and Intelligent Control, Luoyu Road 1037, Wuhan 430074, China

* Correspondence: ouyangchao16@hust.edu.cn; Tel.: +86-138-7110-5370

Received: 30 November 2017; Accepted: 15 January 2018; Published: 18 January 2018

Abstract: Airplane detection in remote sensing images remains a challenging problem due to the complexity of backgrounds. In recent years, with the development of deep learning, object detection has also obtained great breakthroughs. For object detection tasks in natural images, such as the PASCAL (Pattern Analysis, Statistical Modelling and Computational Learning) VOC (Visual Object Classes) Challenge, the major trend of current development is to use a large amount of labeled classification data to pre-train the deep neural network as a base network, and then use a small amount of annotated detection data to fine-tune the network for detection. In this paper, we use object detection technology based on deep learning for airplane detection in remote sensing images. In addition to using some characteristics of remote sensing images, some new data augmentation techniques have been proposed. We also use transfer learning and adopt a single deep convolutional neural network and limited training samples to implement end-to-end trainable airplane detection. Classification and positioning are no longer divided into multistage tasks; end-to-end detection attempts to combine them for optimization, which ensures an optimal solution for the final stage. In our experiment, we use remote sensing images of airports collected from Google Earth. The experimental results show that the proposed algorithm is highly accurate and meaningful for remote sensing object detection.

Keywords: airplane detection; end to end; transfer learning; convolutional neural networks

1. Introduction

Object detection in remote sensing images is important for civil and military applications, such as airport surveillance and inshore ship detection. With the rapid development of high-resolution satellites, high-resolution remote sensing image data increased dramatically, providing the possibility for developing a more intelligent object detection system in remote sensing images. Aircraft detection in remote sensing images is a typical problem of small target recognition under a wide range. Although it has been studied for years [1,2], most of those methods show low efficiency of large-area airplane detection and are often limited by a lack of ability to apply them to other objects. In the face of complex and various object conditions, it is an important and urgent problem to be solved efficiently and to detect specific targets accurately in object detection applications. In this paper, we mainly focus on airplane detection around airports, which means that we assume the airport has been located already by other methods.

In recent years, almost all technologies with outstanding performance in object detection are based on deep convolutional neural networks, which is attributed to the success of AlexNet [3] in the

ImageNet [4] Large Scale Visual Recognition Challenge in 2012, which demonstrated that features extracted by the convolutional neural networks are more robust than hand-crafted features, such as SIFT (Scale-invariant Feature Transform) [5] and HOG (Histogram of Oriented Gradient) [6]. In the following years, GoogLeNet [7], VGG (Visual Geometry Group) [8], and ResNet [9] base networks were designed and greatly improved the accuracy of image classification. Different from image classification, object detection not only needs to identify the object category, but also needs to give the location of the object. In 2014, R-CNN (Regions with CNN features) [10] applied convolutional neural networks in the field of object detection, and continuously made great breakthroughs in this field with great improvement in detection accuracy and speed. At present, the object detection based on deep learning can be mainly divided into two categories. One is the two-stage object detection framework combining region proposal and CNN classification, which is represented by R-CNN, including SPP-NET [11], Fast R-CNN [12], and Faster R-CNN [13]. The other is the object detection framework with a single stage. Using a single convolutional neural network, the object detection problem is transformed into a regression problem, which is represented by YOLO (You Look Only Once) [14] and SSD (Single Shot MultiBox Detector) [15].

Numerous studies have proved that object detection frameworks based on deep learning are not only feasible, but also have very good detection effects on natural images. Most of them are highly ranked in major object detection competitions, such as PASCAL VOC [16] and COCO (Common Objects in Context) [17]. However, few people directly study deep-learning-based object detection in remote sensing images. The primary reason for this is that there is a lot of labeled data in natural images, such as ImageNet, and it is also used widely in natural images. Therefore, deep learning in natural images has developed even faster. Unlike natural images, remote sensing images have some features that natural images do not:

1. Resolution information often is given by the remote sensing images, so the size of the object in the image can be inferred based on some prior knowledge, which is crucial for the object detection task.
2. The observed field of view changes very slightly. Natural images from different perspectives will have a great difference, while all the remote sensing images are obtained from a top-down view, which also makes the visual changes of object usually minimally severe.
3. The object in the remote sensing image is, generally, relatively small when compared with the background, but the current small object detection is not well solved in the natural image. Therefore, improvements still need to be made when applying the detection algorithm on natural images to determine object detection in remote sensing images.

In summary, it can be considered that, in natural images, object detection based on deep learning develops rapidly due to the large amount of data that is annotated. It is also possible to better use metrics to evaluate algorithm performance due to many publicly-available large datasets. However, the data acquisition in remote sensing images is relatively difficult, the standard datasets are relatively few and the application scope is also small. The above reasons are why the research of object detection in remote sensing images lags behind that of natural images. In theory, both natural and remote sensing images are pixel matrices, so this does not affect the application of the deep learning framework in natural images to remote sensing images. The contribution of this paper includes the following points:

1. We collected samples of the airplanes from Google Earth and labeled them manually. In addition to the common data augmentation operations in natural images, we also add rotation and other operations.
2. Using the idea of transfer learning and a limited number of airplane samples for training, an end-to-end airplane detection framework is achieved, as shown in Figure 1.
3. A method is proposed to solve the size restrictions of the input images, which first divides the image into blocks and then detects the airplane.

This paper will be elaborated with the following sections: Section 2 introduces object detection techniques based on deep learning and airplane detection in remote sensing images. Section 3 describes our method in detail. Section 4 presents the experimental results and related analyses. Section 5 provides a summary of the paper.

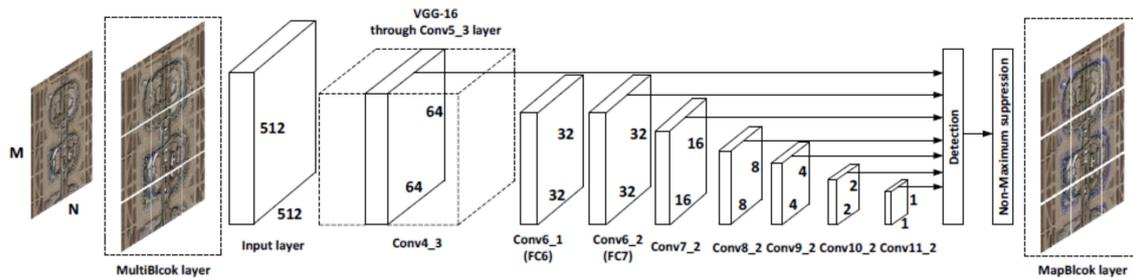


Figure 1. Airplane detection framework. We add a MultiBlock layer and MapBlock layer based on the SSD (Single Shot MultiBox Detector). After the MultiBlock layer, each block is resized to 512×512 . As is shown, we use VGG (Visual Geometry Group) 16 as the base network. The prediction result of each block is mapped back to the original image after passing the MapBlock layer.

2. Related Work

In this section, we will review some techniques of object detection based on deep learning and the related work on airplane detection; this section will also introduce the transfer learning used in this paper. These algorithms are very prominent in the object detection tasks of natural images. Some algorithms have high precision, while others win by speed. However, there are some defects that cannot be neglected when the algorithms are applied to remote sensing images.

2.1. Object Detection with CNNs

The series of R-CNN algorithms have become one of the mainstream techniques in the field of object detection, which benefits from the success of these algorithms in competitions such as PASCAL VOC, and COCO. Its idea, region proposal + CNNs, is very easy to accept. The R-CNN series mainly includes Fast R-CNN and Faster R-CNN. Due to the lack of object size information, a sliding window search in the early stage is unable to determine the size and search scope of the sliding window when detecting objects in natural images. This results in extremely low efficiency. The sliding window algorithm still has high costs, even in remote sensing images with known resolution. Adopting more effective region proposal algorithms can allow for using more sophisticated classifiers, which is widely applied in mainstream object detectors [10–12]. This may also improve detection quality by reducing false positives [18] due to the difference in the number of detection windows. The selective search algorithm used by R-CNN is an alternative to the sliding window algorithm, which maintains a high recall rate while calculating at high speed. The variants of the original R-CNN algorithm, SPP-NET, and Fast R-CNN, inherit this idea. However, the traditional region proposal methods do not perform well on data with more complicated backgrounds. Most of them, such as selective search, cannot take advantage of the computational efficiency of the GPU. Additionally, the parameters of these methods are hard to choose, and are usually determined through a combination of trial and error, and experience. Settings that work well for one image may not work at all for another. The detection speed and recall rate are improved further until the Faster R-CNN automatically extracts the region proposal using the RPN network.

Different from the R-CNN series, the YOLO series adopts the detection method with one stage, which is skipping the step of extracting the region proposal. It uses the method of dividing the input image into an $S \times S$ grid (YOLO) or setting the default box (SSD) to predict the object category and location directly, which further simplifies the training and detecting process. YOLO is very fast; it predicts based on the global information of the image, which is different from the object detection

algorithm based on region proposal. However, YOLO has poor prediction of the object position, and the detection effect on small objects and dense objects is not good. YOLO can reduce the probability of predicting the background as an object, but it also leads to a lower recall rate.

Our network framework is based on the SSD detection model. SSD combines Faster R-CNN's anchor with YOLO's single convolutional neural network. The difference is that the SSD sets anchors on multiple feature maps (called the default box in SSD), which locates the objects more accurately. At the same time, SSD output is a series of fixed-size bounding boxes defined in advance. It is superior to the method of extracting region proposals in terms of speed. Extracting features at different resolutions is also due to the different expression of convolutional neural networks at different layers, the top layer being closer to the semantic information and the bottom layer holding more details of the image. The combination of information in multiple layers can obtain a good detection effect for different object sizes.

Both the R-CNN series and YOLO series of object detection algorithms have requirements for the size of the input image. Faster R-CNN needs to fix the size of the shortest edge of the input image, while scaling the other side. YOLO and SSD both require a fixed size input, so the image will be scaled directly to the required input size. These direct scaling operations are certainly catastrophic for large remote sensing images with small objects. In response to the drawback, we added a MultiBlock layer before the input layer of the network in the testing phase. Specifically, based on the information of the remote sensing image, including width, height, and resolution, the image is divided into multiple blocks, and there is some overlap between the blocks to avoid the object not being detected due to being divided. Meanwhile, after the Detection Output Layer, a MapBlock layer is added to map the object detected in the block back to the position in the original image. It is a simple and effective idea that makes the prediction of large-sized images possible, while not losing detection precision.

2.2. Airplane Detection Method

The robustness of features extracted from convolutional neural networks has far exceeded that of manual design, and the features have achieved great success regarding object classification and detection in natural images. Many researchers have already applied these technologies to the related work of airplane detection in remote sensing images. Wu et al. [19] used a method based on BING [20] and CNN to detect airplanes. They extracted region proposals using BING and classified region proposals using CNNs. This is an R-CNN-based detection method (R-CNN uses the selective search). In the step of extracting region proposals, there are EdgeBoxes [21], CPMC (Constrained Parametric Min-Cuts) [22], MCG (Multiscale Combinatorial Grouping) [23], and Objectness [24], in addition to BING and selective search. Hosang et al. [18] analyzed the performance of these region proposal algorithms in detail and found that these algorithms have low repeatability; they are not robust to noise and disturbance.

Due to the perspective of remote sensing images, most of the objects are rotationally invariant. Training a rotational invariant classifier is crucial. To take advantage of this nature of airplane, Zhang et al. [25] proposed a method that uses extending histogram oriented gradients to obtain new rotationally-invariant features. Wang et al. [26] proposed a rotation-invariant matrix (RIM) to obtain the rotational invariance of features, which incorporates partial angular spatial information. Liu et al. [27] proposed a feature extraction method based on sparse coding for airplane detection. Although these algorithms can both obtain the rotational invariance of the airplane to a certain extent, and improve detection performance, they are not very scalable to other objects. We also take advantage of the rotational invariance in this paper, but we only use it as a means of data augmentation and then use convolutional neural networks to learn this property directly, just as in learning other features of the airplane.

2.3. Transfer Learning

Transfer learning is a research problem in machine learning that focuses on storing the knowledge gained in solving a problem and applying it to different but related problems [28]. Transfer learning is

a very important and effective technology in deep learning. The purpose of transfer learning is to not discard useful information from previous data and apply previously learned knowledge to solve new problems, which can solve problems faster and better.

Fine tuning is one of the most important tools in transfer learning. A large number of image classification and object detection experiments use fine tuning because of the lack of data in a specific task. Many have also experimentally confirmed that fine-tuning convolutional neural networks is better than training from scratch. For example, there have been object detection frameworks based on deep convolutional neural networks in recent years, in which the base network is taken as an important part, such as ZF, GoogLeNet, VGG, and ResNet. These pre-training models are obtained by being applied to image classification tasks. ImageNet is a large visual database used for studying visual object recognition software. Almost all base networks are trained on this dataset. The VGG network we use in our experiments is also trained on this dataset, and then fine-tuned using our dataset to enable this network to recognize the airplane. As shown in Figure 2, by transfer learning, we can achieve object detection in natural images or remote sensing images.

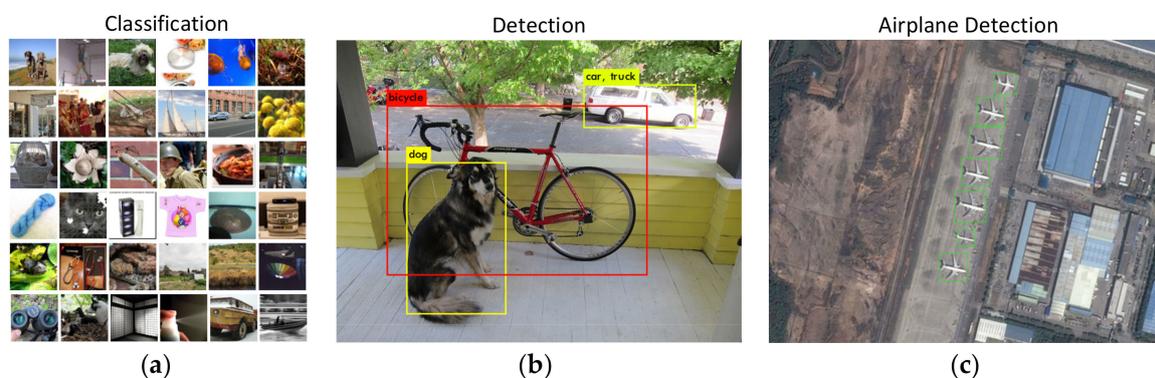


Figure 2. Transfer learning. The model trained by classification tasks can be used as our base network, and then the object detection framework in the natural image can be used to detect the airplane in the remote sensing image. (a) ImageNet dataset used for classification [4]; (b) the object detection result of a natural image, using YOLO (You Look Only Once); and (c) airplane detection in this paper.

3. Methods

3.1. Data

We did much data labeling work for training and testing. The data is collected from satellite images of the world's top 30 airports in Google Earth, such as Hartsfield-Jackson Atlanta International Airport (United States), Beijing Capital International Airport (China), and other airports. Most of them are city airports; the resolutions of airports may be different, ranging from 0.98 m to 10 m, and the frame size is between 2000×2000 and 8000×8000 . Fifteen of these airports were randomly selected as training and validation data for the convolutional neural networks, and the remaining 15 airports were a test set. At such a resolution, a complete satellite image of the airport is more than 5000×5000 in size, which is obviously not suitable for display. We crop the image into slices due to training and testing needs; the size of these slices is 500×500 to 1200×1200 . The final data distribution is shown in the Table 1.

Table 1. The dataset division.

	Number of Samples	Number of Airplanes
Training set	253	2578
Validation set	52	383
Test set	276	2344

In this paper, the only object we want to detect is an airplane, so we only labeled the airplane's location during annotation. The schematic of the annotating sample is shown in Figure 3. The annotated sample is input directly to the convolutional neural networks for training to realize end-to-end training.



Figure 3. The annotated airplane samples.

3.2. The Network Framework

Our airplane detection framework is based on SSD, which is a single convolutional neural network that incorporates feature maps from different layers with fast detection speed and high accuracy. These advantages are suitable for our application scenarios. The first few layers are standard architectures for image classification and were commonly used in the object detection framework, which is also called the base network. We use the VGG16 for the base network. The VGG16's design philosophy is easy to understand and suited for fine tuning, which is also the base network that most researchers use for object detection experiments comparison. After the base network, an additional secondary network structure is added, which replaces the fully-connected network behind the VGG16 and continues adding convolutional layers.

Figure 4 shows the network structure we used during the training phase. As the spatial resolution of the feature map keeps decreasing, the position information of the object is continuously lost. We, therefore, use the shallow features of more complete location information to predict the location and category of objects, since the spatial resolution is higher at this time. Meanwhile, the deep feature map does not lose the estimated position of the object, although it lost more spatial information of the detail. Additionally, the deeper features have more abstract semantic features. Therefore, the shallow features are used to detect small objects, and deep features are used to detect large objects, which can solve the impact of the object scale changes.

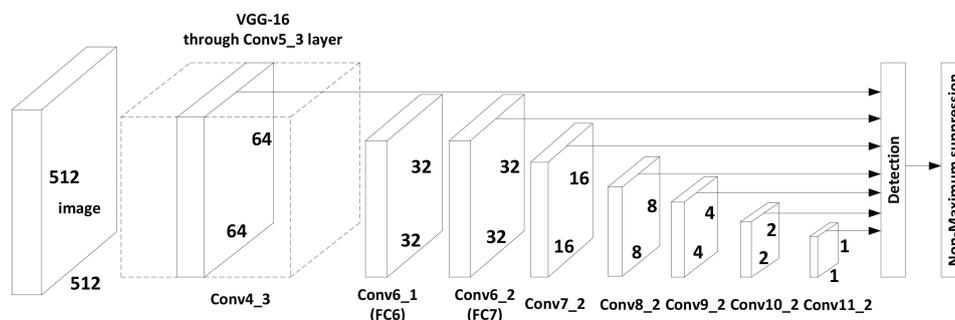


Figure 4. The network architecture. For speeding up training, we do not use MultiBlock and MapBlock in the training phase.

3.3. Training

In this paper, all the experiments we have done are based on an open-source deep learning framework called Caffe [29]. Caffe is used extensively in this area of deep learning, so there are many pre-training models that are based on Caffe. This is very important for our transfer learning, and we directly use the models that have been validated in other areas for fine tuning. The training process is described in the following three concepts.

3.3.1. Data Augmentation

For data in the training phase, the original image is included, as well as the crop of the image, random flip, and the image with added padding after being shrunk. This helps to detect small objects. In addition, a random rotation of an angle added in this paper increases the sample's diversity.

Unlike data augmentation in natural images, most of the objects in a natural image can only be rotated by a relatively small angle, whereas airplanes in remote sensing images can be rotated at any angle. In addition, to increase the robustness of the detector, we also randomly added noise to the training data. The airplane in remote sensing images is not only white in color; we jitter the airplane to increase non-white airplane samples. We also perform affine transformations to the training data due to the different perspectives of remote sensing images. These data augmentations can boost the performance to different degrees. Our experiments show that rotation, affine transformation, and random crops are better than adding noise and jittering. The reason is that the model has a certain degree of anti-noise ability, and the model is not very sensitive to color in this dataset.

3.3.2. The Selection of Positive and Negative Samples

We set six levels of aspect ratios (1, 2/1, 3/1, 1/2, 1/3) for the default box to make the object scalable. In the process of training, the default box is first matched with the ground truth before selecting positive and negative samples. The method of matching is to find a specific default box for each ground truth box with which it has the largest IoU, so that each ground truth box is assigned to a single default box. Then, the remaining non-matched default boxes are matched with any ground truth box. If the IoU of the two is greater than the threshold (set at 0.5 in this paper, which has proved effective in other datasets [16]), it is considered to be matched. After the matching is completed, the matched default box becomes a positive sample, while the non-matched becomes a negative sample. In general, the number of negative samples is much larger than the positive samples, which leads to an imbalance of categories. Therefore, the default box will be sorted according to the confidence of the prediction, and those with high confidence are selected for training. The ratio of the positive and negative samples is controlled at 1:3. In the testing phase, the default box with higher confidence will be selected when performing the network forward computation on the input data.

3.3.3. Loss Function

The loss function uses the SSD approach because airplane detection includes classification and regression. Combining the confidence of the scores with the accuracy of the location forms a multi-task loss. Our loss function is defined as Equation (1):

$$L(x, c, l, g) = \frac{1}{N} \left(L_{conf}(x, c) + \alpha L_{loc}(x, l, g) \right) \quad (1)$$

where N is the number of matched default boxes and α is the balance of two types of losses. The first loss, $L_{conf}(x, c)$ is the loss of confidence, which is actually the Softmax loss. The definition is given by Equation (2):

$$L_{conf}(x, c) = - \sum_{i \in Pos} x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0) \quad \text{where } \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \quad (2)$$

where $x_{ij}^p = \{1, 0\}$ denotes whether the i -th default box matches the j -th ground truth box of class p , c_i^p denotes the confidence that the i -th default box belongs to class p . In this paper, $p = \{0, 1\}$, and when p is 1 it means that it is an airplane (Pos.); when p is 0 it means that it is the background (Neg.). The second item of multi-task loss is the loss of location, which is actually the Smooth L1 loss as shown in Equations (3)–(6).

$$L_{loc}(x, l, g) = \sum_{i \in Pos} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k smooth_{L1}(l_i^m - \hat{g}_j^m) \quad (3)$$

$$\hat{g}_j^{cx} = \frac{g_j^{cx} - d_i^{cx}}{d_i^w} \hat{g}_j^{cy} = \frac{g_j^{cy} - d_i^{cy}}{d_i^h} \quad (4)$$

$$\hat{g}_j^{w} = \log\left(\frac{g_j^w}{d_i^w}\right) \quad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right) \quad (5)$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (6)$$

Here, $(\hat{g}_j^{cx}, \hat{g}_j^{cy}, \hat{g}_j^{w}, \hat{g}_j^h)$ represents the ground truth box, $(d_i^{cx}, d_i^{cy}, d_i^w, d_i^h)$ represents the default box, and $(l_i^{cx}, l_i^{cy}, l_i^w, l_i^h)$ represents the offset of the predicted box relative to the default box. The curve of the Smooth L1 loss is shown in Figure 5, which has the advantage of being less sensitive to outliers than the L2 loss.

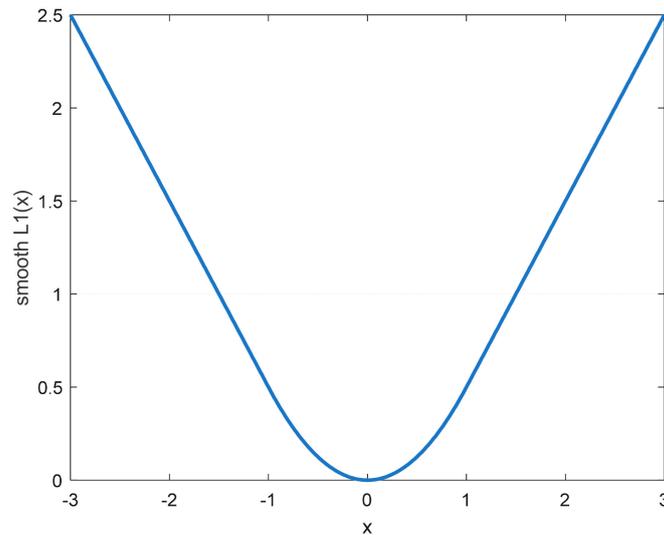


Figure 5. The curve of the Smooth L1 loss.

3.4. Test

3.4.1. MultiBlock and MapBlock Layers

In the testing phase, in order to address different size of input images, we propose a MultiBlock layer and a MapBlock layer. The MultiBlock layer divides the input image into multiple blocks; it is added before the input layer. The MapBlock layer maps the prediction result of each block back to the original image; it is added after the detection output layer. As shown in Figure 6, resizing large images directly will reduce the object information, resulting in decreases in detection performance. The MultiBlock layer and MapBlock layer are proposed to prevent such situations. In this paper, the focus of our research is mainly on the airplane detection of the airport; our research assumes that the

airport has been located. The input size of the data to be tested is more flexible when the MultiBlock and MapBlock layers are applied to the detection architecture.

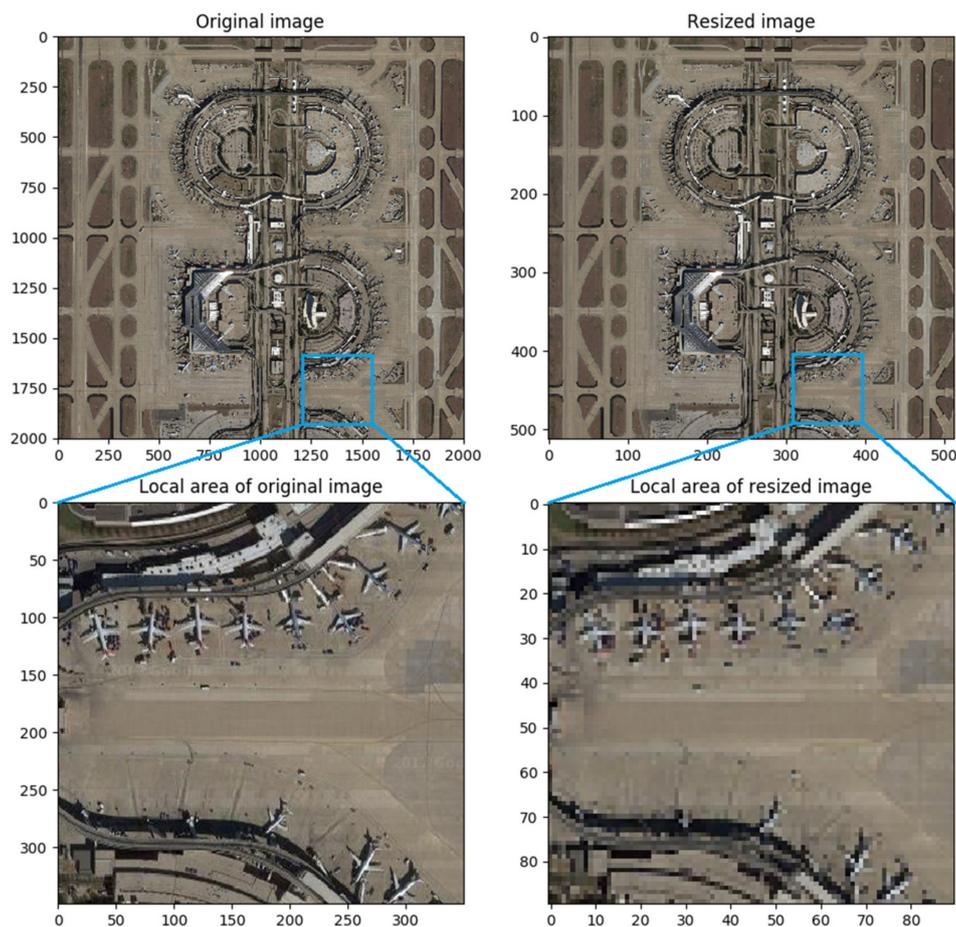


Figure 6. Resizing the original image with a large size. The part marked with the blue box in the figure is enlarged and displayed below it. It can be seen that the resolution of the image is reduced. Since the image is resized directly, the object contour is blurred, which will reduce detection precision.

In this paper, the network input is still 512×512 , so images that do not meet the requirements will be resized. However, the MultiBlock layer does not divide the input image into multiple blocks of 512×512 size, but divides them into multiple blocks around the size of 512×512 and then resizes them to 512×512 when they are input in the network. The advantage of this approach is to reduce the loss of small objects caused by directly resizing large images. At the same time, because of the robustness of the convolutional neural network, the very small resizing of the image will not have a large impact on the final prediction result, so it also has some flexibility in the division of the block.

After the image is processed by the MultiBlock layer, the number of blocks in the vertical and horizontal directions is, respectively, m and n , and the total number is $m \times n$. The calculation formula is shown in Equations (7) and (8):

$$m = \left\lceil \frac{\text{Height} - \text{Overlap}}{\text{Block Height} - \text{Overlap}} \right\rceil \quad (7)$$

$$n = \left\lceil \frac{\text{Width} - \text{Overlap}}{\text{Block Width} - \text{Overlap}} \right\rceil \quad (8)$$

where *Height* and *Width* are the height and width of the original image respectively, and *Overlap* is the overlap distance between the blocks, which can be set according to the resolution of the remote sensing image. In this paper, *Overlap* is 100. *Block Height* and *Block Width* are, respectively, the height and width of the block to be generated. The MapBlock layer maps the detection results of multiple blocks back to the original image to realize the detection for large images. As shown in Figure 7, after a 2000×2000 image passes through the MultiBlock layer, nine partially-overlapping blocks are obtained according to the selected parameters: *Overlap* = 100, *Block Height* = 800 and *Block Width* = 800. When passing through the MapBlock layer, the detection results in each block are merged and mapped back to the original image to obtain the final detection result. In detail, we calculate the position of objects in the original image according to the parameters of the MultiBlock layer and the detection results in each block.

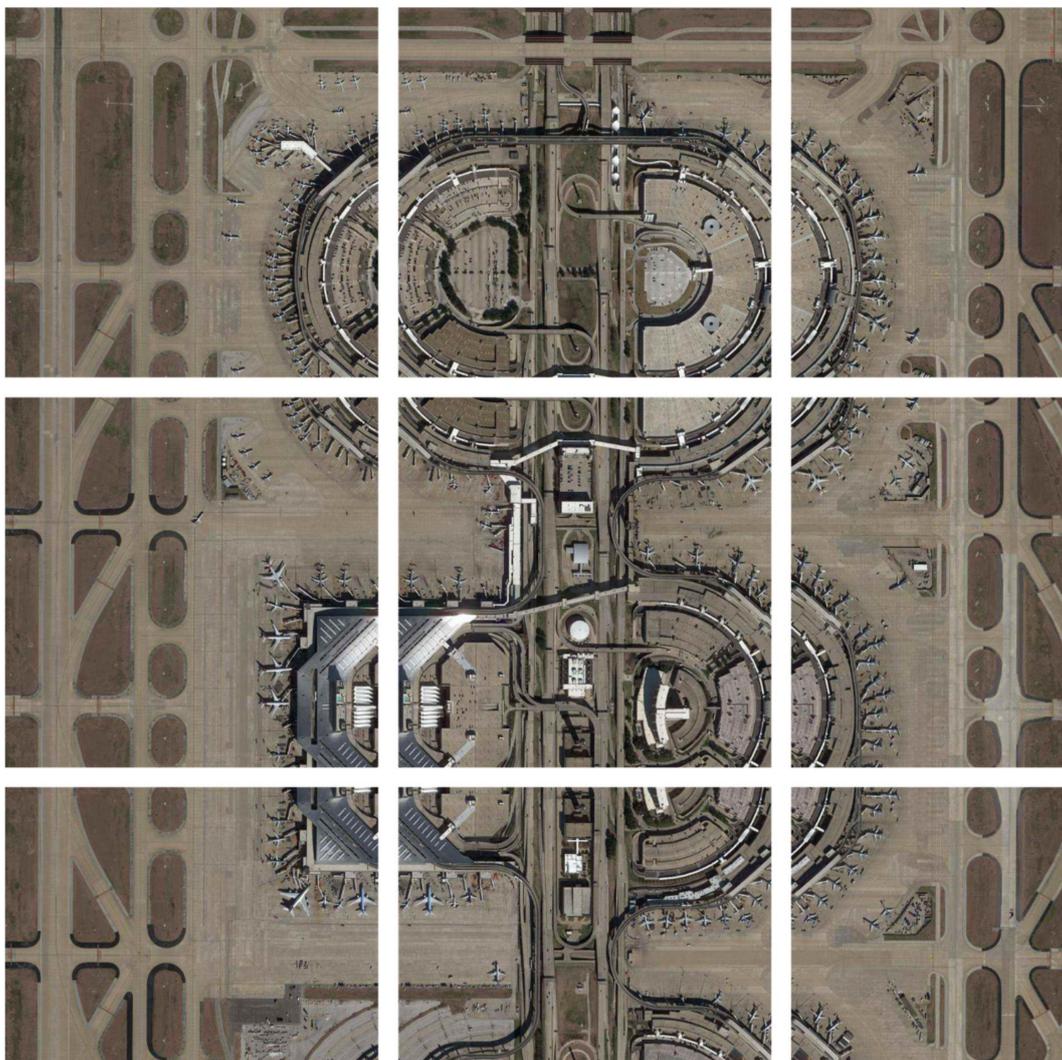


Figure 7. The MultiBlock layer. The input image is divided into nine overlapped blocks after the MultiBlock layer is applied.

3.4.2. Non-Maximum Suppression

We noticed that the SSD performed a non-maximum suppression (NMS) operation when in prediction mode. However, the SSD resizes the input image directly and does not perform NMS operations in the original image size, which is obviously not reasonable for small objects, such as airplanes. Using SSD for airplane detection, we find that there are often two bounding boxes in the same location in our experiment, which can be prevented by adding an NMS operation again.

3.5. Implementation

Our implementation details are listed in this section. In the training phase, we used SGD with a mini-batch size of 32. We used a weight decay of 0.0005 and a momentum of 0.9. We trained our model for up to 100,000 iterations. The learning rate started from 0.001 and was divided by 10 on the 50,000th iteration and the 80,000th iteration. Our training process was run on a GTX 1080 Ti with 11GB of memory. In order to simplify the training process, we did not use the MultiBlock and MapBlock layer method in the training phase. Cropped training data can be considered as blocks of a large-sized image, so training with different cropped images is equivalent to training with multiple block images. In the testing phase, we added the MultiBlock and MapBlock layer for the improvement of small object detection performance and the convenience of practical application.

4. Results and Discussion

4.1. Results

In this paper, we used mean average precision (mAP) as the criteria for airplane detection. In multiple class object detection, we can draw a curve based on recall and precision for each class, where AP is the area under the curve and mAP is the average of multiple categories' APs. Since we only detect airplanes, the area under the PR curve for airplane detection is our evaluation index.

As shown in Figure 8, our method has a mAP value of 96.23% on the test set, which is higher than the 86.28% from the SSD method. With precision higher than 90%, recalls have also reached more than 90%. During the test phase, the GPU we used is NVIDIA GeForce 940 M, and the average prediction time of SSD is 513.76 ms for each image. Since we use MultiBlock and MapBlock layers for large images, the average time for each image we test on the test set is 1934.63 ms. Due to the better adaptability of our method to the image size, the single convolutional neural network guarantees the detection speed and has good detection performance.

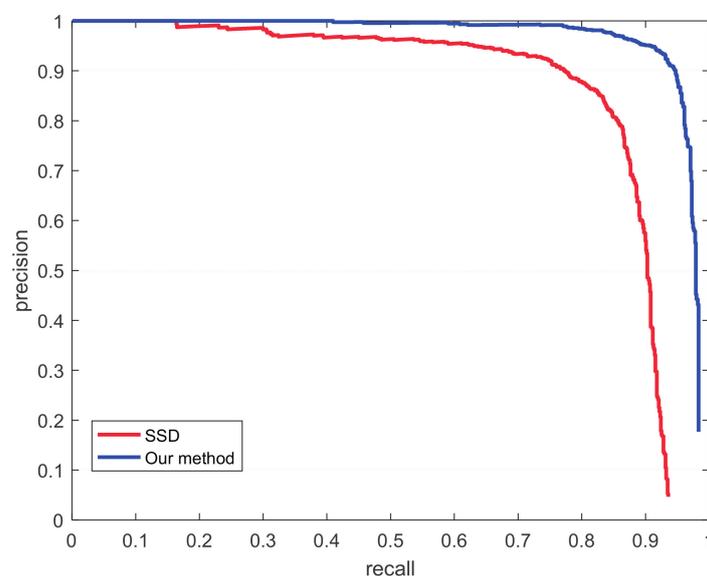


Figure 8. The Precision-Recall (PR) curve. The blue curve is the result of our method on the test set and the red one is the SSD. It can be seen that with the same precision, our method has an obvious advantage in the recall rate.

Figure 9 is the result of our method on some test images. The airplane detection algorithm based on transfer learning adapts to the background very well. First, the false alarm rate is greatly reduced, and, second, the object recall rate is significantly improved. Certainly, this also benefits from increasing the network depth, which makes the extracted features more robust.

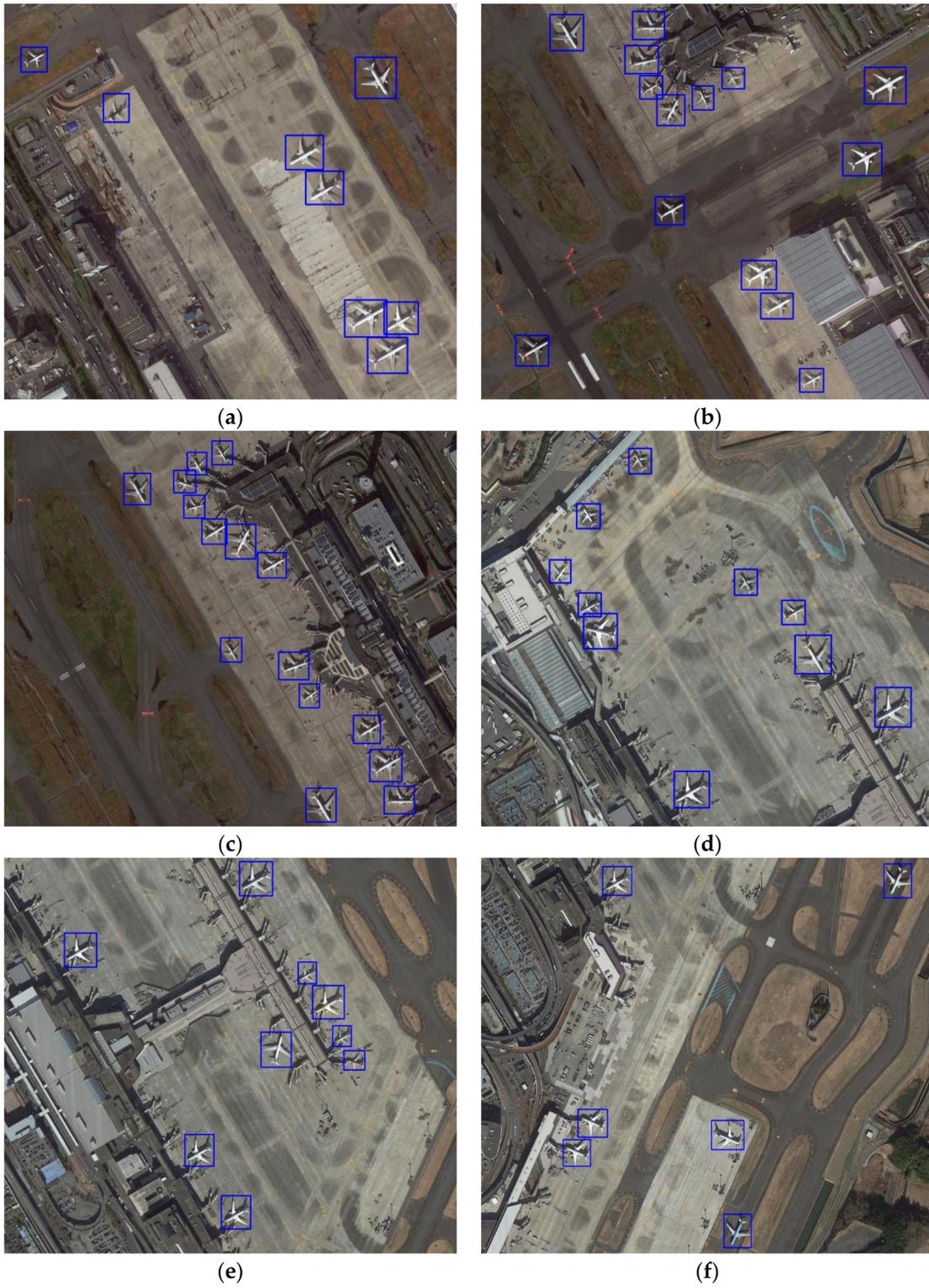
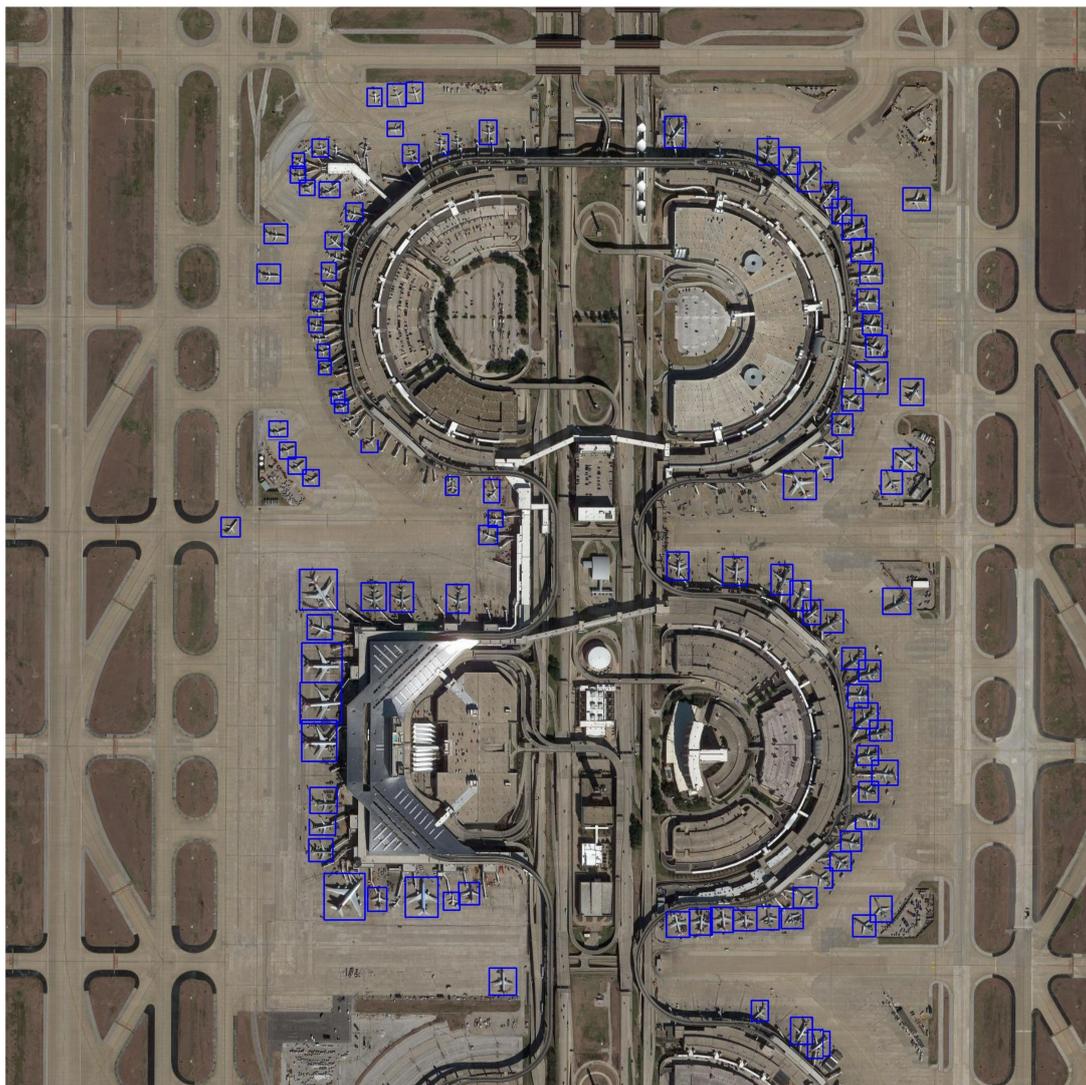


Figure 9. Cont.



(g)

Figure 9. The results of airplane detection. (a–f) are detection results of different small input images; and (g) is a large image; it is clearly seen that the small objects are detected accurately.

4.2. Discussion

First, we describe the several primary ways in which we combat overfitting. Training data is a crucial component of deep learning tasks. It is easy to overfit if only a limited number of samples are used to train a deep convolutional neural network. In this paper, however, we have taken these measures to prevent overfitting: according to transfer learning, we initialize our network using pre-trained models rather than random initialization to address the problems caused by small sample sizes. In the process of training, we also create adequate data augmentation to increase sample diversity. By analyzing the trend of training loss, we use early stopping to avoid over-learning. In our proposed method, the choice of hyper-parameters is also an important factor. Like most prevalent object detection frameworks, this method requires much experimental verification.

Finally, our approach is extensible in two main aspects. First, when it is necessary to detect other objects, such as ships or vehicles, we can achieve good results in new detection tasks by only replacing the data set. Additionally, the proposed MultiBlock and MapBlock layers can be added to other network architecture, such as GoogLeNet and ResNet, as well as self-designed networks. In this paper, we choose VGG16 for balance between model performance and training speed.

5. Conclusions

In this paper, we propose an airplane detection algorithm based on a single convolutional neural network. Through transfer learning and the airplane samples we collected from Google Earth, we have implemented an end-to-end trainable airplane detection framework. We add a rotation operation to increase the diversity of training samples during data augmentation. When dealing with remote sensing images of large size for input, we propose the MultiBlock layer and MapBlock layer, which effectively solves the problem of small object loss caused by directly resizing the image. The airplane detection framework we propose is very effective because it is also applicable to other object detection tasks in remote sensing image processing. Due to the single convolutional neural network, the detection speed is superior to that of the two-stage approach. At the same time, the training process is simplified and easy to converge due to the advantages of the end-to-end trainable framework. Our experimental results also show that the proposed airplane detection algorithm in this paper has good detection performance.

Author Contributions: Z. Chen conceived and designed the study. T. Zhang and C. Ouyang developed the algorithm and performed the experiments. Z. Chen conducted the experiments and analyzed the data. C. Ouyang wrote the paper, and T. Zhang contributed to the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Filippidis, A.; Jain, L.C.; Martin, N. Fusion of intelligent agents for the detection of aircraft in SAR images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 378–384. [[CrossRef](#)]
2. Yao, J.; Zhang, Z. Semi-supervised learning based object detection in aerial imagery. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 1011–1016.
3. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–8 December 2012; Volume 2, pp. 1097–1105.
4. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.F. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
5. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV* **2004**, *60*, 91–110. [[CrossRef](#)]
6. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
7. Szegedy, C.; Liu, W.; Jia, Y.Q.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
8. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv*, **2014**. [[CrossRef](#)]
9. He, K.M.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
10. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
11. He, K.M.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 1904–1916. [[CrossRef](#)] [[PubMed](#)]
12. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
13. Ren, S.; He, K.M.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Computer Vision, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.

14. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
15. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
16. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL visual object classes (VOC) challenge. *IJCV* **2010**, *88*, 303–338. [[CrossRef](#)]
17. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
18. Hosang, J.; Benenson, R.; Dollár, P.; Schiele, B. What makes for effective detection proposals? *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 814–830. [[CrossRef](#)] [[PubMed](#)]
19. Wu, H.; Zhang, H.; Zhang, J.; Xu, F. Fast aircraft detection in satellite images based on convolutional neural networks. In Proceedings of the IEEE International Conference on Image Processing, Quebec City, QC, Canada, 27–30 September 2015; pp. 4210–4214.
20. Cheng, M.M.; Zhang, Z.; Lin, W.Y.; Torr, P. BING: Binarized normed gradients for objectness estimation at 300fps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3286–3293.
21. Zitnick, C.L.; Dollár, P. Edge boxes: Locating object proposals from edges. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 391–405.
22. Carreira, J.; Sminchisescu, C. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1312–1328. [[CrossRef](#)] [[PubMed](#)]
23. Arbeláez, P.; Pont-Tuset, J.; Barron, J.T.; Marques, F.; Malik, J. Multiscale combinatorial grouping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 328–335.
24. Alexe, B.; Deselaers, T.; Ferrari, V. Measuring the objectness of image windows. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2189–2202. [[CrossRef](#)] [[PubMed](#)]
25. Zhang, W.; Sun, X.; Fu, K.; Wang, C.; Wang, H. Object detection in high-resolution remote sensing images using rotation invariant parts based model. *IEEE Trans. Geosci. Remote Sens.* **2014**, *11*, 74–78. [[CrossRef](#)]
26. Wang, G.; Wang, X.; Fan, B.; Pan, C. Feature Extraction by Rotation-Invariant Matrix Representation for Object Detection in Aerial Image. *IEEE Trans. Geosci. Remote Sens.* **2017**, *14*, 851–855. [[CrossRef](#)]
27. Liu, L.; Shi, Z. Airplane detection based on rotation invariant and sparse coding in remote sensing images. *Optik-Int. J. Light Electron Opt.* **2014**, *125*, 5327–5333. [[CrossRef](#)]
28. West, J.; Ventura, D.; Warnick, S. *Spring Research Presentation: A Theoretical Foundation for Inductive Transfer*; Brigham Young University, College of Physical and Mathematical Sciences: Provo, UT, USA, 2007.
29. Jia, Y.Q.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.

