Deeper SSD: Simultaneous Up-sampling and Down-sampling for Drone Detection

Han Sun[1,2], Wen Geng[1,2], Jiaquan Shen[1,2], Ningzhong Liu[1,2], Dong Liang[1,2], and Huiyu Zhou[3]
[1] College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Jiangsu Nanjing, 211106, China
[e-mail: {sunhan, gengwen, shenjiaquan, liangdong}@nuaa.edu.cn, liunz@163.com]
[2] MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
Jiangsu Nanjing, 211106, China
[3] School of Informatics, University of Leicester, United Kingdom
[e-mail: hz143@leicester.ac.uk]
*Corresponding author: Han Sun

Received September 27, 2019; revised September 10, 2020; accepted November 23, 2020; published December 31, 2020

Abstract

Drone detection can be considered as a specific sort of small object detection, which has always been a challenge because of its small size and few features. For improving the detection rate of drones, we design a Deeper SSD network, which uses large-scale input image and deeper convolutional network to obtain more features that benefit small object classification. At the same time, in order to improve object classification performance, we implemented the up-sampling modules to increase the number of features for the low-level feature map. In addition, in order to improve object location performance, we adopted the down-sampling modules so that the context information can be used by the high-level feature map directly. Our proposed Deeper SSD and its variants are successfully applied to the self-designed drone datasets. Our experiments demonstrate the effectiveness of the Deeper SSD and its variants, which are useful to small drone's detection and recognition. These proposed methods can also detect small and large objects simultaneously.

Keywords: Drone Detection, Small Object Detection, Deeper SSD, Up-sampling Modules, Down-sampling Modules

This research was supported by the Fundamental Research Funds for the Central Universities (No. NS2016091).

http://doi.org/10.3837/tiis.2020.12.010 **ISSN : 1976-7277**

1. Introduction

In recent years, drone has been widely used in commercial and entertainment areas. While bringing great convenience, it might be an invasion of personal privacy and even threaten the safety of civil aviation. Therefore, drone monitoring system is essential and drone detection becomes the important part of such system.

Because drone's size is too small in surveillance videos, drone detection belongs to a specific sort of small object detection. As is known, small object detection is a great challenge in computer vision. Common object detection methods based on feature extraction, such as SIFT [1], HOG [2] and Haar-like [3-4], are difficult to extract useful and suitable features from small objects. As for deep learning based methods, they have achieved good performance for normal object detection and recognition. Nevertheless, such popular methods are also easy to fail for small object detection. Because small objects have low resolution and are lack of distinct structure, which are difficult for deep neural networks to learn rich representations. Deep learning based object detection algorithms can be divided into two categories, classification-based detectors and regression-based detectors. And among classification-based methods, R-CNN series [5-10] are the representatives, which are not very effective for small object detection because small objects can easily be lost in complex scenes. And regression-based detectors, which mainly include YOLO series [11-14] and SSD series [15-16], have the same problem. YOLO has a down-sampling factor of 32 in its network and returns a 13×13 prediction grid, which means the object will be lost on the 13×13 prediction grid that if it has less than 32 pixels. As shown in **Fig. 1**, SSD also has this problem. The resolution of the sample input image is 300×300, and the size of the drone object is about 11×19, which only occupies about 0.1% of the whole image. Going through the convolutional layers, the size of drone shrinks about 1×1 when reaching *Conv4_3* layer. After *Conv4_3* layer, the detailed representation of drone object will be gradually or totally lost. Meanwhile, SDD still has such a contradiction. SSD adopts the hierarchical structure of feature pyramid and uses different convolutional layers to detect objects. The low-level feature maps are large, but the semantic information is not enough. While the high-level layers have rich semantic information, but after too many pooling layers, the size of feature map is too small. Therefore, in order to detect small objects, we need not only a large enough feature map to ensure that small object information can be retained during the network transmission, but also sufficient semantic information to distinguish small objects from background.

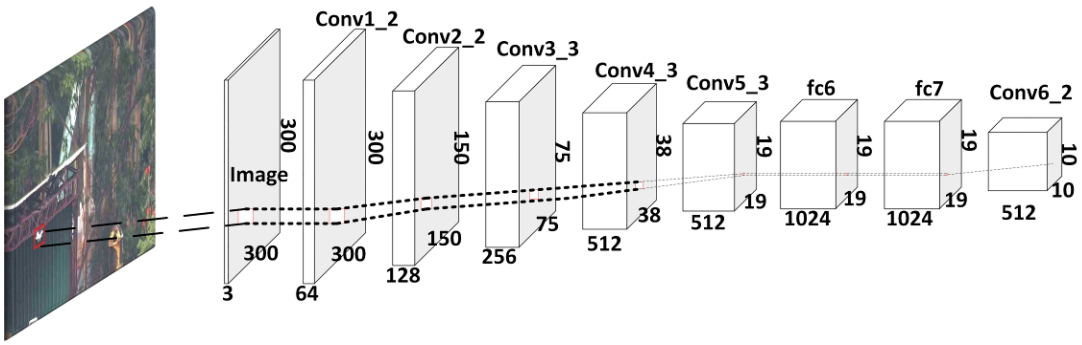


Fig. 1. Illustration of small object detection problem using SSD

According to the definition provided by SPIE, the small object has less than 80 pixels in the 256×256 image, which means that the area of small object is less than 0.12% in the whole image. Furthermore, the small objects have small size, which leads to lack rich features such as shape, edge, and texture. Moreover, the small objects have weak intensity and contrast, so they are easily submerged into background and noise.

Therefore, based on the above analysis of small object detection and the characteristics of drone objects, we propose a novel network Deeper SSD to detect small drone objects, which uses simultaneous up-sampling and down-sampling operation to make full use of the high-level and low-level features of the deep network. For solving the problem shown in **Fig.1**, this paper also adopts two methods, enlarging the input image size to 500×500 and adding three convolutional layers to the standard SSD to deepen the network. In this way, the proposed network is easier to obtain more features beneficial to object classification. For the problem that the low-level feature map has large size with insufficient semantic information, we adopt the up-sampling method, which adds the high-level semantic features to the low-level feature map for improving classification performance. As for the problem that the size of high-level feature map is too small to distinguish drones from the background, we use the down-sampling method to add the low-level features to the high-level feature map for raising the location performance.

In general, the main contributions of our work are as follows.

(1) A novel deep network, Deeper SSD, is proposed for small object detection.

(2) In order to improve the detection accuracy of small objects, we implement up-sampling and down-sampling simultaneously on the Deeper SSD network to make full use of the high-level and low-level features. By this way, the performance of object classification and location can be improved at the same time.

(3) Based on the self-designed drone datasets, we conduct the experiments to evaluate the performance of Deeper SSD and its variants, which are better than traditional SSD, especially for small object detection.

2. Related Work

In this section, we review the related studies in three parts.

2.1 Object Detection

Object detection plays an important role in many computer vision based applications, such as video understanding, detection and tracking monitoring system. At present, most popular object detection methods are based on deep learning.

The deep learning based object detection are mainly divided into two categories. One is the two-stage method, and the other is the one-stage method. The two-stage detectors mainly include R-CNN series [5-10], SPP-Net [17] and HyperNet [18]. These methods generate a series of candidate bounding boxes firstly, and then use the convolutional neural network to classify objects. The one-stage detectors mainly include OverFeat [19], YOLO series [11-14] and SSD series [15-16]. Compared to the two-stage methods, the one-stage methods directly turn the problem of object location into that regression. Therefore, the two-stage method is superior in the accuracy of object location and detection, while the one-stage method has an advantage in the algorithm's speed. Taking Faster R-CNN [7] and YOLO [11] as an example, for Faster R-CNN, due to the structure of Region Proposal Network (RPN), its speed run into bottleneck and it is difficult to satisfy the requirements of real-time applications. However, the detection accuracy is high because RPN first roughly generates candidate bounding boxes and

then goes to more detailed classification objects. As for YOLO, it divides the final feature map into a 2D grid and directly predicts a bounding box using each grid cell, which increases the speed. However, this process can produce many negative samples, resulting in decreased detection accuracy. Moreover, some anchor-free object detection methods [20-22] have been presented recently, which avoid the manual design of anchors and achieve acceptable detection performance.

2.2 Small Object Detection

Although object detection has achieved good performance, small object detection has always been a great challenge [23]. The traditional methods based on hand-crafted features have not achieved perfect results. [24] considers that the small object is the saliency region in the image. [25] uses machine learning based methods to detect model small objects. But small objects have limited pixels and rare information, these traditional methods can hardly extract effective features.

Recently, deep learning based methods have been widely used to improve the performance of small object detection. In 2016, Yongxi Lu et al. proposed the AZ-net network [26], which is based on Faster R-CNN [7], for the case of only a small number of small objects. In 2017, Adam Van Etten proposed the YOLT network [27], mainly to detect small objects in satellite imagery. Guimei Cao et al. presented the Feature-Fused SSD [28], which designed two feature fusion modules, the concatenation module and the element-sum module, and added them to the SSD network to enrich the semantic information by two different ways. Jianan Li et al. utilized the idea of Generative Adversarial Network [29] to create the Perceptual GAN model [30], which narrows the representation difference between small objects and large objects and so can improve the performance of small object detection. And in 2018, Lisha Cui built the MDSSD network [31], which added multi-scale deconvolution fusion modules into the SSD network to increase the low-level semantic information. And Zhenhua Chen et al. proposed Filter-Amplifier Networks [32] to detect densely distributed small objects in images. Kui Fu et al. proposed a novel context reasoning approach for small object detection which models and infers the intrinsic semantic and spatial layout relationships among objects [33]. On the other hand, some researchers take the resolution of image into consideration. Motivated by keeping the benefits of high-resolution images without bringing up new problems, Ziming Liu et al. presented the High-Resolution Detection Network (HRDNet) [34], which can fully take advantage of multiple features and maintain multiple position information. Based on multi-resolution feature extraction, Fan Zhang et al. proposed a simple and effective feature extraction method Multi Resolution Attention Extractor (MRAE) [35] to mine the most useful information of small objects, which far exceeds the powerful baselines and is highly time efficient.

2.3 Drone Detection

Drone detection is a special sort of small object detection. Although drone detection is a challenging studied subject, there are still many attempts which are worth mentioning. [36] integrated a fast salient object detector within the Kalman filtering framework, which can be applied to drone navigation, such as obstacle sense and avoidance. Compared to other trackers, the approach can not only be initialized automatically, it can also achieve the fastest speed and better performance. Although the proposed tracking approach outperformed its competitors in most experiments, it has a key limitation in handling occlusion challenge. [37] developed an object-centric learning-based motion compensation approach which used CNN and Boosted trees methods to detect small fast moving objects like UAVs or aircrafts in complex outdoor

environments. Although this approach outperformed other techniques on their datasets, the precision of UAVs detection still needs to be improved. Y. Chen et al.[38] proposed a drone monitoring system based on Faster R-CNN framework, which was integrated with detection and tracking module. The fully integrated system took advantage of both modules to achieve high performance monitoring. They also developed a model-based data augmentation to enrich the training set[39]. It used an end-to-end neural network, which was based on YOLOv2 [12], to predict the position of drone in the images. The study showed that drones can be detected and distinguished from birds using an object detection model. Considering the deployment on some edge device, some novel lightweight methods such as TIB-Net [40] have been proposed for drone detection. Hu et al. improved original YOLOv3 [13] method and applied it to drone detection task. As a result, the approach eventually improved the accuracy of drone detection while ensuring the detection speed [41].

3. Deeper SSD

In this section, we first review the SSD network briefly and then introduce the Deeper SSD and its variants. After that, a detailed analysis of the Deeper SSD and its variants is given as follows.

3.1 SSD

Fig. 2 shows the architecture of the SSD300x300 [15]. The network uses the standard VGG16 as a feature extractor and then adds additional convolutional layers to the truncated VGG16 network. SSD adopts feature pyramid hierarchy structure and uses multiple convolution layers to predict objects. This network improves *mAP* by multi-scale method.

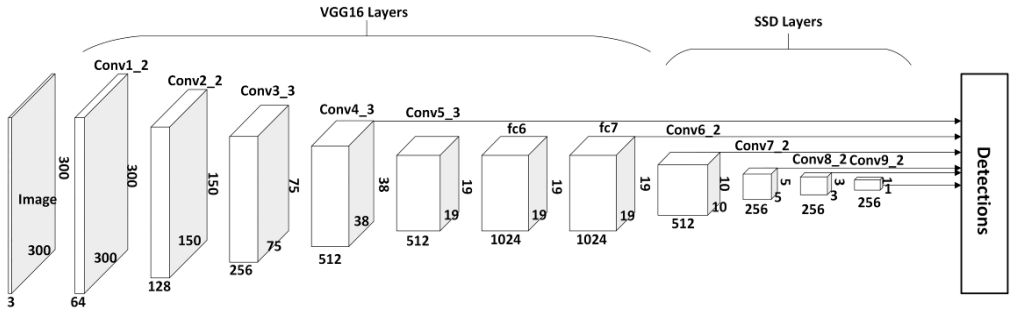


Fig. 2. The Architecture of SSD300x300

However, for small object detection, the detection performance of the SSD is not satisfied. There are two main reasons to explain such results. The first reason can refer to the problem existing in the SSD network described in **Fig. 1**, which means the size of objects is too small for the network. The second reason is that the low-level feature map of the SSD network is large, but the semantic information contained in this feature map is not enough. While the semantic information of the high-level feature map is rich, but after too many pooling layers, the size of this feature map is too small. Therefore, for small object detection, the more feature map size, the more classification features and the more semantic information are very important. Consequently, we propose the Deeper SSD and its variants.

3.2 Deeper SSD

We know that when the object area is smaller than 0.12% of the whole image, the object can be called as small object. As shown in **Fig. 1**, the size of drone is about 11×9 , which is about 0.1% of the image. We can only extract rich information of drones before the *Conv4_3* layer in the SSD. That is to say, after the *Conv4_3* layer, the details of drones will be gradually lost or totally lost. Therefore, the intuitive approach is to increase the input image size, which is used to increase the size of the prediction feature map. At the same time, in order to make it easier to obtain more features that are beneficial for small object classification, the deepening of the network is also essential. Based on the above ideas, we propose a novel network Deeper SSD, which is based on the SSD. **Fig. 3** demonstrates the architecture of the Deeper SSD. In order to show the difference between the Deeper SSD and the original SSD more clearly, the input sample image size of the SSD network should be set to 500×500.

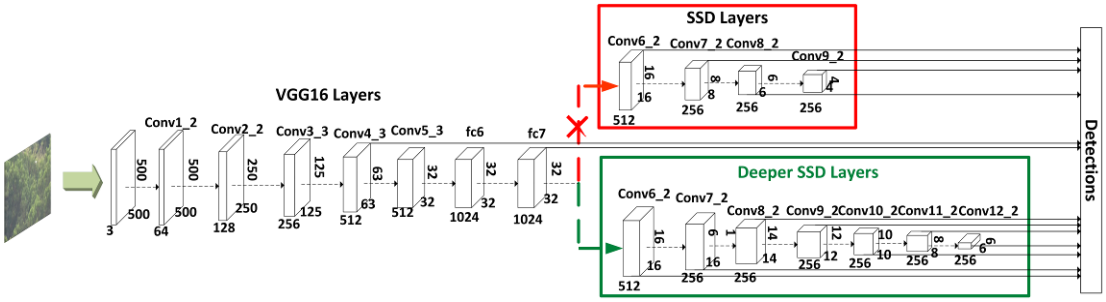


Fig. 3. The architecture of the Deeper SSD

Referring to engineering experience, the performance of small object detection is closely related to the size of feature map. In order to increase the size of high-level and low-level prediction feature maps in the SSD network, we change the size of input image from 300×300 to 500×500 and the stride of *conv6_2* from 2 to 1. Meanwhile, in order to obtain more features that are conducive to small object classification, we add three convolutional layers behind the SSD network, which also means that the number of prediction layers has changed from 6 to 9.

As seen from **Fig. 3**, there are three main differences between the Deeper SSD and the SSD: (1) The size of feature maps outputted from *Conv7_2*, *Conv8_2* and *Conv9_2* layer is increased. (2) Three convolutional layers of *conv10*, *conv11* and *conv12* are added behind the SSD network, making it easier to obtain more features that are useful for small object classification. (3) The number of prediction layers has changed from the original 6 to 9, which increases the possibility of object prediction. We refer to the improved network layers as the Deeper SSD Layers. More detailed differences between the Deeper SSD and the SSD network can be found in **Table 1**.

Table 1. The comparison of the Deeper SSD500x500 and the SSD300x300

Layers	Deeper SSD500x500			SSD300x300		
	Outputs	Size/Strdie/pad	Output Size	Outputs	Size/Strdie/pad	Output Size
Conv1_1	64	3x3/1/1	500x500	64	3x3/1/1	300x300
Conv1_2	64	3x3/1/1	500x500	64	3x3/1/1	300x300
Pool1			250x250			150x150

Conv2_1	128	3x3/1/1	250x250	128	150x150	150x150
Conv2_2	128	3x3/1/1	250x250	128	150x150	150x150
Pool2			125x125			75x75
Conv3_1	256	3x3/1/1	125x125	256	75x75	75x75
Conv3_2	256	3x3/1/1	125x125	256	75x75	75x75
Conv3_3	256	3x3/1/1	125x125	256	75x75	75x75
Pool3			63x63			75x75
Conv4_1	512	3x3/1/1	63x63	512	3x3/1/1	38x38
Conv4_2	512	3x3/1/1	63x63	512	3x3/1/1	38x38
Conv4_3	512	3x3/1/1	63x63	512	3x3/1/1	38x38
Pool4			32x32			19x19
Conv5_1	512	3x3/1/1	32x32	512	3x3/1/1	19x19
Conv5_2	512	3x3/1/1	32x32	512	3x3/1/1	19x19
Conv5_3	512	3x3/1/1	32x32	512	3x3/1/1	19x19
Pool5			32x32			19x19
Fc6	1024	3x3/1/6	32x32	1024	3x3/1/6	19x19
Fc7	1024	1x1/1/0	32x32	1024	1x1/1/0	19x19
Conv6_1	256	1x1/1/0	32x32	256	1x1/1/0	19x19
Conv6_2	512	3x3/2/1	16x16	512	3x3/2/1	10x10
Conv7_1	128	1x1/1/0	16x16	128	1x1/1/0	10x10
Conv7_2	256	3x3/1/0	16x16	256	3x3/2/1	5x5
Conv8_1	128	1x1/1/0	16x16	128	1x1/1/0	5x5
Conv8_2	256	3x3/1/0	14x14	256	3x3/1/0	3x3
Conv9_1	128	1x1/1/0	14x14	128	1x1/1/0	3x3
Conv9_2	256	3x3/1/0	12x12	256	3x3/1/0	1x1
Conv10_1	128	1x1/1/0	12x12			
Conv10_2	256	3x3/1/0	10x10			
Conv11_1	128	1x1/1/0	10x10			
Conv11_2	256	3x3/1/0	8x8			
Conv12_1	128	1x1/1/0	8x8			
Conv12_2	256	3x3/1/0	6x6			

As we know, in the pyramid structure of network, the low-level layers are used to detect small objects, and the high-level layers are used to detect large objects. The increase of the input image size and the deepening of the network not only can increase the size of the prediction feature maps, but also increase the number of prediction layers. Such operations can prevent the premature disappearance of small object fine-grained information in the network and increase the possibility of small objects being predicted in the low-level layers. Here the low-level and high-level layers of the network are a relative concept.

Because the Deeper SSD proposed in this paper still adopts feature pyramid hierarchical structure and uses different network layers to detect objects, the Deeper SSD has the same problem as the SSD. The size of low-level feature map is large, but the semantic information contained is not enough. While the semantic information of the high-level feature map is rich, but after too many pooling layers, the size of this feature map is too small. Therefore, to detect small objects, we need not only a large enough feature map to ensure that small object information can be retained during the network transmission, but also sufficient semantic information to distinguish small objects from background. To solve this problem and further improve small object detection precision, we adopt three methods: up-sampling, down-sampling and simultaneous up-sampling and down-sampling. The specific implementation schemes of these three methods will be given.

3.2.1 Up-sampling Method

In order to enrich semantic information of the low-level feature map, and further improve the performance of small object classification, we adopt up-sampling method for the Deeper SSD. As we know, the low-level feature map is more suitable for the precise location of objects due to small receptive field, but weak semantic information is not conducive to object classification for low-level feature map, especially for small objects. In order to solve this problem, we add the high-level features with rich semantic information to the low-level feature map through the up-sampling operation. We use the Deeper SSD as the backbone network, and then add up-sampling modules, shown in **Fig. 4**.

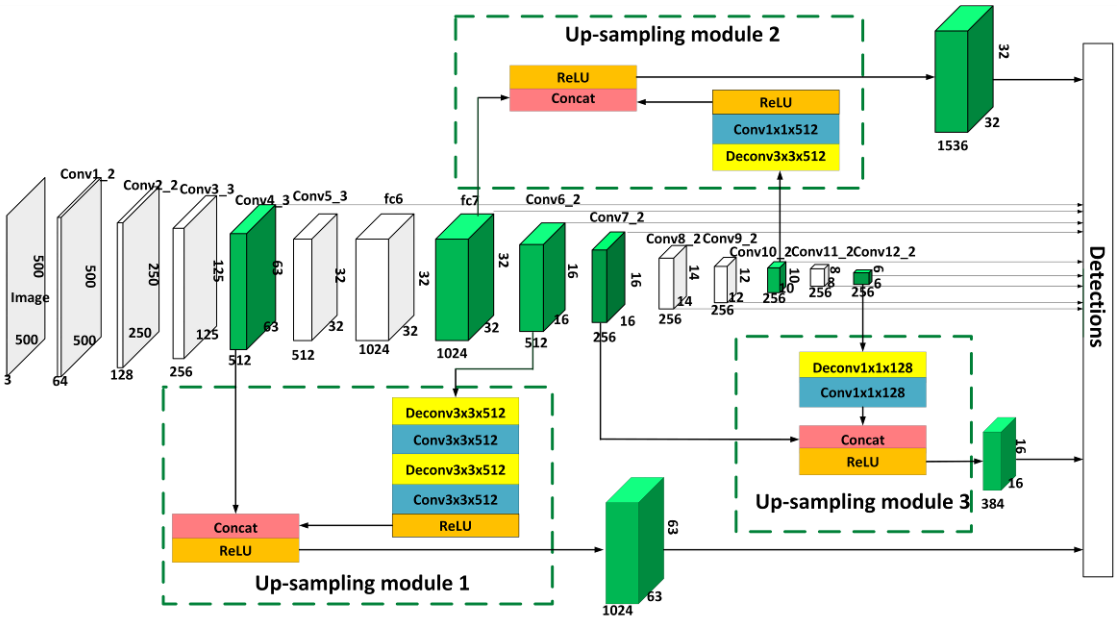


Fig. 4. The architecture of three up-sampling modules for Deeper SSD

Three up-sampling modules are added to the Deeper SSD network. Let us take the up-sampling module 1 as an example. The up-sampling module 1 adopts the up-sampling method for *Conv6_2* layer, and connects the feature map outputted after the up-sampling with the feature map outputted by *Conv4_3* layer. Here the *Concat* operation is used to connect these two feature maps, so we need the same size for both feature maps from different layers.

In order to connect *Conv4_3* and *Conv6_2*, we need to take deconvolution operation on *Conv6_2*. For *Conv6_2*, as shown in **Fig. 4**, we implement two deconvolution layers of stride 3 to achieve up-sampling, and each deconvolution layer can enlarge the feature map triple in size. We use 3x3 or 1x1 kernel size with 256 or 512 outputs. The deconvolution layers are followed by convolution layers. We use the *Concat* layer to connect the *Conv6_2* up-sampling output layer with the *Conv4_3* layer. After that, we add a *ReLU* layer. Finally, we obtain a prediction feature map.

It is worth noting that we reduce the number of deconvolution layers as much as possible while enlarging the size of *Conv6_2* feature map. The up-sampling module output layer acts as a prediction layer. The number of prediction layers of the whole network has been changed to 12. Due to the weak semantic information, the low-level feature map is not suitable for the object classification. Therefore, we adopt *Concat* operation to increase the number of channels

appropriately, that is, increase the number of features appropriately, which can improve the object classification performance. The experiment in section 4.4.2 also proves our idea.

3.2.2 Down-Sampling Method

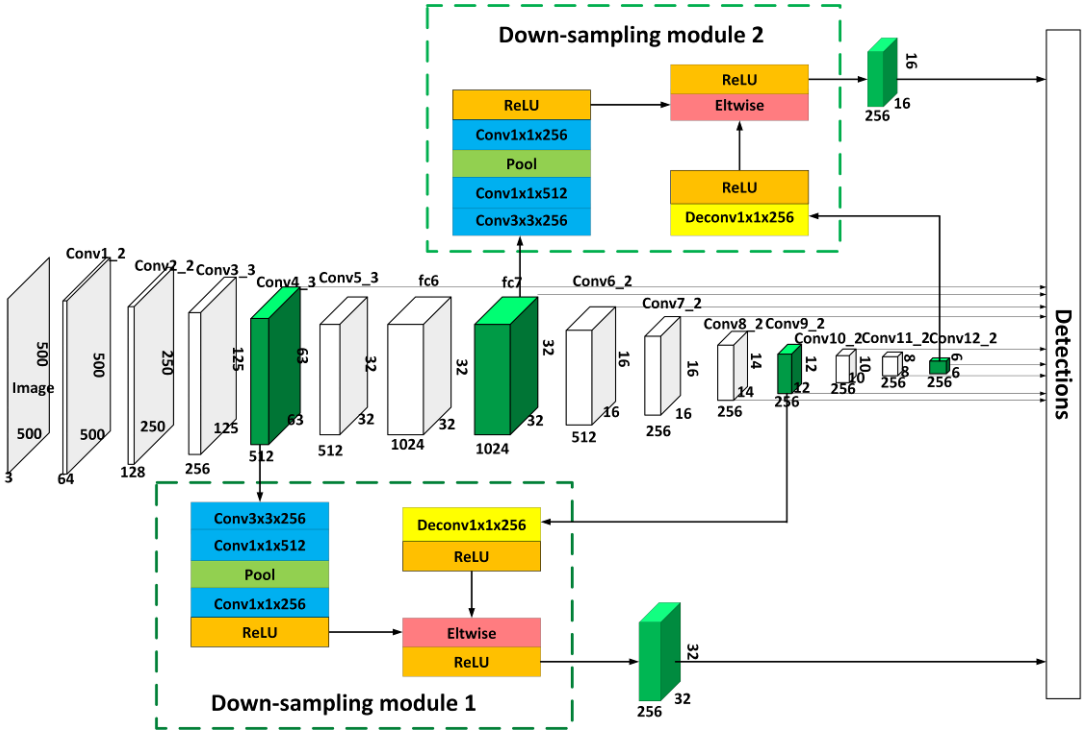


Fig. 5. The architecture of two down-sampling modules

As is known, the semantic information of the high-level feature map is rich, but after too many pooling layers, the size of the feature map is relatively small, and small objects are not easy to be distinguished from the background. To further improve the performance of small object location, we adopt down-sampling method for the Deeper SSD. Since the low-level feature map is more suitable for precise location, so we add the low-level features to the high-level feature map through the down-sampling module. As shown in **Fig. 5**, two down-sampling modules are added to the Deeper SSD. And the output layers of the down-sampling module act as the prediction layers.

Let us take the down-sampling module 1 as an example. The module 1 takes down-sampling method for the *Conv4_3* layer, and merges the feature map outputted after the down-sampling and the feature map outputted by the *Conv9_2* layer. If we use the *Eltwise* operation to merge these two feature maps, we need the same size and dimension of the two feature maps. Therefore, we need to perform a pooling operation of stride 2 for *Conv4_3*, which can reduce the size of the feature map twice. After the pooling layer is the convolution layer, we use 3x3 or 1x1 kernel size with 256 or 512 outputs. However, the size of feature map outputted by the *Conv9_2* layer is still not large enough for small object detection. Here a little trick is adopted. We take a deconvolution operation of stride 2 for *Conv9_2*, which is made to use the rich semantic information of the *Conv9_2* layer and add the information which is suitable for location in the *Conv4_3* layer to the *Conv9_2* layer. And then we use the *Eltwise* layer to

merge the *Conv4_3* down-sampling output layer and the *Conv9_2* layer. After that, a *ReLU* layer and a *Normalize* layer are added. And the feature map of the down-sampling module can be obtained finally. In a word, such down-sampling module output layer acts as a prediction layer. The number of prediction layers for the entire network is 11.

Considering that the high-level feature map is originally suitable for the object classification, we do not need to increase the number of channels / features, we directly use the context information to improve the information volume of each channel, which can improve the object location performance. The experiment in section 4.4.3 also proves our ideas.

3.2.3 Simultaneous Up-sampling and Down-sampling Method

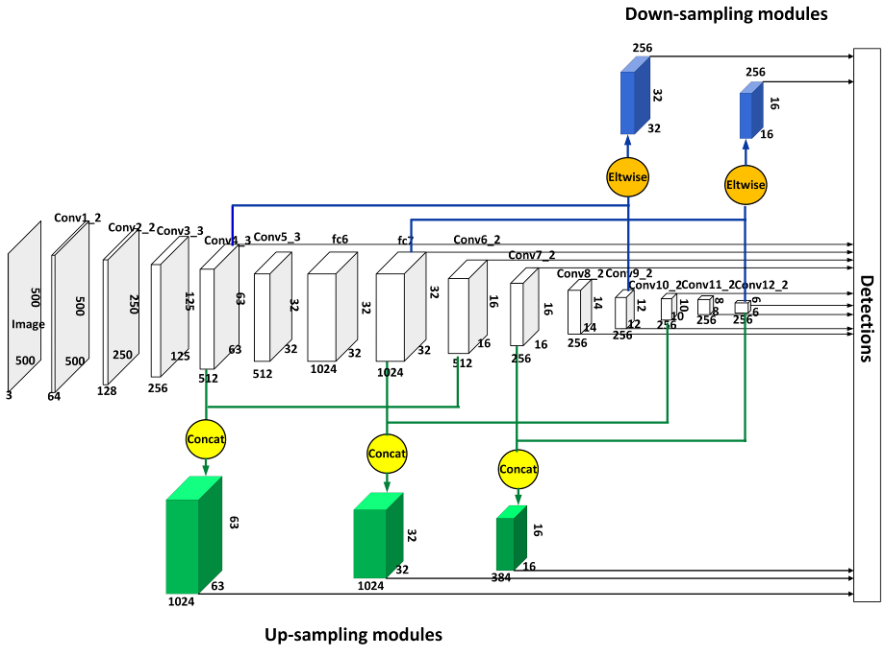


Fig. 6. The Architecture of Simultaneous Up-sampling and Down-sampling Modules

As discussed above, the up-sampling modules increase classification performance, and the down-sampling modules promote location precision. Then the two modules can be added to the Deeper SSD network at the same time, which is shown in **Fig. 6**. The number of prediction layers for the entire network is 14. Here the *Concat* operation is used for the up-sampling modules, while the *Eltwise* operation is used for the down-sampling modules. The intuitive reason is that the up-sampling modules are mainly used to improve the classification performance, which uses the *Concat* operation to increase the number of features, while the down-sampling modules are mainly used to improve the location precision, which uses the *Eltwise* operation to directly utilize context information. We also verify the effectiveness of the *Concat* and *Eltwise* operation through the experiments in section 4.4.

4. Experiments and Analysis

The Deeper SSD and its variants proposed in this paper are evaluated based on drone detection. Here we provide the drone dataset, training stage, evaluation index and experiment results of drone detection in detail.

4.1 Drone dataset

Drone data collection and annotation is essential for the experiments. These drone images are collected from three aspects as shown in **Fig. 7**. One is collected by ourselves, which is captured by surveillance cameras, the two others are collected from [38] and [39]. Nine students spent nearly two months labeling and relabeling all these drone images to create this drone dataset, which includes 32237 images. It can be seen from Figure 7 that the size of drone is too small to be found by human eyes. So drone detection is a challenging task.

(a) Images collected by ourselves (b) Images from [38] (c) Images from [39]

Fig. 7. Example of the drone dataset. The drone objects are inside the yellow bounding box

The analysis of drone's size in this dataset is given in **Table 2**. Here we define the size ratio of the object to the whole image as follow:

$$\partial = object\_size / image\_size \tag{1}$$

In this paper the drone size is sorted according to the value of ¶, which is shown in **Table 2**.

Table 2. The definition of drone size

drone size	the value of ¶	ρ
small (S)	¶£ 0.12%	63.62%
medium (M)	0.12% < ¶£ 1%	36.41%
large (L)	1% < ¶£ 10%	7.28%
super large (XL)	¶> 10%	0.26%

In order to directly illustrate the distribution of drone size in our dataset, we define the ratio of the image number of each type drone size to the image number of the whole dataset as follow:

$$\rho_t = img\_num\_drone\_size_t / img\_num\_whole\_dataset \tag{2}$$

where the value of t represents the type of drone size, and t can be S, M, L, XL.

The ρ in **Table 2** shows that small drones occupy 63.62% among the whole dataset, while medium drones have 36.41% and large ones are 7.54%. Therefore, the dataset focuses on small drones and also considers the variant sizes of drone objects at the same time.

In order to highlight the effectiveness of small object detection, we conduct experiments on the part of small drone objects alone, called as sub dataset S, in which 18684 images are taken as training set and 1825 images are used for validation set. At the same time, to evaluate the performance of proposed method suitable for all size objects, we also do experiments on the

whole dataset W, in which 27236 images belong to training set and 5001 images are used for validation set.

4.2 Training Stage

Our models are trained on the small object dataset S and the whole dataset W. All experiments are performed on a NVIDIA 1080 GPU card with Caffe.

In the experiments, we use the well-trained SSD500x500 as the pre-training model to train the Deeper SSD and its variants. Then the neural network models are fine-tuned based on the training sets of the datasets S and W. For quantitative and qualitative analysis of the proposed neural networks, the basic parameters of all the neural networks are set as follows. max_iter is 150k, $momentum$ is 0.9, $weight\_decay$ is 0.0005, and $batch\_size$ is 8. The learning rate of the neural networks is 10^{-3} for the first iteration and then decreases to 10^{-4}, 10^{-5} and 10^{-6} at 60k, 90k and 120k iterations respectively.

In each iteration of the training stage, the model predicts the category and bounding box of the object, and then updates the network parameters to minimize the loss rate of classification and location. Similar to [15], the loss function is a weighted sum of the localization loss L_{loc} and the confidence loss L_{conf} :

$$L(b,c,l,g)=\frac{1}{N}\left(\alpha L_{loc}(b,l,g)+L_{conf}(b,c)\right) \tag{3}$$

where N is the number of positive samples. $b_{ij}^{p}=\{0,1\}$ is an indicator for matching the i-th default box to the j-th ground truth box of category P. If b is 1, the default box is a positive sample, otherwise, the default box is a negative sample. In our experiment, category P only includes background and drone. The localization loss L_{loc} is a Smooth L1 loss [7] between the predicted box l and the ground truth box g parameters. The confidence loss L_{conf} is the softmax loss over background and drone confidences. The weight term α is set to 1 by cross validation. Similar to Faster R-CNN [7], the localization loss L_{loc} is defined as equation (4):

$$L_{loc}(b,l,g)=\sum_{i\in Positive}^{N}\sum_{m\in\{cx,cy,w,h\}}b_{ij}^{k}\,smooth_{L1}\left(l_i^m-\hat{g}_j^m\right)$$

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

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

$\hat{g}_j^{\mathrm{m}}$ are offsets for the parameters of the j-th default bounding box d . d_i^{m} is the parameters of the i-th default bounding box d . g_j^{m} is the j-th parameters of the ground truth box g . The confidence loss L_{conf} is defined as follows：

$$L_{conf}(b,c)=-\sum\nolimits_{i\in Positive}^{N}b_{ij}^{P}\log\left(\hat{c}_i^{P}\right)-\sum\nolimits_{i\in Negative}\log\hat{c}_i^{0}\ \text{where}\ \hat{c}_i^{P}=\frac{\exp\left(c_i^{P}\right)}{\sum_{P}\exp\left(c_i^{P}\right)} \tag{5}$$

where Positive is the positive sample. Negative is the negative samples。 c_i^P is the confidence of the i-th sample for category P.

In addition, in order to eliminate redundant candidate boxes and find the best object detection location, we use a non-maximal suppression algorithm (NMS). In the experiment, the threshold of NMS is set at 0.5.

4.3 Evaluation Index

As for our model, we adopt five typical indicators to measure our detection performance, namely precision rate, recall rate, PR curve, mAP and detection speed.

PR curve describes the relationship between precision and recall, which takes the X-axis as recall and the Y-axis as precision. The *AP* is the area under the PR curve; here we use the 11-point method to calculate *AP*.

$$AP_{11point} = \frac{1}{11} \sum_{r \in \{0,0.1,\ldots,1\}} \left(p_r \right) \left(p_r = \max \left\{ p_i : r_i \geq r \right\} \right) \tag{6}$$

The r is the recall threshold, and the r is divided into 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0. p_r is the maximum precision with recall greater than or equal to r. The *mAP* is the mean of all *APs*. Since we only detect drones, here m is 1.

$$mAP = \frac{1}{m} \sum\nolimits_{i=1}^{m} \left(AP \right)_i \tag{7}$$

The detection speed is the number of images the model detects per second. It is defined as follows:

$$speed = \frac{the\ total\ image\ of\ detection}{the\ total\ time\ of\ detection} \tag{8}$$

4.4 Experiment Results and Analysis

4.4.1 Deeper SSD

We train the SSD and Deeper SSD on the training sets of datasets S and W, and evaluate them on the validation sets. In order to illustrate the size of input image is closely related to the precision of small object detection, we select 300×300 and 500×500 for the input image size of the SSD, and 500×500 for the Deeper SSD. Here we do not use 300×300 for the Deeper SSD because the feature map size of the *Conv12_2* is zero after convolution.

As shown in **Table 3**, we can see that the *mAP* of SSD500x500 exceeds about 24% and 5% respectively compared with SSD300x300 on the validation set of datasets S and W, which fully illustrates the necessity of enlarging the size of input image.

Compared with the SSD300x300, the precision's promotion of the SSD500x500 on small object validation set S is much higher than that on the whole validation set W. The reason is that the larger input image size is better for small object detection. When the input image size is 300x300, the small object information disappears gradually or even disappears completely in low-level convolution network, and then the network cannot learn small objects at all.

Nevertheless, when the size of input image is 500x500, the information of small objects can be transmitted to the high-level convolution network, and then the network can learn small objects.

When the input image size is fixed at 500×500, our Deeper SSD method achieves better performance on validation sets of data set S and W. On data set S, the *mAP* of our deeper SSD surpasses that of SSD500x500 from 85.61% to 86.91%. Meanwhile on data set W, the result is from 88.93% to 89.56%. Although the detection speed of the Deeper SSD500x500 is a litter lower than that of the SSD500x500, which can still meet the requirements of real-time detection.

Table 3. Detection results of SSD and Deeper SSD on validation sets S and W

neural network	**input size**	***mAP* (%)**		**speed** (FPS)
		Validation Set S	**Validation Set W**	
SSD	300x300	61.52	83.91	31.7
	500x500	85.61	88.93	19.5
Deeper SSD	500x500	**86.91**	**89.56**	18.5

4.4.2 Up-sampling Method

In order to further improve the precision of small object detection, up-sampling method is used to merge high-level features into the low-level feature map. To add up-sampling modules to the neural network in the right way, we try several essential steps.

The first step is selection of the up-sampling module. The second one is the connection way of feature maps. For the selection of the up-sampling module, we adopt three up-sampling modules: module 1, module 2 and module 3. As shown in **Table 4**, these modules are described in detail. Here we conduct independent and combined experiments for the three modules. When one module does not significantly improve the precision of detection, it will not participate the combined attempts certainly. As for the methods of feature maps' connection, we make an attempt on the *Concat* and *Eltwise-sum* operation.

As seen in **Table 5**, compared with *Eltwise-sum* operation, *Concat* operation achieves higher precision on validation set of data set S and W, which shows the effectiveness of *Concat* connection in our up-sampling modules and shows that the appropriate increase of the number of features is beneficial to the object classification performance. Furthermore, these up-sampling modules get higher promotion of precision for small object data set S than the whole data set W, which indicates that up-sampling modules are more conducive to small object detection.

Then the combinations of modules with *Concat* operation are evaluated. The combination of two or three modules can improve the precision of small objects. But for the whole dataset W, the promotion is not obvious. So, for the detection of small and large object simultaneously, we suggest only to use module 1, which can reduce the computation complexity and does not cut down the precision distinctly.

Table 4. The architecture of three up-sampling modules

Up-sampling Modules	**Module 1**		**Module 2**		**Module 3**	
Layers	*Conv4_3*	*Conv6_2*	*Fc7*	*Conv10_2*	*Conv7_2*	*Conv12_2*
Up-sampling Operation		3x3x512*Deconv* 3x3x512*Conv* 3x3x512*Deconv* 3x3x512*Conv* *ReLU*		3x3x512*Deconv* 1x1x512*Conv* *ReLU*		1x1x128*DeConv* 1x1x128*Conv* *ReLU*
Connect Operation	*Concat* *ReLU*		*Concat* *ReLU*		*Concat* *ReLU*	

Table 5. Detection results of the up-sampling modules using different connection methods

Up-sampling Modules		***Concat***							***Eltwise-sum***		
Module 1		√			√	√		√	√		
Module 2			√		√		√	√		√	
Module 3				√		√	√	√			√
***mAP* (%)**	Validation Set S	88.17	88.32	88.12	88.50	88.52	88.31	**88.72**	87.93	87.94	88.09
	Validation Set W	**89.97**	89.69	89.69	**89.97**	**89.97**	89.73	**89.97**	89.70	89.68	89.69

4.4.3 Down-sampling Method

To further improve the detection precision of small drones, down-sampling method is added to the Deeper SSD to merge low-level features into the high-level feature map. Here we still try several necessary steps, such as selection of the down-sampling modules and the connection way of feature maps.

Table 6. The architecture of two down-sampling modules

Down-sampling Modules	**Module 1**		**Module 2**	
Layers	*Conv4_3*	*Conv9_2*	*Fc7*	*Conv12_2*
Down-sampling Operation	3x3x256*Conv* 1x1x512*Conv* *Pool* 1x1x256*Conv* *ReLU*	1x1x256*Deconv* *ReLU*	3x3x512*Conv* 1x1x256*Conv* *Pool* 1x1x256*Conv* *ReLU*	1x1x256*Deconv* *ReLU*
Connect Operation	*Eltwise-sum* *ReLU* *Normalize*		*Eltwise-sum* *ReLU* *Normalize*	

For the selection of the down-sampling modules, we consider two down-sampling modules, module 1 and module 2. These two modules are shown in detail in **Table 6**. We also do independent and combined experiments for these two modules. Here we adopt the *Concat* and *Eltwise-sum* operations for the way of feature maps' connection too.

As seen in **Table 7**, the proposed down-sampling modules achieve higher precision by using *Eltwise-sum* operation than *Concat* operation on validation sets of data set S and W, which shows the effectiveness of *Eltwise-sum* connection and shows that the direct use of context information can improve the object location performance.

So the combinations of modules with *Eltwise-sum* operation is evaluated. Seen from the result shown in **Table 7**, the combination of module 1 and module 2 gets higher precision. Therefore, the proposed method adopts the down-sampling method using the combination of module 1 and module 2 with *Eltwise-sum* connection method.

Table 7. Detection results of the down-sampling modules using different connection methods

Down-sampling Modules		***Concat***		***Eltwise-sum***		
Module 1		√		√		√
Module 2			√		√	√
mAP **(%)**	Validation Set S	87.77	88.16	88.10	88.32	**88.43**
	Validation Set W	89.56	89.58	89.75	89.73	**89.92**

4.4.4 Simultaneous Up-sampling and Down-sampling Method

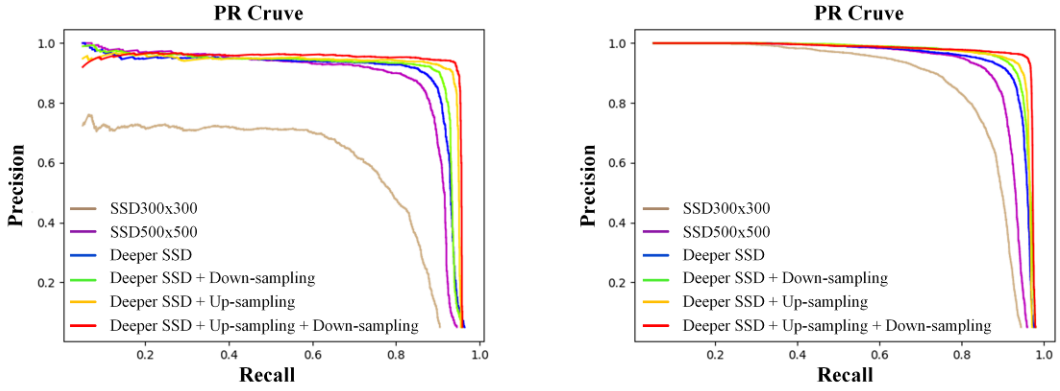
Using up-sampling or down-sampling methods for the Deeper SSD respectively, the precision of small object detection is improved to a certain extent as mentioned above. Meanwhile, we conduct the experiment to conbine the up-sampling and the down-sampling simultaneously for the Deeper SSD as well.

Table 8. The detection results of Deeper SSD using different methods on the validation sets

Method	***mAP*** **(%)**		**speed (FPS)**
	Validation Set S	Validation Set W	
SSD300x300	61.52	83.91	31.7
SSD500x500	85.61	88.93	19.5
Deeper SSD	86.91	89.56	18.5
Deeper SSD + Up-sampling	88.72	89.97	19.2
Deeper SSD + Down-sampling	88.43	89.92	17.5
Deeper SSD + Up-sampling + Down-sampling	89.02	90.15	17.6

As shown in **Table 8**, for validation set of small drones set S, the mAP of simultaneous up-sampling and down-sampling is 89.02%, which is increased 3.41% and 2.11% comparing with the SSD500x500 and the Deeper SSD. It is also higher than that of up-sampling method or down-sampling method, which are 88.72% and 88.43% respectively. For validation set of the whole data set W, the mAP of simultaneous up-sampling and down-sampling also surpasses that of up-sampling and down-sampling independently from 89.97% and 89.92% to 90.15%, and also improves 0.59% comparing with that of the Deeper SSD. Although the speed of the Deeper SSD with simultaneous up-sampling and down-sampling is a little lower than that of SSD500x500, it can still meet the requirements of object detection under many scenarios.

Fig. 8 demonstrates the PR curves of SSD, Deeper SSD and its variants. **Fig. 9** (a) is based on the validation set of the small object data set S, and **Fig. 9** (b) is given for the validation set of the whole data set W. The red curve is the PR curve of the Deeper SSD with simultaneously up-sampling and down-sampling. And it is the optimal curve whether it is in the validation set of small object data set S or the whole data set W. At the same time, the Deeper SSD with up-sampling is better than the that with down-sampling, and they are all better than the Deeper SSD. In sequence, the Deeper SSD is better than the SSD500x500. And that of the SSD300x300 is the worst.



(a) Validation set S (b) validation set W

Fig. 8. PR curves of SSD, Deeper SSD and its variants on validation set S and W

In order to illustrate the effectiveness of our proposed methods, we also conduct experiments using traditional algorithms, such as Faster R-CNN, YOLOv2, YOLOv3, YOLT. As shown in **Table 9**, we can find that the precision of SSD500x500 is higher than Faster R-CNN and YOLOv2. Although the detection precision of SSD500x500 is slightly lower than that of YOLOv3 on the validation set of small object data set S, the detection precision of SSD500x500 surpasses that of YOLOv3 about 3% on the validation set of the whole data set W. Therefore, we choose SSD500x500 as our backbone network. The precisions of the Deeper SSD and its variants are higher than Faster R-CNN, YOLOv2 and YOLOv3, which fully shows the effectiveness of our proposed methods.

Table 9. The detection results with different object detection algorithms

Method	Input Size	*mAP* (%)	
		Validation Set *S*	Validation Set *W*
Faster R-CNN	224x224	31.47	59.87
YOLOv2	416x416	73.94	63.05
YOLOv3	416x416	85.75	86.34
YOLT [21]	416x416	47.79	66.39
SSD	300x300	61.52	83.91
SSD	500x500	85.61	88.93
Deeper SSD	500x500	86.91	89.56
Deeper SSD500 + Up-sampling	500x500	88.72	89.97
Deeper SSD500 + Down-sampling	500x500	88.43	89.92
Deeper SSD500 + Up-sampling + Down-sampling	500x500	89.02	90.15

5. Conclusion

This paper proposes a Deeper SSD neural network with simultaneously up-sampling and down-sampling methods to detect small objects, especially drones. The Deeper SSD network enlarges the input image's size and adds deeper convolutional layers to improve the precision of small object detection. In order to further improve the precision of small object recognition, the Deeper SSD uses simultaneous up-sampling and down-sampling. The operation of up-sampling adds high-level semantic information to the low-level feature map to improve the performance of classification. And the operation of down-sampling adds rich low-level local information to the high-level feature map to improve the precision of object location.

Based on the self-designed data set, experiments demonstrate the effectiveness of the Deeper SSD and its variants, which are useful to small drone's detection and recognition. These proposed methods can also detect small and large objects at the same time. Meanwhile, our proposed method can also be adopted to other object detection neural networks, such as Faster R-CNN and YOLOv3. Our methods can be extended to detect more other small objects, not just small drones. However, we also know that our proposed approach still leaves much to be improved. For example, the detection performance of general object and the inference time still can be improved, and we will continue to extend this work in the future.

Acknowledgments

The authors would like to thank the handling associate editor and all the anonymous reviewers for their constructive comments. This research was supported by the Fundamental Research Funds for the Central Universities (No. NS2016091).

Reference

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the Seventh IEEE International Conference on Computer Vision on Computer Vision*, vol. 99, no. 2, pp. 1150-1157, 1999. Article (CrossRef Link)

[2] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. of European conference on computer vision*, pp. 428-441, 2006. Article (CrossRef Link)

[3] T. Mita, T. Kaneko, and O. Hori, "Joint haar-like features for face detection," in *Proc. of Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, pp. 1619-1626, 2005. Article (CrossRef Link)

[4] J. Cho, S. Mirzaei, J. Oberg, and R. Kastner, "Fpga-based face detection system using haar classifiers," in *Proc. of the ACM/SIGDA international symposium on Field programmable gate arrays*, pp. 103-112, 2009. Article (CrossRef Link)

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 580-587, 2014. Article (CrossRef Link)

[6] R. Girshick, "Fast r-cnn," in *Proc. of the IEEE international conference on computer vision*, pp. 1440-1448, 2015. Article (CrossRef Link)

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards realtime object detection with region proposal networks," *Advances in neural information processing systems*, pp. 91-99, 2015. Article (CrossRef Link)

[8] K. He, G. Gkioxari, P. Dolla´r, and R. Girshick, "Mask r-cnn," in *Proc. of the IEEE international conference on computer vision*, pp. 2980-2988, 2017. Article (CrossRef Link)

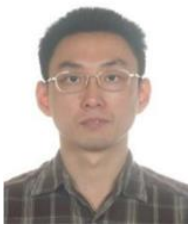
[9] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 6154-6162, 2018. Article (CrossRef Link)
[10] Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, and Y. Fu, "Rethinking Classification and Localization for Object Detection," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10183-10195, 2020. Article (CrossRef Link)
[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 779-788, 2016. Article (CrossRef Link)
[12] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 6517-6525, 2017. Article (CrossRef Link)
[13] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018. Article (CrossRef Link)
[14] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020. Article (CrossRef Link)
[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. of European conference on computer vision*, vol. 9905, pp. 21-37, 2016. Article (CrossRef Link)
[16] C.Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017. Article (CrossRef Link)
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904-1916, 2015. Article (CrossRef Link)
[18] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: Towards accurate region proposal generation and joint object detection," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 845-853, 2016. Article (CrossRef Link)
[19] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013. Article (CrossRef Link)
[20] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proc. of the IEEE international conference on computer vision*, pp. 9627-9636, 2019. Article (CrossRef Link)
[21] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proc. of the IEEE International Conference on Computer Vision*, pp. 6569-6578, 2019. Article (CrossRef Link)
[22] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region proposal by guided anchoring," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2960-2969, 2019. Article (CrossRef Link)
[23] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image and Vision Computing*, vol. 97, 2020. Article (CrossRef Link)
[24] S. Qi, J. Ma, C. Tao, C. Yang, and J. Tian, "A robust directional saliency-based method for infrared small-target detection under various complex backgrounds," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 3, pp. 495-499, 2012. Article (CrossRef Link)
[25] I. El-Naqa, Y. Yang, M. N. Wernick, N. P. Galatsanos, and R. Nishikawa, "Support vector machine learning for detection of microcalcifications in mammograms," in *Proc. of IEEE International Symposium on Biomedical Imaging*, pp. 201-204, 2002. Article (CrossRef Link)
[26] Y. Lu, T. Javidi, and S. Lazebnik, "Adaptive object detection using adjacency and zoom prediction," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2351-2359, 2016. Article (CrossRef Link)
[27] A. Van Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," *arXiv preprint arXiv:1805.09512*, 2018. Article (CrossRef Link)

[28] G. Cao, X. Xie, W. Yang, Q. Liao, G. Shi, and J. Wu, "Feature-fused SSD: fast detection for small objects," in *Proc. of Ninth International Conference on Graphic and Image Processing (ICGIP 2017)*, vol. 10615, p. 106151, 2018. Article (CrossRef Link)
[29] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015. Article (CrossRef Link)
[30] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1951-1959, 2017. Article (CrossRef Link)
[31] M. Xu, L. Cui, P. Lv, X. Jiang, J. Niu, B. Zhou, and M. Wang, "MSSSD: Multi-scale deconvolutional single shot detector for small objects," *arXiv preprint arXiv:1805.07009*, 2018. Article (CrossRef Link)
[32] Z. Chen, D. Crandall, and R. Templeman, "Detecting small, densely distributed objects with filter-amplifier networks and loss boosting," *arXiv preprint arXiv:1802.07845*, 2018. Article (CrossRef Link)
[33] K. Fu, J. Li, L. Ma, K. Mu, and Y Tian, "Intrinsic Relationship Reasoning for Small Object Detection," *arXiv preprint arXiv:2009.00833*, 2020. Article (CrossRef Link)
[34] Z. Liu, G. Gao, L. Sun, and Z. Fang, "HRDNet: High-resolution Detection Network for Small Objects," *arXiv preprint arXiv:2006.07607*, 2020. Article (CrossRef Link)
[35] F. Zhang, L. Jiao, L. Li, F. Liu, and X. Liu, "MultiResolution Attention Extractor for Small Object Detection," *arXiv preprint arXiv:2006.05941*, 2020. Article (CrossRef Link)
[36] Y. Wu, Y. Sui, and G. Wang, "Vision-based real-time aerial object localization and tracking for uav sensing system," *IEEE Access*, vol. 5, pp. 23969-23978, 2017. Article (CrossRef Link)
[37] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting flying objects using a single moving camera," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 5, pp. 879-892, 2016. Article (CrossRef Link)
[38] Y. Chen, P. Aggarwal, J. Choi, and C. C. Jay, "A deep learning approach to drone monitoring," in *Proc. of 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) IEEE*, pp. 686-691, 2017. Article (CrossRef Link)
[39] C. Aker and S. Kalkan, "Using deep networks for drone detection," in *Proc. of 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1-6, 2017. Article (CrossRef Link)
[40] H. Sun, J. Yang, J. Shen, D. Liang, L. Ning-Zhong, and H. Zhou, "TIB-Net: Drone Detection Network with Tiny Iterative Backbone," *IEEE Access*, vol. 8, pp. 130697-130707, 2020. Article (CrossRef Link)
[41] Y. Hu, X. Wu, G. Zheng, and X. Liu, "Object detection of UAV for antiUAV based on improved YOLO v3," in *Proc. of China Control Conference (CCC)*, pp. 8386-8390, 2019. Article (CrossRef Link)

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

Wen Geng received the B.S. degree in China University of Mining and Technology, Xuzhou, China in 2017. She is currently pursuing the master's degree in Nanjing University of Aeronautics and Astronautics, Nanjing, China. Her research interests include deep learning, computer vision and small object detection.

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



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

Dong Liang received the B.S. degree in Telecommunication Engineering and the M.S. degree in Circuits and Systems from Lanzhou University, China, in 2008 and 2011, respectively. In 2015, he received Ph.D. at Graduate School of IST, Hokkaido University, Japan. He is now an Assistant Professor at the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His research interests include computer vision and pattern recognition.

Huiyu Zhou received the B.S. degree in Radio Technology from Huazhong University of Science and Technology of China and a Master of Science degree in Biomedical Engineering from University of Dundee of United Kingdom, respectively. He was awarded a Doctor of Philosophy degree in Computer Vision from Heriot-Watt University, Edinburgh, United Kingdom. He is currently a professor of the School of Informatics, University of Leicester, UK. His research interests include machine learning, computer vision and artificial intelligence.