An Improved Convolutional Neural Networks for Diseased Plant Detection on Unmanned Aerial Vehicle Image

Dashuang Liang[1], Wenping Liu[1] *, Yu sun[1], Lei zhao[1] and Youqing Luo[2]
1 School of Information Science and Technology, Beijing Forestry University, Beijing, 100083, China
2 Beijing Key Laboratory for Forest Pest Control, Beijing Forestry University, Beijing, 100083, China
* Correspondence: wendyl@vip.163.com

Abstract: This paper represents a diseased plants detection approach that can accurately localize the bounding box of plants on unmanned aerial vehicle images. Till now, MobileNet neural network combined with SSD (Single Shot Multibox Detetor), named MobileNetV2 SSDLite (MNet V2+SSDLite) is one of the best object detection algorithms with both high accuracy and fast speed. This paper proposes an enhanced MobileNetV2 SSDLite detector that can significantly improve the performance with just a little speed drop and computation cost increase. Under the hold-out test set, the results show that the proposed diseased plant detection approach achieves an overall detection accuracy of 45.1% by COCO challenge metrics (AP at IoU=0.50:0.05:0.95), while the standard MobileNetV2 SSDLite only achieves 39.2%, when compares to standard MobileNetV2 SSDLite, and the model parameter evaluated by multiply-adds (MAdd) is 1.24G to 947.62M, while the number of parameter is 5.25 to 3.91M. This research indicates that the proposed deep learning detection model provides a better solution in detecting diseases plants with high accuracy and a relatively fast speed.
Keywords: Real-time Detector; Deceased Plant Detection; Convolutional Neural Network.

1. Introduction

Forests play an important role in a country's economic, social and environmental benefits [1], and plant diseases and pests threaten the growth of forests. Traditionally, the severity and extent of plant diseases and pests are manually scored and counted by field investigations with expensive cost and low efficiency [2]. Therefore, an accurate and a faster detection of diseases and pests in plants is very necessary, which could help to develop an early treatment technique with substantially reducing economic losses [3]. Luckily, with the rapid technology development of unmanned aerial vehicle (UAV), it is cheap and fast to get high-resolution images of forest distribution. Based on these images, people can design image processing method to detect the distribution of plant diseases and pests [1]. In the past, spectral detection technology with satellite remote sensing data or traditional computer vision method fused global positioning system. This system and geography information system are used in area detection of pests and achieved relatively good results [4]. However, their efficiency is always very low, and fail to accurately locate the single infected plants. In these years, recent advances in hardware technology have allowed the evolution of deep convolutional neural networks and their large number of applications, including complex tasks such as object recognition and image classification [3]. Among the various approaches, deep learning has played an important role and achieved a great success in computer vision (i.e., image classification [5-7], face recognition [8, 9], segmentation [10, 11]). Nearly all state-of-the-art applications are based on deep learning. In recent years, deep learning has become more and more widely used in detection [9, 12, 13]. Among all these deep-learning algorithms, detection with SSD (Single Shot MultiBox Detector) has been shown to achieve impressive results on the PASCAL VOC [14], COCO [15], and ILSVRC [16] datasets with its high efficiency and accuracy. As the accuracy of deep learning model has already reached very high, one general trend is to make smaller and faster networks models to suit many real world applications, that is enable the model to run timely on a computationally limited platform without much accuracy decease. For example,

models including ShuffleNet [17], MobileNetV1 [18] and MobileNetV2 [19] have been used widely.

Forest diseases and pests detection and segmentation based on remote sensing imagery has been studied worldly. Liu et al., (2017) proposed a novel CNN-based model based on AlexNet to identify four common apple leaf diseases. The model achieves an overall accuracy of 97.62%, and the parameters are reduced by 51,206,928 compared with those in the standard AlexNet model [20]. Näsi et al., (2015) developed a new processing method with high spatial resolution photogrammetric and hyperspectral images to identify pest damage at tree-level, and the results of classification of trees into healthy, infested and dead classes were promising [21]. Lehmann et al., (2015) used a modified Normalized Difference Vegetation Index (NDVImod) derived classification method on the color infrared (CIR) / near infrared (NIR) image to classify the vegetation health and obtained an overall Kappa Index of Agreement (KIA) of 0.81 [22]. With the development of UAV technology, UAV imaging with machine learning has been applied in forest diseases and pests detection. Yuan et al., (2016) calculated a 12-dimension statistical texture information for each super pixel in UAV imaging, and then used the random forest and object-based classification method to extract forest pest areas. The result showed that the method was effective [23]. Dash et al., (2017) built a non-parametric approach to model physiological stress with the time-series multi-spectral imaging acquired by UAV and provided good classification with kappa of 0.694 [1]. Meddens et al., (2013) compared maximum likelihood classification based on single-date and time series of spectral indices based on multi-date, and found that multi-date method was more accurate at intermediate levels of tree mortality, while single-date method was more accurate at high mortality [24]. Michez et al., (2016) used multi-temporal hyper spatial UAV imaging to characterize riparian forest species and health condition, in order to identify which type of variables is derived from hyper spatial imaging and which scale of image analysis is the most relevant to reach our objectives [25]. However, these studies seldom apply the CNN-based model to detect pest forest area, not to mention improving the deep learning model to optimize its efficiency. Therefore, a real-time novel CNN-based model to accurately and timely detect diseased forest plant is proposed in this paper.

A new real-time and robust deep-learning-based detector to rapidly and effectively extract the distribution and area of the diseased plants with UAV imagery is introduced in this paper. It is an enhanced MobileNetV2 SSDLite detector which nearly retains its simplicity and significantly improves its accuracy, achieving state of the art on diseased plants image detection tasks for mobile application. A general overview of the system is presented in Figure 1.

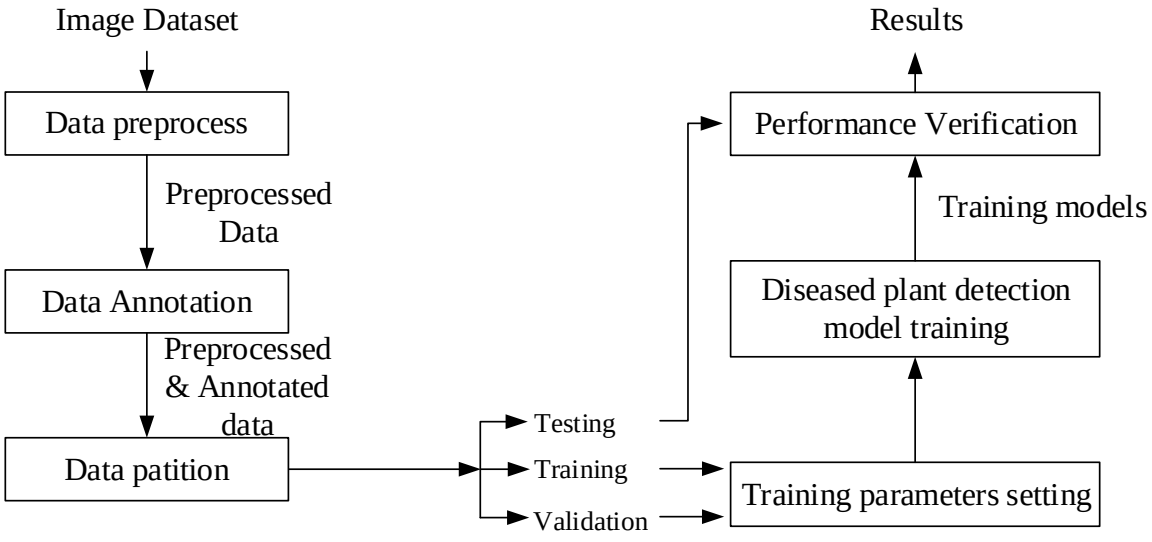


Figure 1. Overview of the proposed deep-learning method for deceased plant detection

The main contributions of this paper are summarized as follows:

A real-time detecting convolutional neural network is proposed and first employed to extract the area of plant diseases and pests on UAV imaging. By analyzing the characteristics of diseased and pest plant in images, a novel deep convolutional neural network model based on MobileNetV2 SSDLite is proposed. Firstly, a lower layers feature map is added as the location and confidence predictor, it can improve the ability of detecting small objects. Secondly, more default boxes are associated with the layer of expanded_conv_13, it is also beneficial to the small objects' detection. At last, an atrous convolution architecture combined with mobilenetV2 block is used to create feature map for predictor at the bottom of the network, this design can get dense feature map for predictor with high efficiency.

A transplanted model is successfully finished which can work on mobile phone or other embedded devices with low computation cost. With the input image of size 1000 × 1000, it can detect all the infected plants on it in about 1.4 seconds on the phone with cpu of valon 625 (MSM8953), it can also be transplanted to any other mobile device, such as the camera on UAV.

At last, this study also gives an advice about the flying height of UAV that can get relative high accuracy of infected trees detection, and it can be taken as an advice when doing forest filed investigations with UAV.

2. Materials and Methods

2.1. Study area

The study area is located in Lingyuan city, Liaoning province, which is in the Northeast of China (40°35'50" - 41°26'6"N, 118°50'20" - 119°37'37"E). Lingyuan is a city in the west of Liaoning province in Northeast of China, bordering Hebei province and Inner Mongolia. It lies 93.3 kilometers long from south to northeast, and 66.1 kilometers wide from east to west. Its perimeter is 318 kilometers, with a total area of 3278 km^2, accounting for 2.2% of the total area of the province.

This place is mostly dominated by Chinese pine (*Pinus tabuliformis*) mixed with a few of poplar (*Populus*) in it. Dendroctonus (*Scolytidae*) has caused damage and tree mortality within the area. In order to quickly detect the disaster level of the forest, 5 square plots (30 m × 30 m) are established as the sample plots in this area. They are named as site 1, site 2, site 3, site 4 and site 5. The corner of each sample plot was located by a GPS device. Figure 2 shows the location and the distribution of these plots on map.

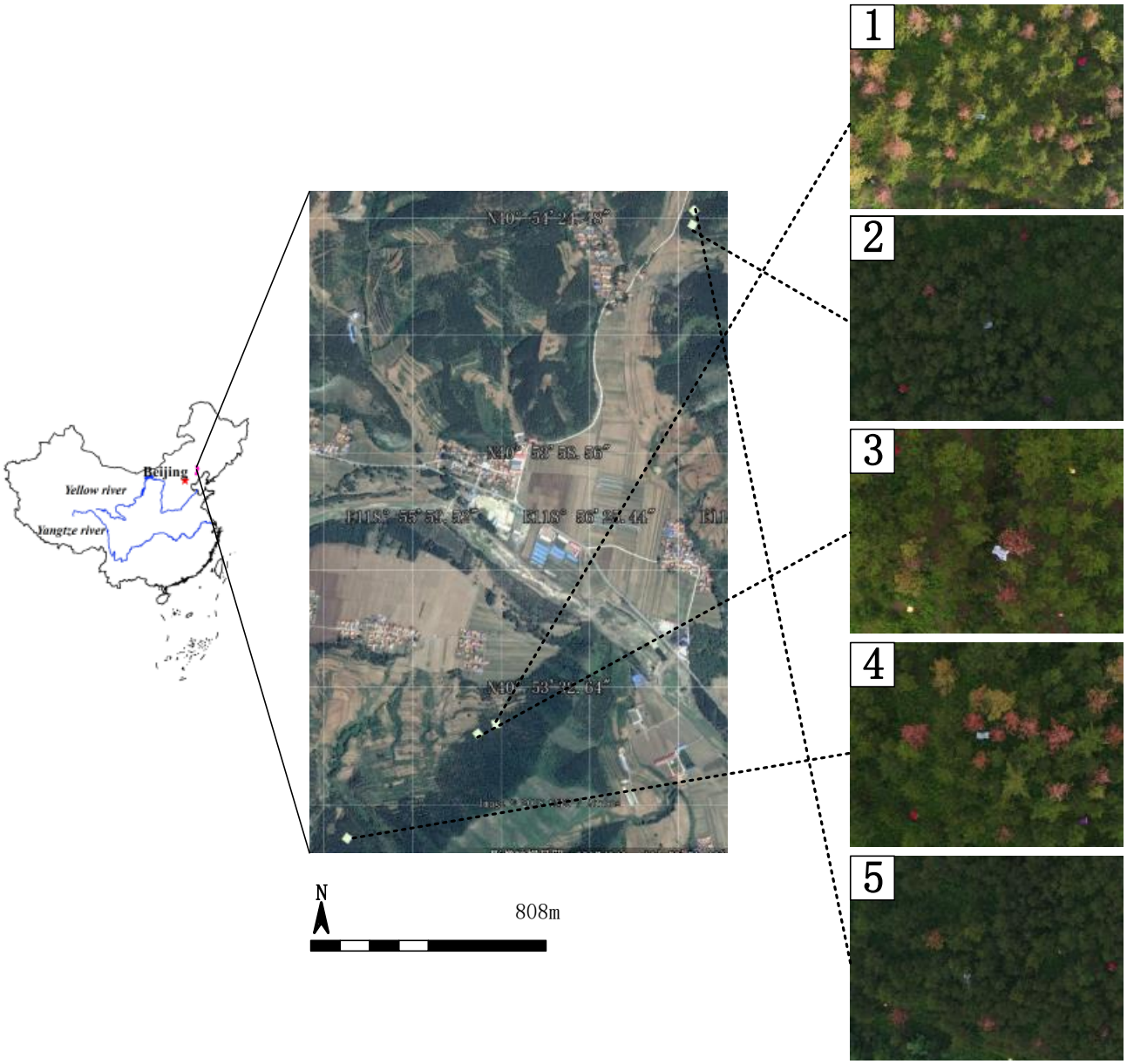


Figure 2. The study area

2.2. Data Collection

Because one of the main objectives of the study is to detect forest health condition, the aerial data survey is planned for the late growing season, which is the best time window for catching the leaf symptoms. Therefore, our UAV-based data acquisition was carried out in the study areas on 11 August and 12 August 2017. The UAV model is four rotor Dji inspire2, carrying a dji x5 professional camera with an effective resolution of 2×10^7 pixels, providing an image size of 5280 × 3956 pixels. It was windless during the flights. In order to capture images without exposures and shadows, the capture time when the sunlight was too strong or too weak is avoided. In reality, for site 1, besides dataset captured in the normal conditions, we also got a set of pictures with strong sunlight.

In every sample site, UAV started from the altitude of 25 m or 30 m, trying to take at least one picture with the step of 5 m as it gone up about 200 m high. When it gone down, it also tried to take one or more pictures at each 5 m. Because the maximum height was limited by the battery condition during the capture period, the number of final captured pictures were different in every site. For site 1, it has 70 pictures with strong sunlight, and 88 pictures with normal conditions separately, while site 3

has 58 pictures and site 4 had 61 pictures.

In addition, as to site 2 and site 5, besides capturing pictures from different height with a certain center as described above, some pictures were also captured at a certain height with route scanning format.

2.3. Architecture of detection network

Base network the proposed enhanced MobileNetV1-SSD detector is derived from standard MobileNetV2 SSDLite [19]. The standard MobileNetV2 SSDLite follows the methodology of SSD. It contains 1 standard convolution and 17 depthwise separable convolutions (depthwise followed by 1×1 projection). Conv_11 and conv_13, appending with four additional convolutional layers with spatial resolution decaying by a factor of 2 with depths 512, 256, 256, 128 respectively as the predictor [26]. It is an efficient model for objects detection with both high accuracy and fast speed. With the willingness to build a more efficient network in diseased plants detection, some new designs are made on the base of MobileNetV2 SSDLite. The details are as follows:

One lower layers feature map is added as a predictor. Firstly, besides using expanded_conv_13, conv_1, layer_19_2_Conv2d_2_3x3_s2_512, layer_19_2_Conv2d_3_3x3_s2_256, layer_19_2_Conv2d_4_3x3_s2_256 and layer_19_2_Conv2d_5_3x3_s2_128 as the location and confidence predictor, an predictor connected to expanded_conv_10 layer which is at a lower level of network is added for detection, because low layer can capture more detailed information of the images. It is benefit to the small objects detection. The feature map resolution in this layer is 38 × 38 and the size of the default box is computed as follows:

Algorithm 1 The method of create default boxes.

Input: an array of aspect ratios AR, $minS$, maxS
Output: $candidateList$

for each ar in AR **do**
  if $ar = 1$ **then**
    $bw \leftarrow MinS$
    $bh \leftarrow MinS$
    add bw,bh to $candidateList$
    $bw \leftarrow sqrt(MinS * MaxS)$
    $bh \leftarrow sqrt(MinS * MaxS)$
    add bw,bh to $candidateList$
  else
    $bw \leftarrow MinS * sqrt(ar)$
    $bh \leftarrow MinS/sqrt(ar)$
    add bw,bh to $candidateList$
  end if
end for

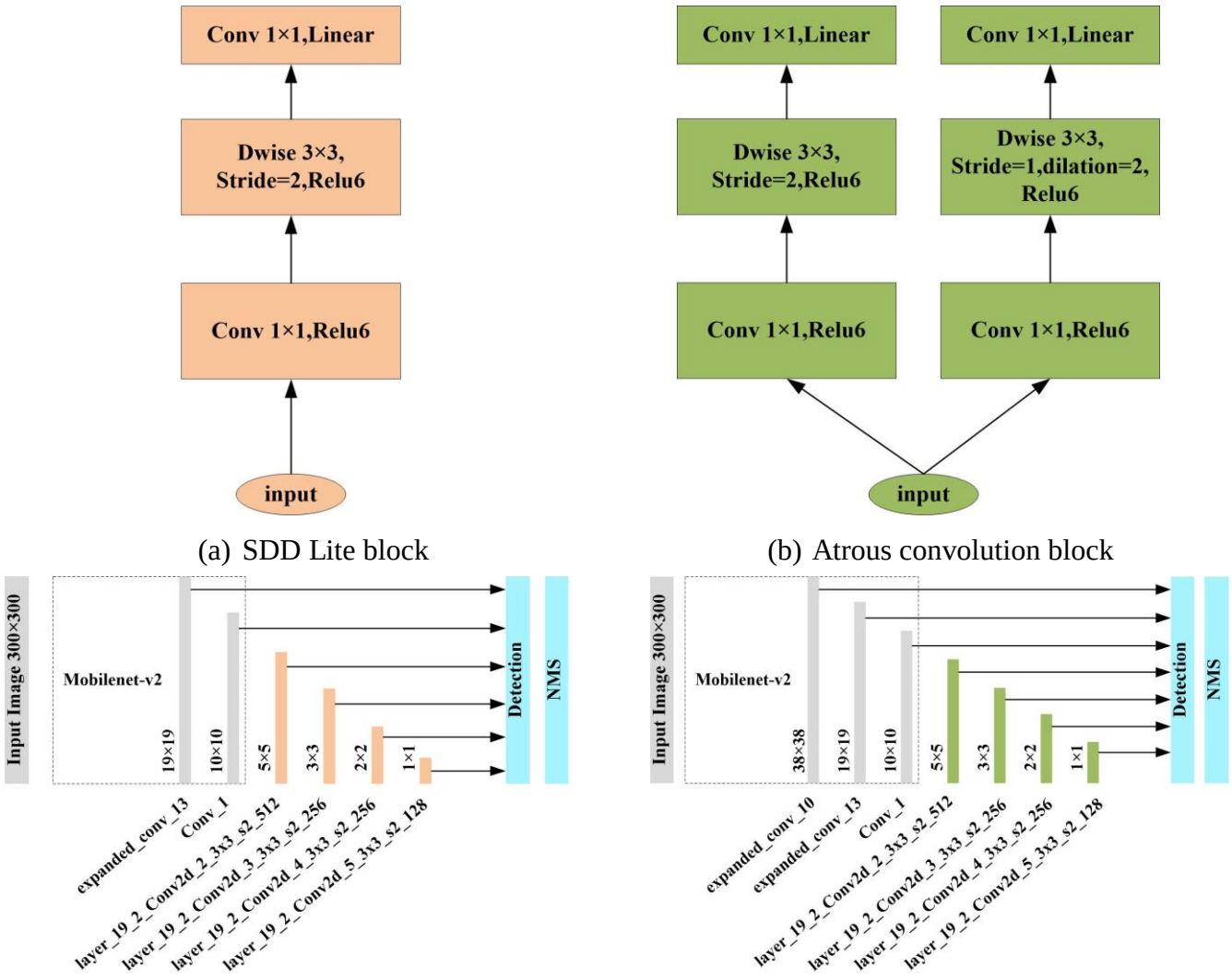
In this layer, the minS is set to 30, maxS is 60 in this layer, and four aspect ratios compose an array of $[1, 2, 1/2, 1/3]$. At last, from the computation method above, 6 default boxes will be created for each cell of this feature map.

More default boxes 6 different default boxes are associated with the layer of expanded_conv_13. On the contrary, in original MobileNetV2 SSDLite, it is only associated with 3 default boxes. More default boxes will improve the detection accuracy, although it will also increase the compute efficiency and greatly reduce memory. As the expanded_conv_13 layer is a relative low layer in the Mobilenet v2 network architecture, it also has more detailed information of the original input images. So adding more default box in this layer is also benefit to the small objects detection.

For a cell in a feature map, c class scores and the 4 location offset values are computed for each box, then the number of filters results in its feature map is:

$$N = (c+4)*k*m*n$$

m and n is the size of feature map, k is the number of default boxes, c is the class number. When k changes from 3 to 6, it will yield twice of filters than the original as to this layer.



(a) SDD Lite block (b) Atrous convolution block

(c) MobileNetV2 SSDLite (d) Our enhanced MobileNetV2 SSDLite

Figure 3. Comparison of MobileNetV2 SSDLite model architecture and ours

Atrous convolution to create feature map for predictor in the original SSDLite model, a prediction layer is one regular 1 × 1 convolution combined with a separable convolution (depthwise followed by 1 × 1 projection), like the figure 3(a) shows. As atrous convolution allows to keep the resolution of feature map unchanged while enlarging the field of view of filters without increasing the number of parameters or the amount of computation [27], it is considered to add in the prediction layers. After some experiments, for one prediction layer, besides of keeping the regular prediction layer, a regular 1 × 1 convolution with an atrous and depth wise convolution followed by a pointwise convolutional architecture is added to efficiently produce dense feature maps for the detection predictor, this newly designed architecture is shown in figure 3(b). It can not only keep the benefit of regular SSDLite model, but also the benefit of depth wise and atrous convolution operations which can help to get dense feature map for predictor with high efficiency.

In order to decrease the computation cost and parameters, there are only four additional atrous convolution blocks are added in this network, starting from “layer_19_1_Conv2d_2_1x1_256” with its feature map size of 10 × 10 till to the layer “layer_19_1_Conv2d_5_1x1_64” with the feature map size of 1 × 1. The detailed architecture of this block is shown in Figure 3(b).

Overall, the proposed network is an enhanced MobileNetV2 SSDLite. 11 various feature maps are used to combine with the location and class confidence predictors, while the MobileNetV2 SSDLite only has 6 various feature maps connected to the predictor. What’s more, the number of default boxes is 12492 compares to 1917 in MobileNetV2 SSDLite. The details of the differences between MobileNetV2 SSDLite and our network architecture is shown in Figure 3(c) and Figure 3(d).

2.4. Experiments

2.4.1. Data preparation

In the fieldwork, the green color is classified as healthy plant, while those with yellow color are taken as plant with infected stage, and the red is serious infected plants, which are the dead plants. In this study, we only focus on diseased plant detection; that is to say, our detector only cares about the plants with yellow or red color. As the original size of images is 5280 × 3956 pixels, which is too big to train a network, hence a set of image tiles are created by cropping each original aerial image using a sliding window with size of 1000 × 1000 pixels and stride of 1000 pixels. In this way, one big aerial image can be split into 24 small images. Before training, the images of site 1, site 3 and site 4 are split into training and validation dataset, while the images of site 2 and site 5 are chosen as testing dataset. Finally, the training and validation datasets contain 5281 tiles and 1319 tiles respectively, and the test dataset contains 2306 images. All the images are manually labeled with ground truth bounding boxes and assigned class labels infected or dead (only one per bounding box).

In this study, the training data contains about 15792 object instances, including 2433 instances for the infected category and 13539 instances for the dead category, while the test dataset has 102 infected instances and 603 dead instances respectively.

2.4.2. Training

As Caffe [28] is one of the most famous and fastest deep learning frameworks for convolutional neural networks, it is used to train models on our GPU machine in this experiment. For the backbone of the network architecture proposed in the essay is the same with MobileNetV2 SSDLite, and at the beginning of training, the network is fine-tuned from an initialization with the pre-trained MobileNetV2 SSDLite model on PASCAL VOC dataset and trained in an end-to-end manner.

Table 1. Parameters of network training

Mini-batch size	16
max_iter:	300000
lr_policy	multistep
stepvalue:	40000, 80000, 100000
Initial learning rate	0.00025
Weight decay	0.00005
gamma:	0.5

In all these experiments, images are resized to a fixed shape with 300 × 300 pixels, the class number is 3 including the background class, and every image is augmented just as the SSD does before training. As Lmdb file format is the common file format for Caffe framework, it is chosen in the training experiments as the input file format. Firstly, the pixel values of every image is normalized to

the range of -1 to 1, because this operation can enable or accelerate the speed of convergence. After that, images with their label are packed in the lmdb file. The network are trained on a single NVIDIA Titan 12GB GPU. The training was stopped after 90 epochs, which took roughly 11 days. RMSProp learning method is used as the gradient descent algorithm. The detailed training hyper-parameters are listed in Table 1.

2.4.3. Test and comparison

To make a full comparison with other models, original SSD and standard MobileNetV2 SSDLite are also trained and then are tested and compared with this model. Moreover, as the former version of mobile models named MobileNetV1 is proposed by Howard et al., (2017) [18], its application to objects detection called MobileNetV1 SSD also shows to have good performance, it is also be used as the comparative model in this experiment. All the models are trained with the same dataset and the same training hyper-parameters, and then be tested with the same test dataset.

To qualitatively test the final detection accuracy of this model, COCO challenge metrics (AP at IoU = 0.50:0.05:0.95) is used as the performance indicator. The original input images' size is 1000, and the input resolution of all the models is 300. The number of parameters which indicate the size of a model, and the number of operations measured by multiply-adds (MAdd) that denote the compute cost are also computed.

In order to further analysis our model, test dataset are spited into 3 parts, they are named as large, medium and small dataset classified through objects' scale in images. The big one is images with object scales larger than 300 × 300 pixels, the small one is images with their object sizes smaller than 200 × 200 pixels, and the medium one' images have object areas that are between 300 × 300 pixels and 200 × 200 pixels. All the models are tested with these three datasets.

3. Results

3.1. Comparison of different models

The results of different models are shown in Table 2. It can be clearly seen from the table that enhanced MobileNetV2 SSDLite (ours) performs really well on both classes, while the computation efficiency are relative high and parameters are relative small. Thought SSD performs really well in mAP (total), but its computation efficiency is too slow and parameters are too big to do real time detection work. In contrast, the mAP (total) gap between enhanced MobileNetV2 SSDLite and SSD is 4.9%, while the Multiply-Adds are only about thirtieth of SSD, and the number of parameters is just about a quarter of SSD. Notably, the enhanced MobileNetV2 SSDLite model can outperform both MobileNetV2 SSDLite and MobileNetV1 SSD with a great improvement. The comparison between MobileNetV2 SSDLite and enhanced MobileNetV2 SSDLite can also demonstrates that the enhanced strategy can improve the performance significantly with just a little speed drop.

Table 2. Detection results comparison using different frameworks and network architectures. mAP is reported with COCO primary challenge metric (AP at IoU=0.50:0.05:0.95)

method	mAP(infected)	mAP(dead)	mAP(total)	MAdd	Params
SSD 300	0.259	0.545	0.402	30.48G	23.88M
MobilenetV1_SSD 300	0.103	0.482	0.293	1.13G	5.52M
MobileNetV2 SSDLite 300	0.286	0.497	0.392	947.62M	3.91M
E MobileNetV2 SSDLite 300 (ours)	0.318	0.583	0.451	1.24G	5.25M

In Figure 4, they are some detection samples with these models. The first row of the Figure 4 is an image that contains infected plants with some red color. They are very easy to be misclassified into the dead category. The second row is an image that is full of dead plants. Some of the plants are intersected with each other, which increases some difficulties to a model to distinguish one object from another. The last row is an image with some small objects, which is the difficult problem for most of the objects detection methods. It can be seen that the proposed model can perform well on small objects which has low ratio of the bounding box area over the whole image.

MobilenetV1_SSD MobileNetV2 SSDLite SSD Ours GroundTruth

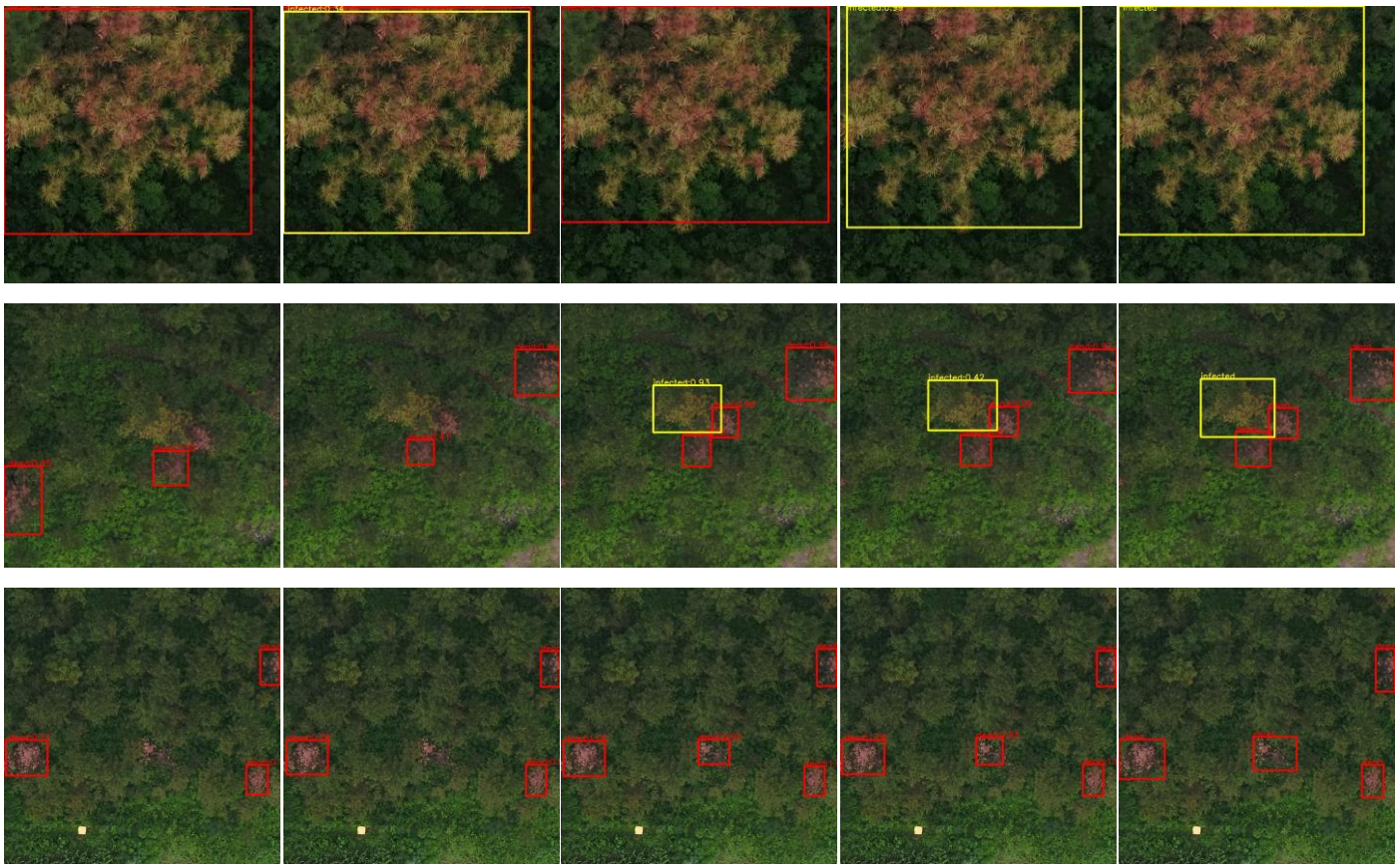


Figure 4. Sample of different models. The left column shows the result of MobilenetV1_SSD model, the second column shows the result of MobileNetV2 SSDLite, the third column shows the detection result of SSD model, the fourth column is the result of our model and the last column is the ground truth

Figure 5 shows performance for different models on different sizes of objects. Three sub datasets are selected from the original test dataset to do further test. They are named as big, medium and small dataset classified through objects' scale in images. The big one is images with object scales larger than 200*200 pixels, the small one is images with their object sizes smaller than 100*100 pixels, and the medium one' images have object areas that are between 100*100 pixels and 200*200 pixels. All the models are tested with these three datasets. As can be seen from the figure 5, the proposed model performs relatively well on any size of objects, especially, it can greatly outperforms other models with on small objects which has low ratio of the bounding box area over the whole image.

Figure 6 shows the recall vs. average precision cursive of all the models. In the figure, the proposed model have few superiority when the recall value is small (<0.3). The precision superiority becomes obviously as the recall value become large, when the recall value comes to 0.9, the precision difference of all the algorithms start to reduce. Considering the reality use case of most object detection

algorithms, the model precision should be measured at a high recall value, for example, at the value of 0.7. At this recall value, the proposed model can performs significant well than other models.

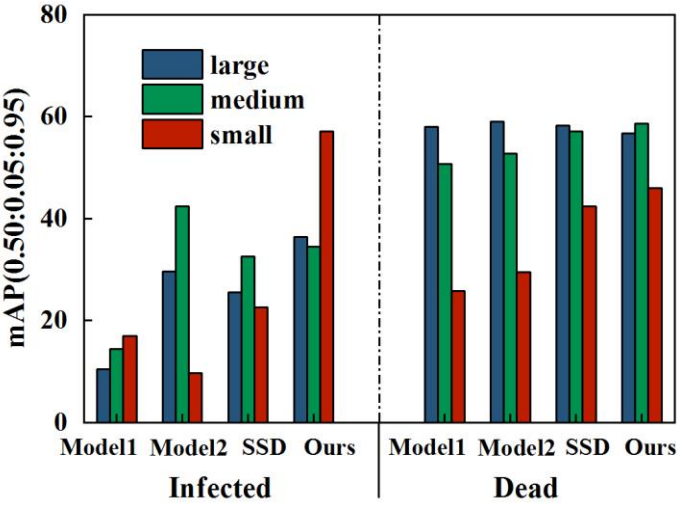


Figure 5. Result of different scales (Model1, Model2, SSD, and Ours indicate MobilenetV1_SSD 300, MobileNetV2 SSDLite 300, SSD 300, and E MobileNetV2 SSDLite 300, respectively).

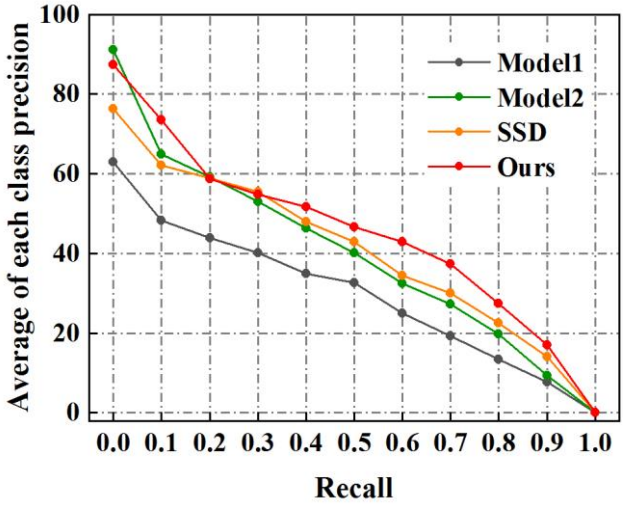


Figure 6. Recall vs. average precision (Model1, Model2, SSD, and Ours indicate MobilenetV1_SSD 300, MobileNetV2 SSDLite 300, SSD 300, and E MobileNetV2 SSDLite 300, respectively).

3.2. Transplant to mobile device

In order to test the efficiency of the model running on a mobile device, the model is transplanted to a mobile phone with CPU of valon 625 (MSM8953) at last. Figure 7 shows some results of the transplanted demo runs on mobile phone, the running time is about 950 milliseconds for a picture with the size of 1000*1000 pixels.

Figure 7. Running results on a mobile phone

4. Conclusion

This study proposes an enhanced MobileNetV2 SSD model to detect infected and dead plants from the images captured by UAV. Firstly, an additional connection from expanded_conv_10 which is at a lower level of VGG network to the location and confidence predictor is added. Then, the predictor combined with layer of expanded_conv_13 is associated with 6 different default boxes rather other 3. Moreover, an atrous convolution block is newly designed to create more feature map for object bounding boxes and confidence detectors. From the results of test, all these improvements show quite well performance that outperforms the state-of-art real time detection methods on a number of various images. Especially, the experiments' results show that the new method can perform well on small objects detection, and achieving a high overall accuracy. In addition, this study also gives the best capturing height of UAV in order to get relatively high over overall detecting accuracy. The whole method provides a general way to detect dead and infected plants in a wide variety of UAV images.

Acknowledgments: This research was financially supported by the Beijing's Science and Technology Planning Project (No. Z171100001417005) and the Fundamental Research Funds for the Central Universities (NO. 2015ZCQ-XX).

Author Contributions: Wenping Liu and Dashuang Liang designed the research, Yu Sun, Dashuang Liang and Lei Zhao conducted the training and test. All authors contributed to the write-up of the manuscript.

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

Reference

1. Dash, J.P.; Watt, M.S.; Pearse, G.D.; Heaphy, M.; Dungey, H.S. Assessing very high resolution UAV imagery for monitoring forest health during a simulated disease outbreak. *Int. J. Photogramm. Remote Sens*. **2017.** 131, 1-14.

2. Chiu, S.F. Investigations on botanical insecticides in south China - an update. Botanical Pesticides in Integrated Pest Management, Rajahmundry, India, **1993.** pp. 134-137.

3. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* **2017.** 17(9), 2022.

4. Cao, L. Research Department. The Research Progress on Machine Recognition of Plant Diseases and Insect Pests. *Chinese Agricultural Science Bulletin* **2015.** 31(20), 244-249. (Abstract in English)

5. Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet classification with deep convolutional neural networks. International Conference on Neural Information Processing Systems. Curran Associates Inc. **2012.** pp. 1097-1105.

6. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. arXiv: 1409.4842v1. **2014.**

7. He K.; Zhang X.; Ren S.; Sun, J. Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision & Pattern Recognition, **2015.** pp. 770-778.

8. Turk, M.A.; Pentland, A.P. Face recognition using eigenfaces. Computer Vision and Pattern Recognition, 1991. Proceedings CVPR 91. IEEE Computer Society Conference on. IEEE Xplore. **2014.** pp. 586-591.

9. Liu, L.; Tran, T.D.; Sang, P.C.; Partial face recognition: A sparse representation-based approach. IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE. **2016.** pp. 2389-2393.

10. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE T. Pattern Anal.* **2014.** 39(4), 640-651.

11. Chen, L.C.; Papandreou, G..; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE T. Pattern Anal.* **2016,** 40(4), 834-848.

12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. *IEEE T. Pattern Anal.* **2015.** 39(6), 1137-1149.

13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. arXiv:

1506.02640v5. **2015.**
15. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, Deva.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. arXiv: 1405.0312. **2014.** 1, 7.
16. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A.; Li, F. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vision* **2015.** 115(3), 211-252.
17. Zhang, X,; Zhou X.; Lin M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. arXiv: 1707.01083. **2017.**
18. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv: 1704.04861v1. **2017.**
19. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. arXiv: 1801.04381v3. **2018.**
20. Liu, B.; Zhang, Y.; He, D.J.; Li, Y. Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry* **2017.** 10(1), 11.
21. Näsi, R.; Honkavaara, E.; Lyytikäinensaarenmaa, P.; Blomqvist, M.; Litkey, P.; Hakala, T.; Viljanen, N.; Kantola, T.; Tanhuanpää, T.; Holopainen, M. Using UAV-based photogrammetry and hyperspectral imaging for mapping bark beetle damage at tree-level[J]. *Remote Sens.* **2015.** 7(11), 15467-15493.
22. Lehmann, J.R.K.; Nieberding, F.; Prinz, T.; Knoth, C. Analysis of unmanned aerial system-based CIR images in forestry—a new perspective to monitor pest infestation levels. *Forests* **2015.** 6(3), 594-612.
23. Yuan, Y.; Hu, X. Random Forest and Objected-Based Classification for Forest Pest Extraction from Uav Aerial Imagery. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLI-B1, **2016.** 1093-1098.
24. Meddens, A.J.H.; Hicke, J.A.; Vierling, L.A.; Hudak, A.T. Evaluating methods to detect bark beetle-caused tree mortality using single-date and multi-date Landsat imagery. *Remote Sens. Environ.* **2013.** 132(10), 49-58.
25. Michez, A.; Piégay, H.; Lisein, J.; Claessens, H.; Lejeune, P. Classification of riparian forest species and health condition using multi-temporal and hyperspatial imagery from unmanned aerial system. *Environ. Monit. Assess.* **2016.** 188(3), 146.
26. Huang, J.; Guadarrama, S.; Murphy, K.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z. Speed/accuracy trade-offs for modern convolutional object detectors. arXiv: 1611.10012v3. **2016.**
27. Chen, L.C.; Papandreou, G..; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv: 1412.7062. **2015.**
28. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R..; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. arXiv:1408.5093v1. **2014.«地址块»**