IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 38, No. 1, February, 2021, pp. 61-71
Journal homepage: http://iieta.org/journals/ts

Detection of Dense Small Rigid Targets Based on Convolutional Neural Network and Synthetic Images

Xianrong Zhang[1,2], Gang Chen[1*]

[1] Zhejiang University, Hangzhou 310058, China
[2] Zhijiang College of Zhejiang University of Technology, Shaoxing 312030, China

Corresponding Author Email: cg@zju.edu.cn

https://doi.org/10.18280/ts.380106

Received: 9 October 2020
Accepted: 28 December 2020

Keywords:
target recognition, artificial data, rice planthoppers, deep learning (DL), convolutional neural network (CNN), faster region-based CNN (Faster-RCNN)

ABSTRACT

Facing the image detection of dense small rigid targets, the main bottleneck of convolutional neural network (CNN)-based algorithms is the lack of massive correctly labeled training images. To make up for the lack, this paper proposes an automatic end-to-end synthesis algorithm to generate a huge amount of labeled training samples. The synthetic image set was adopted to train the network progressively and iteratively, realizing the detection of dense small rigid targets based on the CNN and synthetic images. Specifically, the standard images of the target classes and the typical background mages were imported, and the color, brightness, position, orientation, and perspective of real images were simulated by image processing algorithm, creating a sufficiently large initial training set with correctly labeled images. Then, the network was preliminarily trained on this set. After that, a few real images were compiled into the test set. Taking the missed and incorrectly detected target images as inputs, the initial training set was progressively expanded, and then used to iteratively train the network. The results show that our method can automatically generate a training set that fully substitutes manually labeled dataset for network training, eliminating the dependence on massive manually labeled images. The research opens a new way to implement the tasks similar to the detection of dense small rigid targets, and provides a good reference for solving similar problems through deep learning (DL).

1. INTRODUCTION

With the explosive development of deep learning (DL) [1], algorithms based on convolutional neural networks (CNNs) have achieved great success in target detection [2-6]. This type of algorithms has been effectively applied to solve problems in various industries [7-10]. Compared with traditional target detection algorithms, CNN-based algorithms support data-driven feature extraction with the aid of artificial neural networks (ANNs), acquiring deep abstract features of a specific dataset after learning numerous samples. These abstract features are more robust and generalizable than manually extracted features.

However, the successful application of DL methods hinges on a large amount of training data. As a result, the network training faces a huge obstacle: the lack of sufficient labeled data as training samples. The lack is particularly severe, if the detection targets belong to a small yet highly professional field. It is a very costly task to label the samples from such a field, requiring lots of manpower and time. Besides, the labeling personnel must have strong professional background knowledge. The labeling task is especially costly, when the targets to be labeled belong to different types, the difference in visual features between the types is small, the target size is limited, the target density is high, and the images are from multiple sources.

To overcome the lack of massive labeled data for DL methods, scholars have conducted lots of research, and proposed many different methods. For example, transfer learning [11], small sample learning [12], unsupervised and weakly supervised learning [13] have been adopted to lower the dependence on a large amount of labeled data. But none of these methods can achieve a comparable performance as supervised learning.

Some scholars tried to enhance and expand data by means of image processing algorithms [14-16]. The core idea is to regularize the target images in the original training set to generate new target images, without changing their labels. This approach can effectively expand the size of the training set. But the expansion effect depends on the completeness of the original dataset. The types of samples not present in the original dataset cannot be generated through this approach.

Some scholars trained the network with artificial synthetic data, and combined the synthetic images with real images into a complete training set [17, 18]. In this way, new types of target images can be generated. However, their research is limited to specific work scenarios. In most cases, the data are synthetized manually, for the lack of an automatic end-to-end synthesis algorithms. Moreover, their strategy is not universally applicable, due to the limitation of work scenario.

Therefore, this paper proposes an automatic end-to-end synthesis algorithm to generate a huge amount of labeled training samples. In the absence of labeled training images, the proposed method can synthetize a training set containing all target classes based on the standard images in the target classes, and automatically label each training image. Using the generated synthetic training set, the network can be trained progressively and iteratively. Apart from reducing the

workload of manual labeling, this training process helps to adapt CNN-based target detection algorithms to the identification of multi-class dense small rigid targets.

2. LITERATURE REVIEW

In DL-based target detection, the network needs to be trained by lots of labeled training samples. When this technology is applied to solve industrial problems, it is a very difficult task to acquire numerous real images and label them correctly. The lack of training data poses a major obstacle to the successful application of this technology. Many scholars have resorted to various methods to avoid this obstacle.

One of these methods is to fully utilize the information carried in existing training samples, using techniques like transfer learning and data enhancement. The traditional data enhancement strategies include image processing operations such as transform, rotation, flipping, and scaling. Through these operations, new samples can be generated from the existing samples in the original dataset, which to a certain extent increases the number of available training images. However, if the original dataset lacks some types of samples, it is impossible to generate these types of samples through data enhancement.

Cubuk et al. [19] designed an efficient data enhancement strategy that automatically looks for the optimal combinations of the dataset, using a search algorithm. But the performance improvement of the strategy cannot exceed the improvement range dependent on the training capacity provided by the data enhancement algorithm. Lim et al. [20] improved Cubuk's strategy to achieve the same performance at a faster search speed. Buslaev et al. [21] realized a fast, flexible data enhancement function library, which provides most of the common image data enhancement functions, but does not contain any novel method.

In many studies, data enhancement is adopted to expand the number of samples for target detection in a particular domain. For example, Wang et al. [22] employed several data enhancement algorithms suitable for target detection of synthetic aperture radar (SAR) images, and effectively expanded the number of training samples. Wu et al. [23] discussed the role of common data enhancement techniques for expanding sample set in broader domains. In general, data enhancement alone mainly mines deep visual features from the existing samples. The mining effect is not ideal, if the current dataset is too sparse or lack some types of samples.

Due to the difficulty in acquiring lots of correctly labeled real images, the synthetic image technology has been introduced to increase the sample size, such as to train the network more adequately. So far, researchers have attempted to generate synthetic images by various methods. For instance, Narayanan et al. [24] synthetized aerial images with a game engine; this approach is not universal, for its applicable scope depends on the functions of the game engine. Rajpura et al. [25] used a three-dimensional (3D) engine to synthesize images of the items in the refrigerator. With the help of a 3D engine, Xu et al. [26] synthesized multiple mutually occluded pedestrian images in surveillance video. Both Rajpura and Xu Jian relied on 3D engine to identify spatial relationship between different targets in the synthesis images. Thus, their approaches are both limited by the specific scene of the task and the functions of the 3D engine. Lu et al. [27] synthetized a moving target detection dataset through affine transform, trained the deep CNN on the dataset, and verified the trained network on a test set of real images; but this strategy can only deal with moving targets, rather than general targets.

In addition, Jiang et al. [28] synthetized a dataset for logo detection, and proved the effectiveness of the method; however, the method only applies to target detection problems, in which the logos have fixed graphical features. Jin et al. [29] separated the vehicle foreground from real images, and embedded it in various complex scenes, producing multiple training images; with vehicle as the detection target, this method can effectively pinpoint targets of the same class amidst abundant samples, but cannot detect various types of targets with a high inter-class similarity out of many samples. Through image processing, Xu et al. [30] synthesized the smoke shrouding effect, and successfully applied it to video smoke detection; nevertheless, this approach is not suitable to general problems of target detection, because the targets must overlap with the background, rather than cover the background. Similarly, O'Byrne et al. [31] explored video smoke detection of underwater scene images with artificial synthetic data.

Most of the above methods are not applicable to similar problems in other industries. Some of them are constrained by image synthesis methods and tools, and some are limited to special application scenarios.

Some scholars utilized more complex algorithms to synthesize training data. Frid-Adar et al. [32] generated training data by generative adversarial network (GAN). Wang et al. [33] adopted Wasserstein GAN + gradient penalty (WGAN-GP) to expand the number of cooperative target images, and applied the expanded image set to the detection network based on you look only once (YOLO). Kim and Myung [34] cascaded autoencoders into a GAN, and synthetized images with the network for the target detection of jellyfish swarm. Nonetheless, these methods are unlikely to be transplanted to other fields, owing to the complex mechanism, complicated network structure, and high overload of network training.

In all the studies above, the generation methods for training set face two limitations. Some methods have a high complexity and a huge training overhead. Some are limited to specific scenarios, and not applicable to similar problems. To recognize dense small rigid targets from various sources, this paper proposes an end-to-end automatic synthesis algorithm for training images. The synthetized dataset was combined with a few real images to train the network progressively and iteratively. The proposed algorithm was proved effective through an example analysis on target detection problem of rice planthoppers.

The rest of this paper is organized as follows: Section 3 describes the methodology of this research, provides the end-to-end automatic synthesis algorithm for training images, explains the selection of network training parameters, and introduces the features of sample classification in the target detection problem of rice planthoppers; Section 4 carries out an experiment, analyzes the experimental results, and verifies the effectiveness of our algorithm; Section 5 summarizes the findings of this research.

3. METHODOLOGY

3.1 Samples and evaluation metrics

In the target detection problem of rice planthoppers, there

are three kinds of targets: brown planthoppers, gray planthoppers, and white-backed planthoppers. Each kind of planthoppers can be divided by shade into a dark type and a light type. For every kind of planthoppers, each insect needs to through five nymphic stages before becoming an imago. Every imago is either male or female, and long-winged or short-winged. To sum up, the rice planthoppers in this research fall into 54 subcategories. As shown in Table 1, the standard images of 47 subcategories were obtained for our experiment. The training images were synthetized from the 47 different kinds of standard images.

Table 1. Classes of rice planthoppers

	D/1	D/2	D/3	D/4	D/5	L/1	L/2	L/3	L/4	L/5	D/L/F	L/L/F	D/S/F	L/S/F	D/L/M	L/L/M	D/S/M	L/S/M
B																		
G														X			X	X
W						X	X										X	X

Note: B, G, and W refer to brown, gray, and white-backed rice planthoppers, respectively; D/L is dark type or light type; 1-5 stand for the five nymphic stages; F/M is female or male imago; L/S is long-winged or short-winged.

Careful observation shows that some classes of standard images have very similar appearances, with very minor differences. Therefore, the target detection problem of rice planthoppers aims to classify targets with high inter-class similarity. Since there are so many targets of similar classes in the original images, it is a challenging task to recognize and count the targets in each class, even the recognition and counting are performed by experts with rich experience in the domain.

Under the application scenario of our problem, estimating the total number of rice planthoppers in the target images is more important than the accurate classification of an individual rice planthopper. Hence, the effectiveness of target detection method should be evaluated by the ability to detect the rice planthoppers in the images, rather than the correct classification of the detected targets.

Next is a brief introduction to the evaluation metrics of experimental results. Precision and Recall are two common metrics of the performance of target detectors:

$$P = TP / (TP + FP) \tag{1}$$

$$R = TP / (TP + FN) \tag{2}$$

where, TP is the number of correctly categorized positive samples; (TP+FP) is the total number of samples categorized as positive; (TP+FN) is the total number of actual positive samples.

Many other metrics, including F1 score, mean average precision (mAP), receiver operator characteristic (ROC) curve, and area under curve (AUC), are developed from precision and recall. Because our aim is to evaluate the effectiveness of the proposed algorithm, precision and recall were selected as the basic metrics for quantitative analysis.

In the application scenario of our problem, the images in some classes bear high resemblance in appearance, and the key task is to identify every rice planthopper. Therefore, the targets that are correctly recognized and located are still meaningful, even if they are categorized into wrong classes.

3.2 End-to-end automatic image synthesis algorithm

In the target detection problem for rigid targets, there are graphical differences between the targets in real images, which arise from the variation in shooting environment, shooting devices, light conditions, positions, and angles. Despite these differences, the rigidity of the targets ensures the stability of the visual features that differentiate different types of targets. This is the root reason for the effective simulation of real images by image processing algorithms.

Take the target detection of rice planthoppers as an example. The target images could come from various sources, such as professional lab devices, field trap and shooting devices, drone shooting devices, and the Internet (Figure 1). If the real images are directly taken as training samples, it is very laborious for professional personnel to label all the images for network training. In fact, the labeling task is a daunting task, for the sheer number of targets, graphical differences of targets in the same class, and the numerous connection weights of the network.

Figure 1. Multi-source target images

In this paper, manual labeling is replaced with an end-to-end automatic image synthesis algorithm to generate the training set. As shown in Figure 2, the algorithm produces the synthetic dataset in three steps: First, collect the standard images on each type of targets; Second, perform data enhancement on the standard images to simulate every possible appearance of the insect on real images; Third, superimpose the processed insect images as foreground onto various backgrounds, i.e., the images of different typical scenes, such as close-up images on field crops, real images shot in the field, and monochrome images.

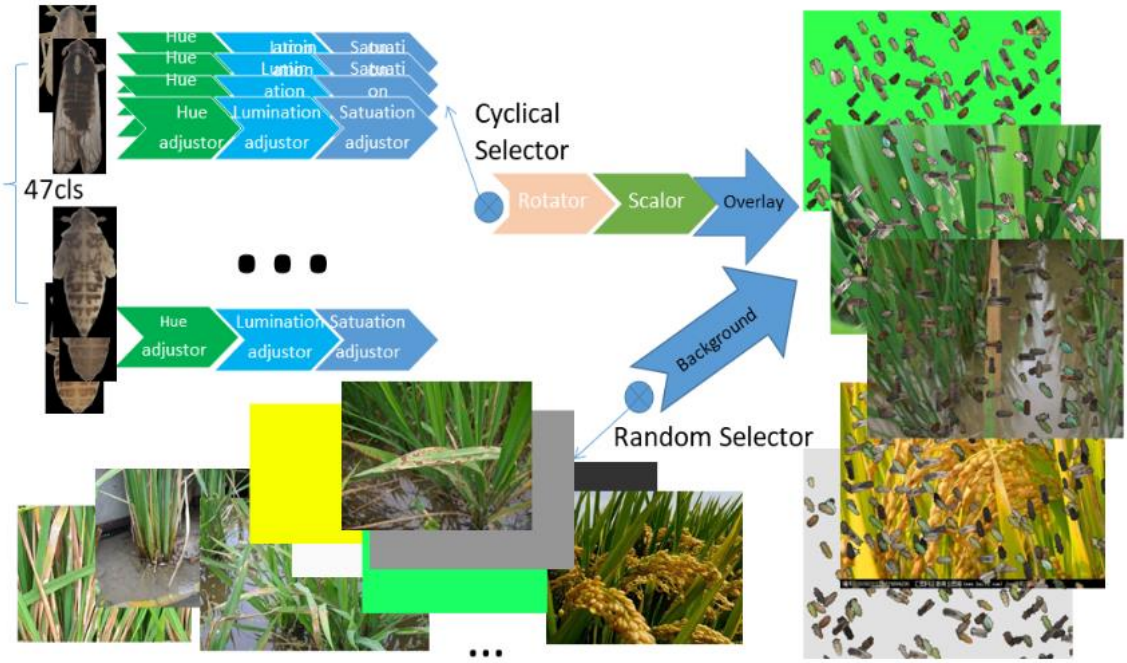


Figure 2. End-to-end automatic image synthesis algorithm

The real images, collected from the field, contain various changes induced by the number, pose, and direction of insects, as well as the shooting environment, device location, and device orientation. These changes should be retained as much as possible in the synthetic image set, such that the synthetic dataset has similar feature distribution as the real images. For this purpose, the standard images on rice planthoppers were preprocessed by image processing algorithm to mimic the changes in the real environment. The preprocessing procedures is detailed below.

(1) Rotation

The various orientations of targets in the labeled images were simulated by rotation:

$$A_1 = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, 0 \le \theta \le 2\pi \tag{3}$$

where, θ is the rotation angle. In this paper, θ is changed at the step size of 15° in the interval of [0°, 345°].

(2) Scaling

The size variation of targets induced by different sources of labeled images and the distances to the camera was simulated by scaling:

$$A_2 = \begin{pmatrix} S & 0 \\ 0 & S \end{pmatrix}, s \ge 0 \tag{4}$$

where, S is scaling ratio. Three different scaling sizes were used in the experiment, creating an image pyramid for each target. The network trained by the image pyramids can correctly detect targets of different sizes. Note that some information of the original image might get lost through rotation and scaling. To prevent the error accumulated by step-by-step reduction, the images of different rotation angles and scaling ratios were all directly transformed from the high-resolution original images. This strategy has a much smaller information loss than the existing rotation and scaling methods based on small targets in real images.

(3) Hue/lightness/saturation (H/L/S) adjustment

In the labeled images, the targets have L and S changes caused by lighting and device conditions. In our experiment, the three attributes of the color space, namely, H, L, and S, were changed by the step size of 1/10 in the change interval.

Figure 2 illustrates the generation of training images. For the standard images on 47 types of rice planthoppers, an image processor was generated for each image, and used to adjust H, L, and S by the said step size in the change interval. After each adjustment, one of the 47 image processors was chosen as the foreground, and subject to rotation and scaling, before being superimposed on a randomly selected background. Meanwhile, the coordinates of the superimposed position were written into the Extensible Markup Language (XML) file as the label data. This process was repeated until the number of foregrounds superimposed on the background reached the preset number. Then, a training image and its label file were generated. The above steps were iteratively implemented till all the generated images were superimposed. During each superimposition, the image processor was selected cyclically, such that the target images of all types were superimposed on the training images in turn. This method balances the distribution of different classes of samples, avoiding the imbalance of sample distribution.

Following the above synthesis method, the total number of generated target samples can be calculated by:

$$24 \times 3 \times 10 \times 10 \times 10 \times 47 = 3,384,000$$

Suppose 100 target images are superimposed on each training image. Then, a total of 3,384,000 training images could be obtained. It would be immensely difficult to manually collect so many labeled training images. The difficulty lies in the acquisition of enough images, and the correct labeling of targets in numerous images. By contrast, our synthesis algorithm automates the generation of training images and the accurate labeling of targets.

3.3 Network structure and training strategy

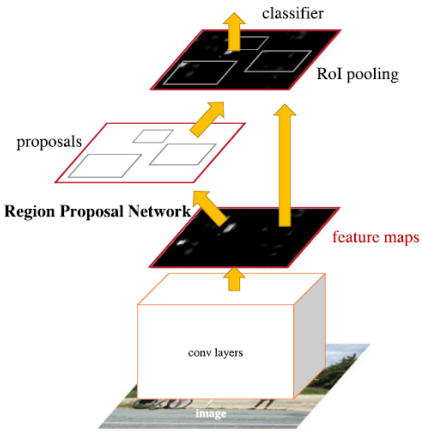
Considering the application scenario, the target samples of this research have the following features:

1. The main goal of the task is to detect rice planthoppers in multi-source images. From different sources, the images differ in resolution and background.

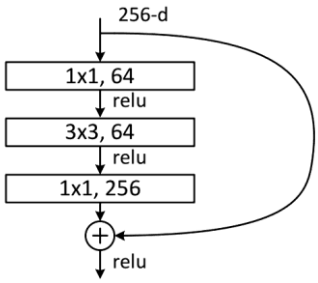
2. The targets in the images are very small, usually as large as a dozen of pixels. The effective detection of small targets is critical to this research. The detection model should be able to recognize targets of various sizes.

3. The application scenario, as a mode of sampling analysis, has a low requirement for real-timeliness. Compared with detection accuracy, model speed is not a key element.

Through the above analysis, faster region-based CNN (Faster-RCNN), with residual network 101 (ResNet101) as the backbone, was selected as the detection model. The main structure of the FASTER R-CNN is shown in Figure 3(a). The convolutional layers adopt the ResNet101 structure, which includes a 99-layer CNN stacked from the modules in Figure 3(b), an input convolutional layer, and a fully-connected layer.



(a) Structure of Faster-RCNN



(b) Bottleneck module of ResNet101

Figure 3. Structure of Faster-RCNN with ResNet101 as the backbone

In actual network training, the initialization of network weights directly impacts the convergence speed of the network. The common ways to initialize weights include full-zero initialization, random initialization, Xavier initialization, and He initialization.

Transfer learning of suitable pre-training network parameters is an effective way to reduce the difficulty of network learning [11]. The main idea is to avoid re-training the entire network with limited training data in the target domain, starting from the lowest layer. In the field of image processing and computer vision, transfer learning often assumes that low-level image features, such as edge and simple geometry, are independent of the actual image contents in the target domain. Therefore, these underlying features can be learned by any dataset containing lots of available training data. The pre-trained model can be used as training benchmark, and further finetuned to adapt to the target domain. In this way, the training objective can be achieved with fewer data than direct training from the initial state. In our experiment, transfer learning was adopted to prevent training the network from random weights.

3.4 Experimental procedure

The procedure of our experiment was designed as follows: First, take the weights of the pre-trained network as the initial weights; train the network with the initial training set generated by the synthesis algorithm mentioned in 3.2, and verify the trained network with a few real images; treat the targets not detected or incorrectly detected in the test set as the targets for the next cycle, convert them into new training data, and superimpose the data on the original training set for the iterative training of the network; repeat this step until the network reaches the preset detection standard. Overall, image synthesis is implemented iteratively throughout network training, rather than only once before the first training. The overall procedure of our experiment is detailed below.

Step 1. Generate training images and label files by the method described in Section 3, and create the dataset required for Faster R-CNN training.

Step 2. Initialize the weights of the network as the pre-trained network weights.

Step 3. Start training with the dataset generated in Step 1.

Step 4. Terminate the training when the total loss reaches the preset level or the number of training steps surpasses the preset value.

Step 5. Apply the trained network to detect the labeled images, compare the detection results with the real values, and jump to Step 9 if the difference is smaller than the preset value.

Step 6. Separate the targets not detected or incorrectly detected in Step 5, treat the target images as foregrounds, and synthetize them into new training images by the method specified in Section 3.

Step 7. Progressively train the network obtained in Step 5 with the training images generated in Step 6.

Step 8. Return to Step 5.

Step 9. Perform detection with new labeled images, repeat Steps 5-8 until the test images are used up, and terminate the training.

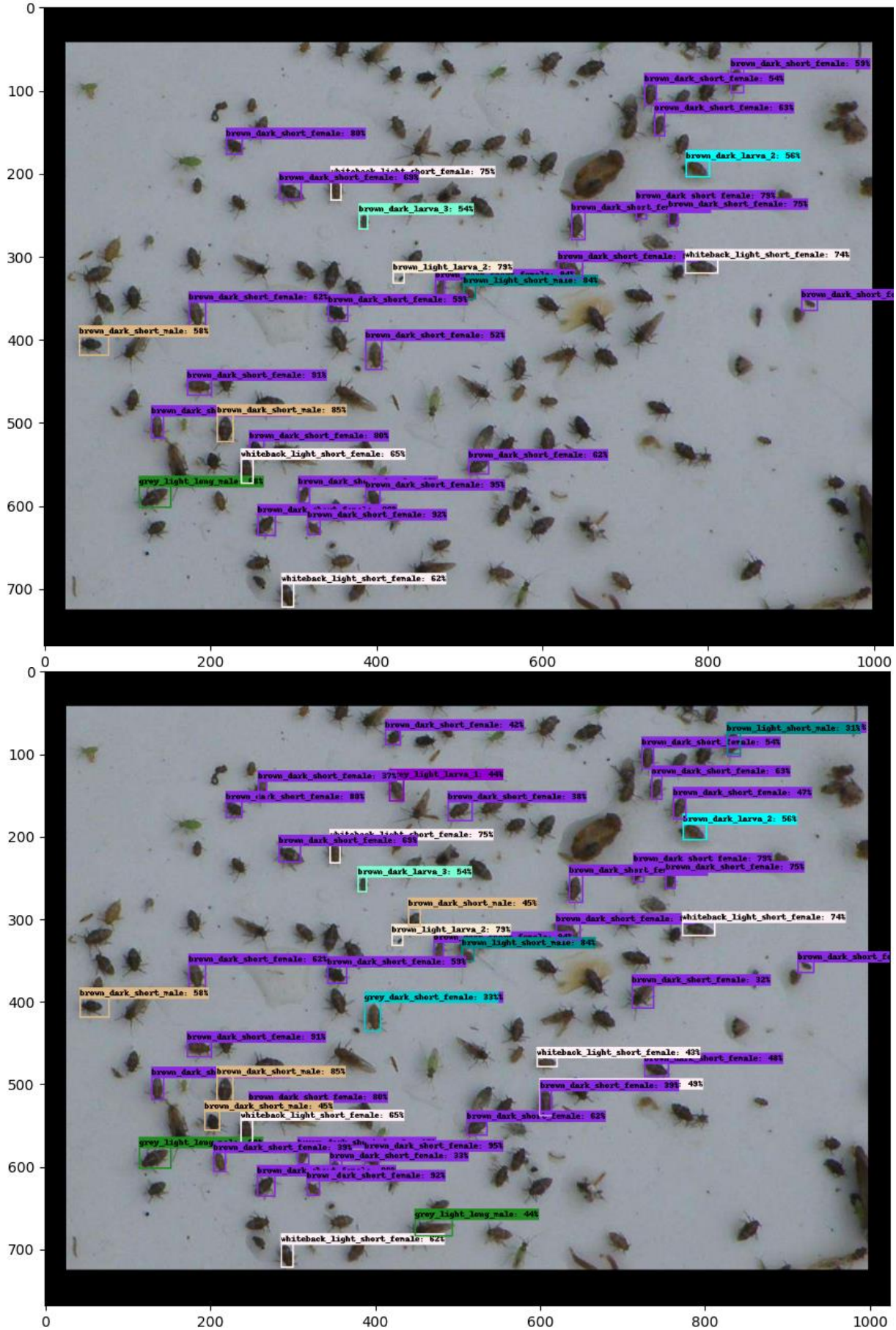
Step 10. Obtain the final results of network training.

4. RESULT ANALYSIS AND DISCUSSION

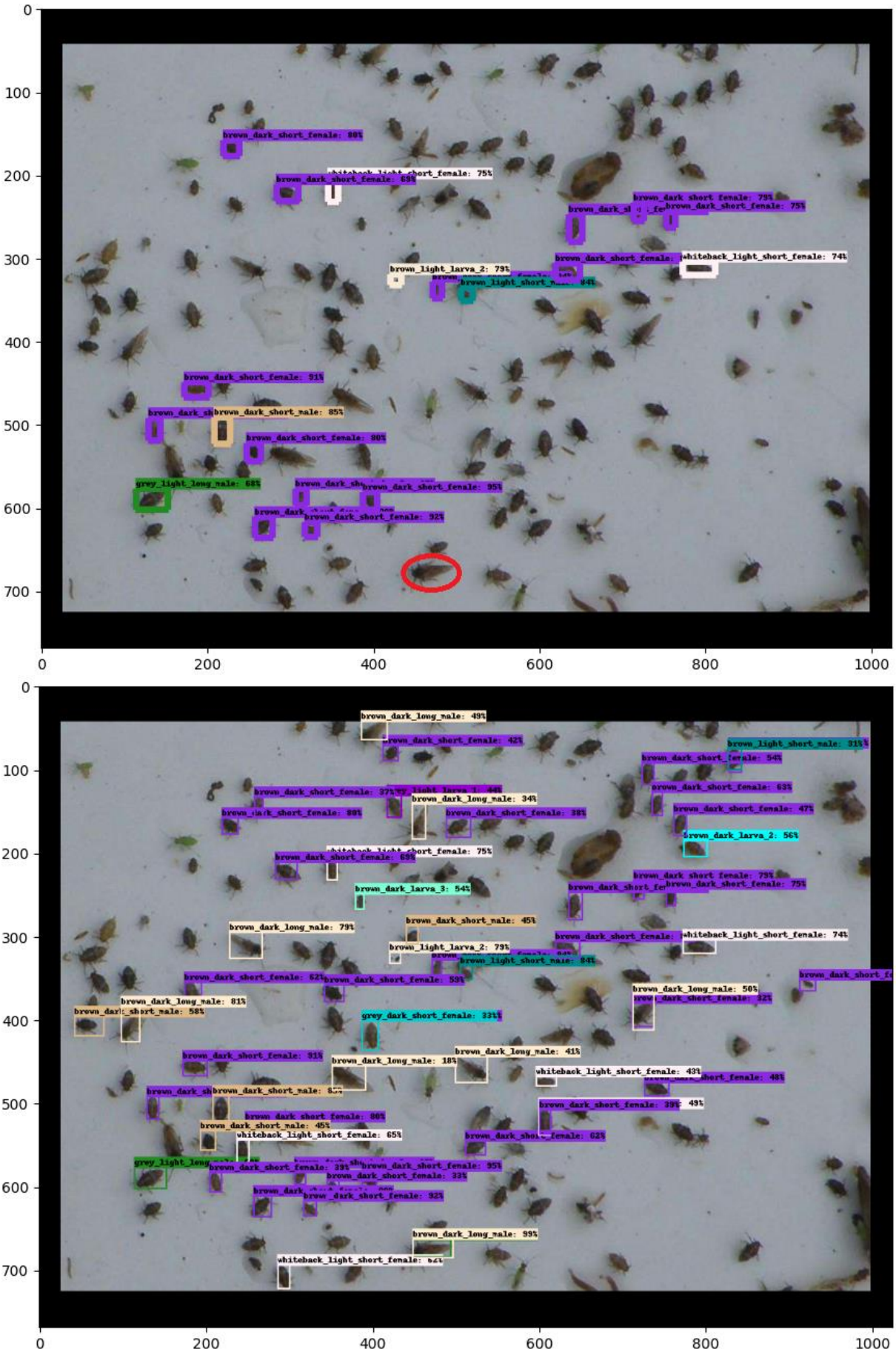
Our experiment attempts to verify the feasibility and effectiveness of synthetic training images for neural network training.

Figure 4 presents the real image detection results of our network trained by synthetic image data. The experimental results show that, after being trained by the synthetic image set, the network could correctly detect part of the targets, a sing of the feasibility of our strategy. However, quite a few targets were not detected, due to the limited number of standard images adopted in image synthesis. In our experiment, the training images on the same type of objects were derived from the same original standard image. Although the sample size

was greatly expanded through image processing, it is not sufficient to cover all the actual changes of different types of insects with similar features in various real images (Figure 4a).



(a) Target detection results on real images after initial training with synthetic training set (confidence threshold: 0.9, 0.7, 0.5, and 0.1)



(b) Target detection results after iterative training with additional training samples synthetized from the targets in red ellipses

Figure 4. Target detection results on real images after initial training with synthetic training set and after one iterative training with additional new synthetic training samples

To make up for the defect, the targets not detected were used to synthetize new training images. The new synthetic images were supplemented to the training set for progressive training of the network. Experimental results show that this strategy could quickly improve the recall in the target detection of real images from the same source.

During the experiment, a single target was selected in turn, and taken as the standard image. The training samples generated from the standard image were applied to iteratively train the network. The results suggest that this approach effectively enhanced the recall of samples in the same class. After several iterative trainings, the network achieved satisfactory effect on the test set. The effects of multiple iterative trainings are given in Table 2.

To clearly present the effect of iterative training, only a single target sample was selected for image synthesis in each iteration. Figure 4b compares the detection effects before and after one iterative training. The results show that the proposed method can generate a training set from the few samples, using the end-to-end automatic synthesis algorithm, and the training set can replace the set of real images with lots of manual labels in the initial training of the network. Throughout the network training, our method could realize satisfactory detection results, with a small amount of manual labeling in the testing phase.

Table 2. Detection results after initial training and iterative training

Number of iterations	Number of actual positives	TP	Number of missed positives	Miss rate	FP	Recall (%)
0	120	44	76	63.33	1	36.67
1	120	55	65	54.16	1	45.83
2	120	67	53	44.17	3	55.83
3	120	81	39	32.50	5	67.50
4	120	90	30	25.00	4	75.00
5	120	97	23	19.17	6	80.83
6	120	104	16	13.33	8	86.67
7	120	109	11	9.17	15	90.83

Note: FP means the number of negatives incorrectly identified as positives; the samples were classified by two criteria: if the confidence of a sample is greater than zero in only one class, then the sample will be assigned to that class; if the confidence of several samples is greater than zero in one class, the sample with the highest confidence will be assigned to that class.

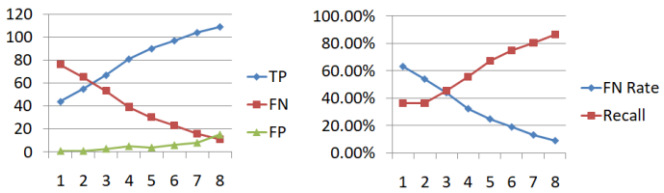


Figure 5. Trends of TP, FN, FP, FN rate, and recall after multiple iterative trainings

As shown in Figure 5, the FN rate of the network exhibited a rising trend after multiple iterative trainings. This might be attributed to the following reason: After more diverse samples are added to the training set, the distance between the depth features in adjacent classes will be close to each other in the depth feature space, due to the similarity between different types of image features in the application scenario of our problem. With the enhancement of detection ability, the network is more likely to detect the targets on the edge of sample clusters, pushing up the probability of mistaking a target of a class for one of an adjacent class.

The proposed method has the following advantages over the training with manually labeled sample set:

(1) The premise of manual labeling is to obtain sufficient real images, especially if the network is large and the targets are diverse. In industrial practice, network training often requires thousands to tens of thousands of images. It is a difficult task to gather so many training images. In the professional field of our problem, the image collection is even more unlikely to complete, because the training set must contain dozens of sample images. Meanwhile, the success of the DL hinges on the sufficiency of training images. Thus, it is impossible to complete the network training for our task, solely based on manually labeled training samples. In other words, the DL is not very applicable to tasks similar to ours. In contrast, our end-to-end automatic synthesis algorithm can generate the training set with minimal manual intervention.

(2) Compared with manually labeled images, synthetic images as training samples contain accurately classified and positioned targets. During manual labeling, the operator must be very careful to mark each target with a box in every real image and specify the class of that target. However, errors are commonplace in manual labeling, especially if the original images are fuzzy for reasons of poor lighting, camera vibration, low resolution, or ultra-small targets. If any error occurs, the data will be incorrect, and cannot be removed through data cleaning. In our task, the targets are divided into very refined classes, with a high inter-class similarity. In this case, wrong labeling is very likely to occur. Any misclassification or inaccurate positioning of targets by the operator will introduce wrong data or noises to the training set, which cannot be easily removed.

(3) It is hard to balance the targets of different types in the manually labeled image data. This is particularly true, when the targets to be recognized and located belong to various types. In our problem, a total of 47 types of rice planthoppers need to be detected. However, some types of insects seldom appear in the real images captured in the field. In many cases, most real images contain only the insects of the same type, while some types of insects are not contained in any image. If the initial training set is synthetized by our method, the data will be easy to process, such that different types of insects appear at more balanced frequencies.

(4) If the dataset is manually labeled based on the real images, it is difficult to separate the targets from backgrounds through preprocessing, not to mention facilitating target feature extraction. If the dataset is synthetized by our method, it would be convenient to process the targets and backgrounds differently and superimpose the processed parts together, making it easier to train the network.

5. CONCLUSIONS

To adapt the CNN to the image detection of dense small rigid targets, this paper proposes an end-to-end automatic image synthesis algorithm, which reduces the workload and technical difficulty of training set generation by manual labeling. Taking a single standard image as the original sample, the proposed algorithm was adopted to create a training set for initial network training. Then, the samples not detected in network testing were iteratively compiled into a new training set for progressive training of the network. The target detection network was trained and tested based on the DL. The main conclusions are as follows:

(1) The initial training of deep network can be effectively implemented by our strategy, i.e., the end-to-end automatic synthesis of training set by image processing, based on a single standard image and multi-source background images.

(2) Taking a few real images as the test set, it is possible to quickly find the deviation of the training set from the actual values. Then, the training set could be supplemented with the minimal manual intervention.

(3) The proposed method can complete network training for image detection of dense small rigid targets, in the absence of lots of labeled real images.

The proposed method provides a reference for solving the difficulty in labeling for image detection of dense small rigid targets. The following aspects of our method need further research: Currently, the target images must be manually separated to generate additional training samples, after the sample targets are selected in the testing and iterative training phase. To overcome the complexity of manual operation, the future research could try to replace the manual separation with automatic or interactive operation based on image segmentation techniques. In addition, the network training results hinge on the quality of the standard images for targets of each type in the initial samples. This is particularly the case, when the targets belong to various classes with high inter-class similarity. If the standard images are of poor quality, it is easy to misjudge the classes of the targets. To realize ideal results, the network must be trained iteratively for many rounds. However, the more the iterations, the higher the FN rate of the network tested on real images. The internal mechanism and solution of this problem should be further studied.

ACKNOWLEDGMENT

This work is supported by the Fund of Zhejiang Provincial Department of Education (Grant No.: Y201432150).

REFERENCES

[1] Hinton, G.E., Osindero, S., Teh, Y.W. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18(7): 1527-1554. https://doi.org/10.1162/neco.2006.18.7.1527

[2] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6): 84-90. https://doi.org/10.1145/3065386

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

[4] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. (2016). Ssd: Single shot multibox detector. In European Conference on Computer Vision, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[5] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788.

[6] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778.

[7] Torney, C.J., Dobson, A.P., Borner, F., Lloyd-Jones, D.J., Moyer, D., Maliti, H.T., Hopcraft, J.G.C. (2016). Assessing rotation-invariant feature classification for automated wildebeest population counts. Plos One, 11(5): e0156342. https://doi.org/10.1371/journal.pone.0156342

[8] Idé, T., Katsuki, T., Morimura, T., Morris, R. (2016). City-wide traffic flow estimation from a limited number of low-quality cameras. IEEE Transactions on Intelligent Transportation Systems, 18(4): 950-959. https://doi.org/10.1109/TITS.2016.2597160

[9] Buzin, A.R., Macedo, N.D., De Araujo, I.B.B.A., Nogueira, B.V., de Andrade, T.U., Endringer, D.C., Lenz, D. (2017). Automatic detection of hypoxia in renal tissue stained with HIF-1alpha. Journal of Immunological Methods, 444: 47-50. https://doi.org/10.1016/j.jim.2017.02.005

[10] Rey, N., Volpi, M., Joost, S., Tuia, D. (2017). Detecting animals in African Savanna with UAVs and the crowds. Remote Sensing of Environment, 200: 341-351. https://doi.org/10.1016/j.rse.2017.08.026

[11] Pan, S.J., Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10): 1345-1359. 10.1109/TKDE.2009.191

[12] Snell, J., Swersky, K., Zemel, R.S. (2017). Prototypical networks for few-shot learning. in Advances in Neural Information Processing Systems (NIPS). arXiv preprint arXiv:1703.05175.

[13] Tian, G.L., Zhou, S., Sun, G.X. (2020). A novel intelligent recommendation algorithm based on mass diffusion. Discrete Dynamics in Nature and Society, 2020: 1-9. https://doi.org/10.1155/2020/4568171

[14] Talukdar, J., Biswas, A., Gupta, S. (2018). Data augmentation on synthetic images for transfer learning using deep CNNs. In 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), 215-219. https://doi.org/10.1109/SPIN.2018.8474209

[15] Shin, H.C., Lee, K.I., Lee, C.E. (2020). Data augmentation method of object detection for deep learning in maritime image. In 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), 463-466. https://doi.org/10.1109/BigComp48618.2020.00-25

[16] Li, H., Rao, J., Zhou, L., Zhang, J. (2019). Valid Data Augmentation by Patch Alpha Matting. In 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP), pp. 361-366. https://doi.org/10.1109/SIPROCESS.2019.8868572

[17] Mahmood, F., Durr, N.J. (2018). Deep learning-based depth estimation from a synthetic endoscopy image training set. In Medical Imaging 2018: Image Processing,

10574: 1057421. https://doi.org/10.1117/12.2293785
[18] Alhaija, H.A., Mustikovela, S.K., Mescheder, L., Geiger, A., Rother, C. (2018). Augmented reality meets computer vision: Efficient data generation for urban driving scenes. International Journal of Computer Vision, 126(9): 961-972. https://doi.org/10.1007/s11263-018-1070-x
[19] Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 113-123.
[20] Lim, S., Kim, I., Kim, T., Kim, C., Kim, S. (2019). Fast autoaugment. arXiv preprint arXiv:1905.00397.
[21] Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A. (2020). Albumentations: fast and flexible image augmentations. Information, 11(2): 125. https://doi.org/10.3390/info11020125
[22] Wang, S.Y., Gao, X., Sun, H., Zheng, X.W., Sun, X. (2017). An aircraft detection method based on convolutional neural networks in high-resolution SAR images. Journal of Radars, 6(2): 195-203. https://doi.org/10.12000/JR17009
[23] Wu, T.Y., Xu, Y.C., Zhao, P.F. (2020). Dataaugmentation technology for improving target image recognition. Laser Journal, 41(5): 69-100. https://doi.org/10.14016/j.cnki.jgzz.2020.05.096
[24] Narayanan, P., Borel-Donohue, C., Lee, H., Kwon, H., Rao, R. (2018). A real-time object detection framework for aerial imagery using deep neural networks and synthetic training images. In Signal Processing, Sensor/Information Fusion, and Target Recognition XXVII, 10646: 1064614. https://doi.org/10.1117/12.2306154
[25] Rajpura, P.S., Bojinov, H., Hegde, R.S. (2017). Object detection using deep cnns trained on synthetic images. arXiv preprint arXiv:1706.06782.
[26] Xu, G., Zhang, Y., Zhang, Q., Lin, G., Wang, J. (2017). Deep domain adaptation based video smoke detection using synthetic smoke images. Fire Safety Journal, 93: 53-59. https://doi.org/10.1016/j.firesaf.2017.08.004
[27] Lu, Y.Q., Sun, J.Y., Ma, S.W. (2019). Moving object detection based on deep convolutional neural network. Journal of System Simulation, 31(11): 2275-2280. https://doi.org/10.16182/j.issn1004731x.joss.19-FZ0368
[28] Jiang, Y.C., Ji, L.X., Gao, C., Li, S.M. (2018). Research on synthesis data generation method for logo recognition. Chinese Journal of Network and Information Security, 4(5): 21-31. https://doi.org/10.11959/j.issn.2096-109x.2018043
[29] Jin, X.Y., Yin, Q., Ni, J., Zhou, Y.S., Zhang, F., Hong, W. (2020). SAR target detection network based on scenario synthesis and anchor constraint. Journal of Nanjing University of Information Science & Technology, 12(2): 210-215. https://doi.org/10.13878/j.cnki.jnuist.2020.02.008
[30] Xu, J., Ding, X.Q., Wang, S.J., Wu, Y.S. (2009). Detection, location and labeling under a multi-moving-person, multi-view set. Journal of Tsinghua University (Science and Technology), 49(8): 1139-1143.
[31] O'Byrne, M., Pakrashi, V., Schoefs, F., Ghosh, B. (2018). Semantic segmentation of underwater imagery using deep networks trained on synthetic imagery. Journal of Marine Science and Engineering, 6(3): 93. https://doi.org/10.3390/jmse6030093
[32] Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H. (2018). GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing, 321: 321-331. https://doi.org/10.1016/j.neucom.2018.09.013
[33] Wang, J.L., Fu, X.S., Huang, Z.C., Guo, Y.Q., Wang, R.T., Zhao, L.Q. (2020). Multi-type cooperative targets detection using improved YOLOv2 convolutional neural network. Optics and Precision Engineering, 28(1): 251-260. https://doi.org/10.3788/OPE.20202801.0251
[34] Kim, K., Myung, H. (2018). Autoencoder-combined generative adversarial networks for synthetic image data generation and detection of jellyfish swarm. IEEE Access, 6: 54207-54214. https://doi.org/10.1109/ACCESS.2018.2872025