A Modified Centernet for Crack Detection of Sanitary Ceramics

Xiaogang Jia
Research Institue of Intelligent Control and System
Harbin Institute of Technology
Harbin, Heilongjiang 150001, China
Email: 18846827115@163.com

2nd Given Name Surname
Research Institue of Intelligent Control and System
Harbin Institute of Technology
Harbin, Heilongjiang 150001, China
Email: 18846827115@163.com

3rd Given Name Surname
Research Institue of Intelligent Control and System
Harbin Institute of Technology
Harbin, Heilongjiang 150001, China
Email: 18846827115@163.com

***Abstract*—In this paper, we propose a modified Centernet to complete the defect detection of Sanitary Ceramics. Generally, visual quality inspection is rather important during the productive process of Sanitary Ceramics and it is nearly impossible to inspect the massive images by hand. Consequently, it is necessary to devise an accurate and real-time system to process the data. However, due to the varied shapes and backgrounds of ceramics, conventional computer vision methods are usually not robust to all those variables. Detectors based on Deep Learing start to be adopted in recent years, but most algorithms require some carefully devised anchor boxes and post-processing methods, which also bring more computational costs. Here we decide to take advantage of the anchor-free model, Centernet. We change the main structure to fit our own data and use an extra classification branch to help the model focus on the defect areas. The results have shown the great power of this model. Without even any post-processing methods, our model performs very well for both classification and localization.**

***Index Terms*—defection detection, sanitary ceramics, centernet**

I. Introduction

Sanitary ceramics are widely used recently and there have been many types of their products. Generally, sanitary ceramics are attributed to brittle materials, so it is likely to create all kinds of defects during the productive process [1], such as pinhole, crack, blob, scratch etc. Moreover, the processing techologies are pretty complex, which makes the situation even worse. The existence of defects will seriously affect the tightness, fatigue limit, corrosion resistance, abrasion resistance and some other characteristics of sanitary ceramics. All those influences can cause performance degradation and even bring deadly hidden dangers. Consequently, it is imperative to inspect the quality of sanitary ceramics.

Currently, many companies still use human vision to complete quality inspection which has low detection efficiency and cannot guarantee the precision. More importantly, it is hard for manual labours to keep working all the time. Although there are some works that use image processing methods to produce enhanced images to help workers work effectively [2], the key problems are still unresolved. In order to realize automated visual inspection, we need an accurate and robust algorithm to detect the defects in ceramics.

In ceramics industry, researches usually utilize a series of image processing methods to localize the defects in an image [2]–[4], including filtering operators, morphological operators, image binarization and edge detection etc. The main idea of these methods is to extract the important features of defects and then use a classifier to perform classification. However, the practical scenes can be pretty complex and in this case, the extracted features can be unreliable, which may lead to unpredictable issues. Therefore, feature extractors that are robust to different backgrounds are essential.

In the last ten years, thanks to the prosperity of Deep Learning and rapidly improved computational ability, computer vision has made a great of achievements. This success mainly profits from the Convolutional Neural Networks(CNNs). CNNs learn to extract the most valuable features in the images and many works have verified its robustness and effectiveness. In many fields of computer vision such as classification, object detection and instance segmentation, more and more models based on CNNs are proposed in recent years. Therefore, researchers start to take advantage of CNNs to detect defects in ceramics and some detection methods based on CNNs have been proposed [16]–[18].

In this paper, we mainly focus on one of the most important defects in ceramics, which is crack. This kind of defect is usually the most dangerous and common one [1]. Tt is hard to detect all those cracks due to the different scale. Here we modify the standard Centernet [5] to make it perform well in both small and big cracks. Given the low resolution of our DataSet, we make predictions on a feature map with the same scale of the input, which can reserve the important features and guarantee the precision. Besides, to make the network can focus on the area of cracks, we also introduce an extra branch to classify the foreground and background. Finally, we simplify the whole modle structure to realize real-time detection.

The paper is organized as follows: Section II reviews related works on defect detection. Section III describes the main theories of Centernet and the modifications we have made. Section IV describes the details about the backbone structure, training process and inference. Section V shows our experiment settings and detection results. Section VI gives the

final conclusions.

II. Related Work

A. Defect Detection Using Traditional Methods

Before the prevalence of Convolutional Neural Networks, researchers usually took advantage of image processing methods and traditional classifier to detect defects in ceramics.

In [1], image processing techniques like edge detection and morphological operations are used to enchance the original image to make it more suitable for the human to recognize different defects. Hocenski et al. [2] proposed an improved canny edge detector to inspect edges and faults on ceramic tiles. In [3], Rotation Invariant Measure of Local Variance (RIMLV) operator and a morphological operator are applied to detect the defect regions, and then a multi-class support vector machine classifier is used to classify the defects. Artificial Neural Networks(ANN) are also used in some defect detection systems. In [4], a five-layers neural network was proposed to classify the defective and defectless feature vectors. Liu S et al. proposed a PSO neural network to detect defects in fabrics in [5]. The paper determined that the PSO-BP neural network has shorter training period and higher accuracy.

B. Defect Detection Using Deep Learning

Due to the great success in AlexNet [6], Convolutional Neural Networks started to show its efficiency and accuracy in both classification and detection. In the field of object detection, detectors usually can be divided into two patterns: two-stage detectors and one-stage detectors. Represented by the series of R-CNN [7]–[9] model, two-stage methods generally split the whole process into two parts. First, the model extracts thousands of proposed regions which are likely to contain objects. Then the followed subnetwork will classify all those regions and regress the corresponding boxes. For now, many improved two-stage methods have been developed, such as R-FCN [10], Mask-RCNN [11], Cascade-RCNN [12] etc. Different from the two-stage algorithms, one-stage detectors do not extract ROIs and directly process the classification and localization on the original image. Yolov3 [13] and SSD [14] are widely used in the industrial field. RetinaNet [15] introduced Focal Loss to alleviate the imbalance between foreground and background.

Li et al. [16] used SSD to perform surface defect detection. They adopted MobileNet [17] as the backbone to achieve real-time detection. In [18], an improved Faster R-CNN was used to detect the defects on irregular surfaces of sanitary equipment.

Whether a detector is two-stage or one-stage, generally it needs several prior bounding boxes or so-called anchor boxes so it can regress the bounding boxes. But the scale and shape of defects vary in different environment. In this case, it is hard to devise appropriate bounding boxes and using anchor boxes can bring more computational costs. Since the popular anchor-free model, CornerNet [19], were developed by Law and Deng, a series of anchor-free networks have been proposed in recent years [20]–[24]. Most of these detectors regard keypoints like corner points or center points as positive samples, and then regress the shape of the objects.

III. The Modified CenterNet

The main difference between anchor-based methods and anchor-free methods is actually how to define positive samples. Most anchor-based models regrad anchor boxes whose IOU with groundtruths is bigger than a threshold(usually 0.5) as positive samples. In CenterNet [22], the center point of a object is a positive sample. In order to detect an object, we need to identify the center and then regress the width and height at that point. At present, the most powerful structure for keypoints detection is Hourglass [25]. However, this backbone is so complex that it can hardly achieve real-time detection in industrial application. Except for Hourglass srtucture, ResNet [26] and DLA [27] were also discussed in [22]. To improve the efficiency of CenterNet without sacrificing accuracy, we modify the original model based on ResNet. The overview of CenterNet is shown in Fig. 1.

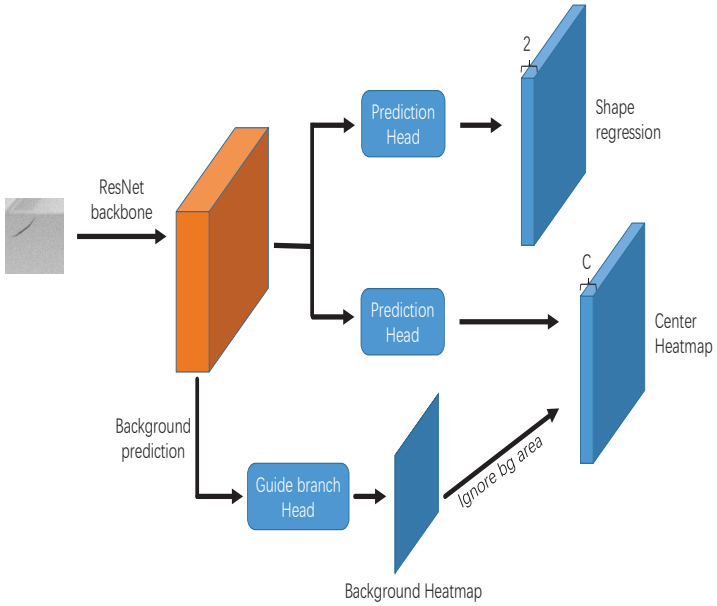
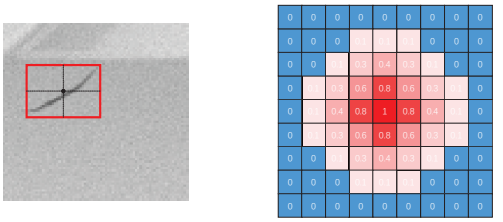


Fig. 1. Overview of CenterNet. Here we add an extra branch to predict backgrounds and the results are used to define an ignoring area for the Center Heatmap.

A. Detecting Centers

For an image of size $H \times W \times 3$, we predict a center heatmap to perform both classification and localization. The size of each heatmap is $C \times H \times W$. It has C channels which indicate the categories. Here we only detect cracks on sanitary ceramics, so $C = 1$. We do not take advantage of the channel of background, because we find that this will seriously affect the prediction scores of the foreground. We reduce the resolution in the ResNet by $4\times$ and then upsample the feature map to the original scale. Therefore, the resolution of input image equals that of center heatmap. For a crack defect, only the center of its bounding box is positive and it is set by 1. All



(a) We use the center to represent a crack.

(b) Gaussian Lable. The center of the Gaussian function is the positive point.

Fig. 2. heatmap

other locations are negative and are set by 0. However, in this case, positive samples are much less than negative samples. This imbalance will degrade the performance of the whole model. So for points that are around the center, CenterNet reduces their contributions to the loss according to a Gaussian function [19] [22]. That fuction is given by $e^{-\frac{x^2+y^2}{2\sigma^2}}$. The parameter σ is determined by the radius r of the area around the center (here we adpot the method in [19] that bounding boxes produced by points in the area must have at leats 0.3 IOU with the groundtruth). Then $\sigma = \frac{1}{3}r$. The Gaussian label of the center heatmap is shown in Fig. 2.

The training loss is same as the loss in [19].

$$L_{cls} = \frac{-1}{N}\sum_{c=1}^{C}\sum_{i=1}^{H}\sum_{j=1}^{W}\begin{cases}(1-p_{cij})^{\alpha}log(p_{cij}) & \text{if } y_{cij}=1\\ (1-y_{cij})^{\beta}(p_{cij})^{\alpha}log(1-p_{cij}) & \text{if } y_{cij}<1\end{cases} \tag{1}$$

B. Bounding Boxes Regression

For each defect in an image, we predict its shape at the center point. Assuming that $(x^i_{min}, y^i_{min}, x^i_{max}, y^i_{max})$ is the label of defect i. The size of prediction layer in the shape subnetwork is $2 \times H \times W$. We assign $box^*_i = (x^i_{max} - x^i_{min}, y^i_{max} - y^i_{min})$ at the center of defect i. Then we apply smooth L1 Loss [8] to regress boungding box at each positive point:

$$L_{reg} = \frac{1}{N}\sum_{i=1}^{N} smooth_{L_1}(box^*_i - box_i) \tag{2}$$

where $smooth_{L_1}(x)$ is defined as:

$$smooth_{L_1}(x) = \begin{cases}0.5x^2, & \text{if } |x|<1\\ |x|-0.5, & \text{otherwise}\end{cases} \tag{3}$$

C. Background Prediction

In the original CenterNet, we observe a serious problem that scores in the center heatmap are relatively lower than socres in other detectors, especially for small objects. This phenomenon may lead to misses for some defects during the inference, because we need use a score threshold to filter other backgrounds. Therefore, we think that if the center heatmap branch could only focuses on defect areas, then the scores of positive points may get increased. So we just add an extra branch to filter most areas of backgrounds.

The size of background heatmap is $1 \times H \times W$. We simply regard all points that fall out of the center area(defined by radius$r$) as positive samples in this branch. We use the same focal loss in the part of Detecting Centers and all the hyper-parameters are unchanged. During the training and test, the point whose score is higher than 0.5 in the background heatmap is just ignored in the center heatmap. In this way, the center prediction head can give more reliable results.

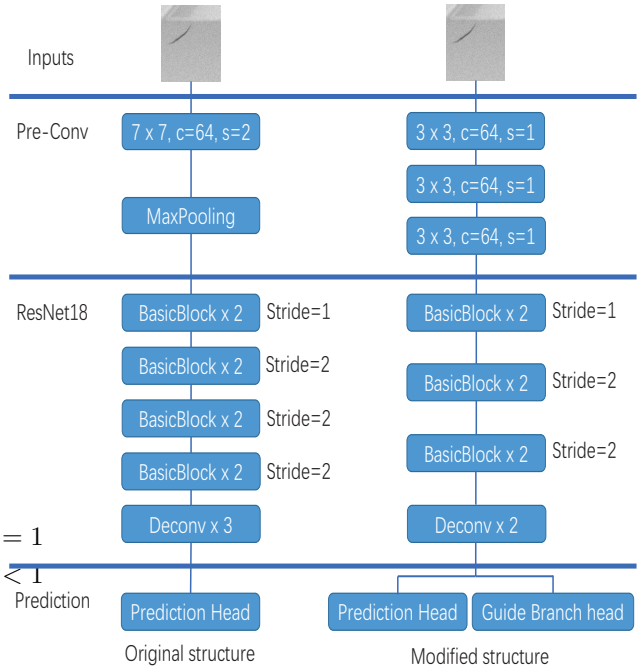


Fig. 3. Overview of CenterNet

IV. Implementation Details

Considering the limited amount of our data, a complex deep convolutional neural network may lead to serious overfitting. Therefore, we modify the standard Centernet based on the ResNet18 backbone.

A. ResNet

Residual network is first proposed in [26], aiming to accelerate the optimization process and alleviate the problem of vanishing gradients. After that, many object detection works also prove the efficiency of residual networks. Here we adopt the basic backbone, ResNet18.

According to [29], the downsampling operation in the first convolution layer may degrade the performance of the model, especially for small objects. But in most cases, the input resolution is relatively big, such as 416, 512, 608 or 800. So if they remove the downsampling operation at first, the model may suffer from the extra computational costs. Based on the fact that our input resolution is pretty small, so we do not need to worry about that problem. In consequence, we replace the

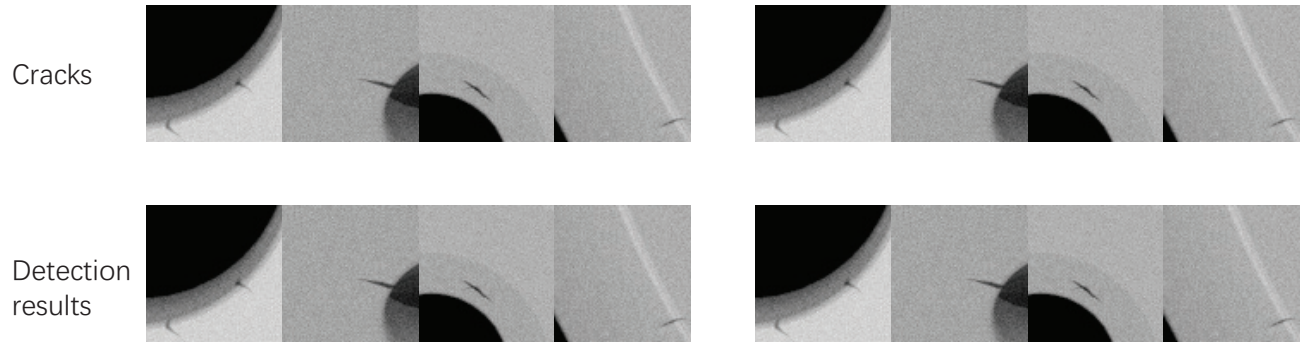


Fig. 4. examples for dataset

first 7 × 7 convolution layer with stride 2 by three stacked 3 × 3 convolution layers with stride 1. To save computational costs, the channel of each 3 × 3 convolution layer is set by 64. The maxpooling layer is simply removed.

In [29], they also stress the importance of BatchNorm [30] in both backbone subnetwork and detection subnetwork. However we find that BatchNorm layer in detection head is not helpful in our case. So we add BatchNorm after every convolution layer except for the detection part. We apply three stacked 3 × 3 convolution layers and one 1 × 1 convolution layer for both classification and regression. The parameters are not shared in these two branches. The comparison between the whole model structure and the original CenterNet structure are showed in Fig. 3.

B. Training

The resolution of all images in our dataset is 80 × 80. Since the stride in the whole network is 1, the output resolution is also 80 × 80. To relieve the problem of overfitting, we use random cropping, random horizontal flipping, random vertical flipping and random color jittering to augment the original data. We use Adam [28] to optimize the training loss. The total loss is the sum of losses for three branches.

$$Loss = \alpha L_{cls} + \beta L_{reg} + \gamma L_{guide} \tag{4}$$

where α, β and γ are the weights for three branches respectively. Here $\alpha = 1.0$, $\beta = 0.1$, $\gamma = 1.0$. Since the model structure is relatively simple and the training resolution is small, we can use a batch size of 16 and train the whole model with only one GPU. We train the whole network for 50 epochs with initial learning rate 2.5×10^{-4}. We also drop the learning rate to 2.5×10^{-5} after 40 epochs.

C. Inference

During inference, we first pick points in the Background Heatmap whose scores are higher than 0.5. Then in the predicted Center Heatmap, scores of those points are set by 0. We use a 3 × 3 max pooling layer on the Center Heatmap to filter additional points around the centers. Then we set a threshold of 0.1 and points whose scores are lower than that threshold are ignored. As a result, the left points are the centers of cracks. For every center point, Center Heatmap provides its location (x^i, y^i). Then the branch of Shape Regression gives its shape (w^i, h^i). So we can get the corresponding bounding box:

$$\begin{aligned} x_{min} &= x^i - w^i/2 \\ y_{min} &= y^i - y^i/2 \\ x_{max} &= x^i + y^i/2 \\ y_{max} &= y^i + y^i/2 \end{aligned} \tag{5}$$

where (x_{min}, y_{min}) is the top-left point and (x_{max}, y_{max}) is the bottom-right point of the crack.

V. Experiments

We implement the modified CenterNet using PyTorch 1.2.0, CUDA 10.0, and CUDNN7.1. The size of the model is only 30Mb, which allows us to run both the training and testing process on our local computer with Inter Core i7-8750H CPU and GeForce GTX 1060.

A. Data Description

The whole dataset is divided into two parts: 907 images with defects and 1500 images with only backgrounds. We train the modified CenterNet only on the images with defects and the training samples contain 726 images and the testing samples contain 181 images. Several images are shown in Fig. 4.

For images with defects, we adopt the metric of average precision(AP) which was first proposed in VOC2007 [32] to evaluate the model. A bounding box is defined as positive if its IOU with the groungtruth is higher than 0.5. For images with only backgrounds, we still use the modified CenterNet to process them and if the output of an image contains predicted bounding boxes, then this image is regarded as a negative sample.

TABLE I
MAIN DETECTION RESULTS

Method	Backbone	Resolution	Downsampling Scale	Model Size	AP	FPS
CenterNet [23]	ResNet-18	80	4	30Mb	0.3	0.4
ours:						
CenterNet	Modified ResNet-18	80	1	30.0Mb	0.3	0.4
CenterNet + Adaptive Feature Fusion	Modified ResNet-18	80	1	30.3Mb	0.3	0.4

B. Detection Results

Table 1 shows our results on the defect data. To verify the effectiveness of our modifications, we also train the original CenterNet on the same dataset and all hyper parameters remain unchanged as our main settings.

C. Detection Results

We also evaluate our model on images with only backgrounds. The results are shown in Table 2.

TABLE II
PREDICTIONS FOR BACKGROUNDS

images	FP	3	Precision
1500	0.2	0.3	0.4

CONCLUSION

In this paper

REFERENCES

[1] Karimi M H, Asemani D. Surface defect detection in tiling Industries using digital image processing methods: Analysis and evaluation[J]. ISA transactions, 2014, 53(3): 834-844.

[2] Elbehiery H, Hefnawy A, Elewa M. Surface defects detection for ceramic tiles using image processing and morphological techniques[J]. 2005.

[3] Hocenski Z, Vasilic S, Hocenski V. Improved canny edge detector in ceramic tiles defect detection[C]//IECON 2006-32nd Annual Conference on IEEE Industrial Electronics. IEEE, 2006: 3328-3331.

[4] Hanzaei S H, Afshar A, Barazandeh F. Automatic detection and classification of the ceramic tiles' surface defects[J]. Pattern Recognition, 2017, 66: 174-189.

[5] Branca A, Delaney W, Lovergine F P, et al. Surface defect detection by a texture analysis with a neural network[C]//Proceedings of 1995 IEEE International Conference on Robotics and Automation. IEEE, 1995, 2: 1497-1502.

[6] Liu S, Liu J, Zhang L. Classification of fabric defect based on PSO-BP neural network[C]//2008 Second International Conference on Genetic and Evolutionary Computing. IEEE, 2008: 137-140.

[7] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[8] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[9] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[10] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]//Advances in neural information processing systems. 2015: 91-99.

[11] Dai J, Li Y, He K, et al. R-fcn: Object detection via region-based fully convolutional networks[C]//Advances in neural information processing systems. 2016: 379-387.

[12] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.

[13] Cai Z, Vasconcelos N. Cascade r-cnn: Delving into high quality object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6154-6162.

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

[15] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21-37.

[16] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.

[17] Li Y, Huang H, Xie Q, et al. Research on a surface defect detection algorithm based on MobileNet-SSD[J]. Applied Sciences, 2018, 8(9): 1678.

[18] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.

[19] Zhou Z, Lu Q, Wang Z, et al. Detection of Micro-Defects on Irregular Reflective Surfaces Based on Improved Faster R-CNN[J]. Sensors, 2019, 19(22): 5000.

[20] Law H, Deng J. Cornernet: Detecting objects as paired keypoints[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 734-750.

[21] Tian Z, Shen C, Chen H, et al. Fcos: Fully convolutional one-stage object detection[C]//Proceedings of the IEEE International Conference on Computer Vision. 2019: 9627-9636.

[22] Zhou X, Zhuo J, Krahenbuhl P. Bottom-up object detection by grouping extreme and center points[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 850-859.

[23] Zhou X, Wang D, Krähenbühl P. Objects as points[J]. arXiv preprint arXiv:1904.07850, 2019.

[24] Kong T, Sun F, Liu H, et al. Foveabox: Beyond anchor-based object detector[J]. arXiv preprint arXiv:1904.03797, 2019.

[25] Zhu C, He Y, Savvides M. Feature selective anchor-free module for single-shot object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 840-849.

[26] Newell A, Yang K, Deng J. Stacked hourglass networks for human pose estimation[C]//European conference on computer vision. Springer, Cham, 2016: 483-499.

[27] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[28] Yu F, Wang D, Shelhamer E, et al. Deep layer aggregation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 2403-2412.

[29] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[30] Zhu R, Zhang S, Wang X, et al. ScratchDet: Training single-shot object detectors from scratch[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2019: 2268-2277.

[31] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.

[32] Everingham M, Van Gool L, Williams C K I, et al. The pascal visual object classes (voc) challenge[J]. International journal of computer vision, 2010, 88(2): 303-338.