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

Xianqiang Yang
Research Institue of Intelligent Control and System
Harbin Institute of Technology
Harbin, Heilongjiang 150001, China
Email: xianqiangyang@hit.edu.cn

Huijun Gao
Research Institue of Intelligent Control and System
Harbin Institute of Technology
Harbin, Heilongjiang 150001, China
Email: hjgao@hit.edu.cn

***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 Learning 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 introduce an extra branch with shallow layers to strengthen the feature representation. 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*—defect 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 classified as 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 technologies 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 [5]–[7].

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]. It is hard to detect all those cracks due to their different scales. Here we modify the standard CenterNet [8] to make it perform well in cracks of varied shapes. 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 strengthen the power of the whole network, we also add an extra branch to fuse its features with the backbone's features. Finally, we simplify the whole model 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.

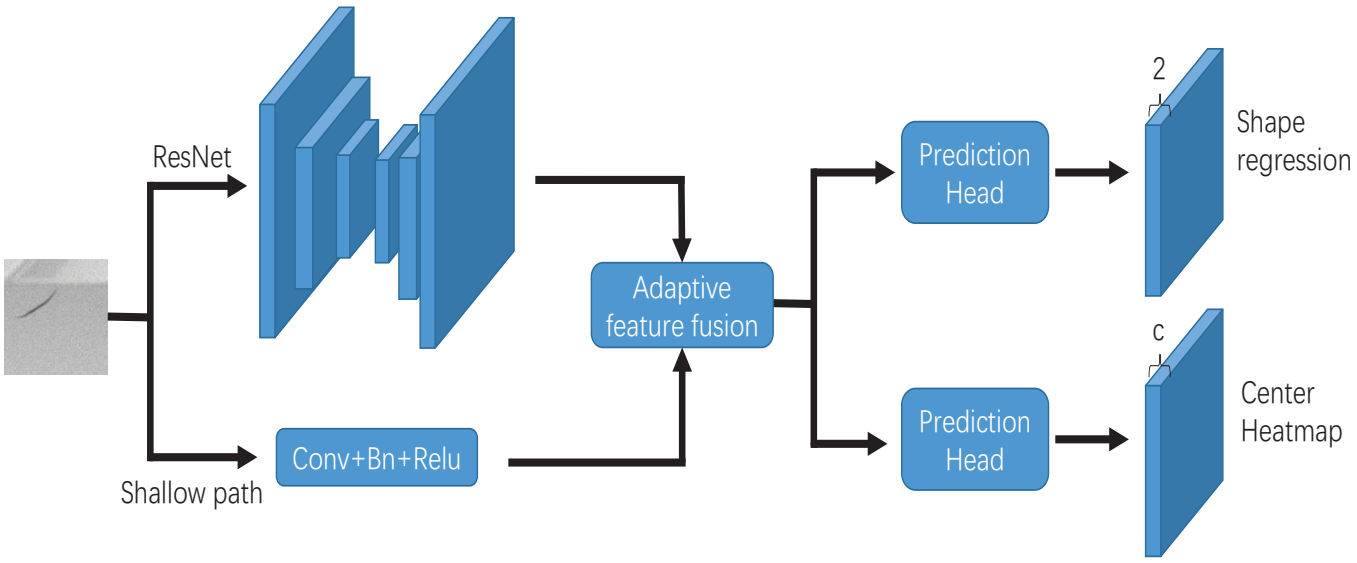


Fig. 1. Overview of CenterNet. Here we add an extra branch to utilize the features from the first layer. We also adaptively fuse the main features with the features from the extra branch.

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 [2], image processing techniques like edge detection and morphological operations are used to enhance the original image to make it more suitable for the human to recognize different defects. Hocenski et al. [3] proposed an improved canny edge detector to inspect edges and faults on ceramic tiles. In [4], 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 [9], a five-layers neural network was proposed to classify the defective and defectless feature vectors. Liu et al. proposed a PSO neural network to detect defects in fabrics [10]. 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 of AlexNet [11], 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 [12]–[14] 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 [15], Mask-RCNN [16], Cascade-RCNN [17] 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 [18] and SSD [19] are widely used in the industrial field. RetinaNet [20] introduced Focal Loss to alleviate the imbalance between foreground and background.

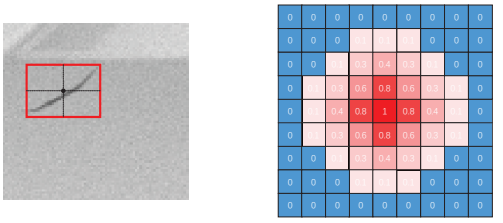
Li et al. [5] used SSD to perform surface defect detection. They adopted MobileNet [6] as the backbone to achieve real-time detection. In [7], an improved Faster R-CNN was used to detect the defects on irregular surfaces of sanitary equipment.

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 to the groundtruths. But the scales and shapes of defects vary in different environment. In this case, it is hard to devise appropriate bounding boxes and besides, using anchor boxes can bring more computational costs. Since the popular anchor-free model, CornerNet [21], were developed by Law and Deng, a series of anchor-free networks have been proposed in recent years [8], [22]–[25]. 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 regard anchor boxes whose IOU with groundtruths is bigger than a threshold(usually 0.5) as positive samples. In CenterNet [8], the center point of an 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 [26]. However, this backbone is so complex that it can hardly achieve real-time detection in industrial application. Except for Hourglass structure, ResNet [27] and DLA [28] were also discussed in [8]. 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.



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

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

Fig. 2. Illustration of 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 C 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 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 [21] [8]. 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 (to determine r, here we adopt the method in [21] that bounding boxes produced by points in the area must have at least 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 [21]. Assuming that p_{cij} is the predicted score of class c at the point (i, j) and y_{cij} is the corresponding label, the loss is defined as:

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

where N is the number of cracks in an image and α and β are hyper-parameters. This loss is a variant of Focal Loss [20] and it additionally introduces β to control the contributions of points around positive samples. Here $\alpha = 2$ and $\beta = 4$.

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 [13] to regress bounding box at each positive point:

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

where box^*_i is the predicted bounding box and $smooth_{L_1}(x)$ is defined as:

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

C. Adaptive Feature Fusion

According to [29], for a deep Convolutional Neural Network, features from layer1 and layer2 generally represent corners, colors and edges etc. With the network getting deeper, features are more discriminative and better for classification. Therefore, we try to combine the features from both shallow and deep layers. To achieve an effective fusion, we choose to let the network learn to determine the weights between those features. In [30], an adaptive feature fusion method is used to fuse features from feature pyramids. Here we directly use a 3×3 convolutional layer to process the input image and adaptively fuse the features with the main features from ResNet. Furthermore, this combination provides a shortcut between the input and the deep layers. Such a shortcut is also used in [31] and the effectiveness has also been proved, but here the shortcut consists of only one convolutional layer. In this case, the whole model has a notable improvement without bringing much computational costs.

We denote the features from ResNet as P_{res} and the features from the original image as P_{img}. Then the fusion function is given by:

$$P_{fusion} = a \cdot P_{res} + b \cdot P_{img} \quad (4)$$

where a and b represent the weights of features from the ResNet and shallow layer respectively. Here a and b are computed by 1×1 convolutional layer followed by a softmax function. Therefore, the network can learn to form the best features which are beneficial for both classification and regression.

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.

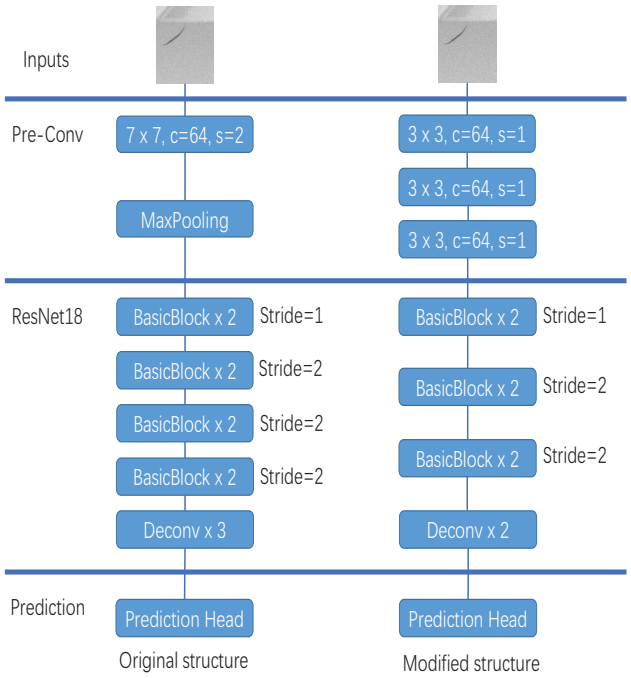


Fig. 3. Comparison between the original CenterNet and the modified one.

A. ResNet

Residual network is first proposed in [27], 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 [32], 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 first 7×7 convolutional layer with stride 2 by three stacked 3×3 convolutional 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 [32], they also stress the importance of BatchNorm [33] 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 [34] to optimize the training loss. The total loss is the sum of losses for two branches.

$$Loss = \alpha L_{cls} + \beta L_{reg} \tag{5}$$

where α and β are the weights for two branches respectively. Here $\alpha = 1.0$, $\beta = 0.1$. 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 25 epochs with initial learning rate 2.5×10^{-4}. We also drop the learning rate to 2.5×10^{-5} after 20 epochs.

C. Inference

During inference, we use a 3×3 max pooling layer on the Center Heatmap to filter additional points around the centers. Then we choose top 20 points and 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:

$$\begin{aligned} x_{min} &= x^i - w^i/2 \\ y_{min} &= y^i - y^i/2 \\ x_{max} &= x^i + y^i/2 \\ y_{max} &= y^i + y^i/2 \end{aligned} \tag{6}$$

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 34.3Mb, 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.

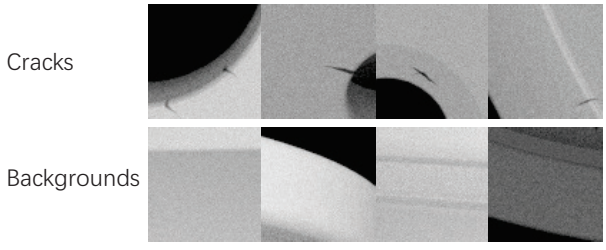


Fig. 4. Examples for dataset

TABLE I
MAIN DETECTION RESULTS

Method	Backbone	Resolution	Downsampling Scale	Volume	AP	FPS
CenterNet [8]	ResNet-18	384	4	75.7Mb	94.20	20
ours:						
CenterNet	Modified ResNet-18	80	1	34.0Mb	95.48	33
CenterNet + Adaptive Feature Fusion	Modified ResNet-18	80	1	34.3Mb	96.16	30

A. Data Description

The whole dataset is divided into two parts: 907 images with defects and 1523 images with only backgrounds. We train the modified CenterNet only on the images with defects. 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 [35] to evaluate the model. A bounding box is defined as positive if its IOU with the groundtruth 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.

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. Due to the large downsampling scale of the original ResNet18, we have to train the original CenterNet at resolution of 384. From the table, the modified ResNet-18 can achieve the AP(95.48), which is 1.28 higher than the AP(94.20) of original backbone and the FPS is also pretty fast, so it can perform real-time detection. After we add the adaptive feature fusion branch, the AP reach 96.16 and it does not bring too much extra computation costs, which proves the effectiveness of this branch.

We also evaluate our model on images with only backgrounds. The results are shown in Table 2. It can be seen that our model is also robust to images without cracks.

TABLE II
PREDICTIONS FOR BACKGROUNDS

images	TP	FN	Accuracy
1523	1492	31	97.96

VI. CONCLUSION

In this paper, we modify the original CenterNet based on ResNet18 to detect cracks on the surface of Sanitary Ceramics. We first simplify the whole backbone, so the model can run pretty fast on an ordinary GPU. Then to augment the feature representation, we remove the downsampling operation at the first two layers and use an adaptive feature fusion method to combine the features from shallow layer and deep layer. By experiments, the predicted results of cracks achieve the AP(96.16). Therefore, we conclude that our modified CenterNet is pretty effective and it has satisfied the requirements of real-time detection.

REFERENCES

[1] M. H. Karimi and D. Asemani, "Surface defect detection in tiling industries using digital image processing methods: Analysis and evaluation," *ISA transactions*, vol. 53, no. 3, pp. 834–844, 2014.

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

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

[4] S. H. Hanzaei, A. Afshar, and F. Barazandeh, "Automatic detection and classification of the ceramic tiles' surface defects," *Pattern Recognition*, vol. 66, pp. 174–189, 2017.

[5] Y. Li, H. Huang, Q. Xie, L. Yao, and Q. Chen, "Research on a surface defect detection algorithm based on mobilenet-ssd," *Applied Sciences*, vol. 8, no. 9, p. 1678, 2018.

[6] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[7] Z. Zhou, Q. Lu, Z. Wang, and H. Huang, "Detection of micro-defects on irregular reflective surfaces based on improved faster r-cnn," *Sensors*, vol. 19, no. 22, p. 5000, 2019.

[8] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[9] A. Branca, W. Delaney, F. P. Lovergine, and A. Distante, "Surface defect detection by a texture analysis with a neural network," *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1497–1502, 1995.

[10] S. Liu, J. Liu, and L. Zhang, "Classification of fabric defect based on pso-bp neural network," *2008 Second International Conference on Genetic and Evolutionary Computing*, pp. 137–140, 2008.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[13] R. Girshick, "Fast r-cnn," *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, pp. 91–99, 2015.

[15] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *Advances in neural information processing systems*, pp. 379–387, 2016.

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

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

[18] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *European conference on computer vision*, pp. 21–37, 2016.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[21] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 734–750, 2018.

[22] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9627–9636, 2019.

[23] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 850–859, 2019.

[24] T. Kong, F. Sun, H. Liu, Y. Jiang, and J. Shi, "Foveabox: Beyond anchor-based object detector," *arXiv preprint arXiv:1904.03797*, 2019.

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

[26] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," *European conference on computer vision*, pp. 483–499, 2016.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[28] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2403–2412, 2018.

[29] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *European conference on computer vision*, pp. 818–833, 2014.

[30] S. Liu, D. Huang, and Y. Wang, "Learning spatial fusion for single-shot object detection," *arXiv preprint arXiv:1911.09516*, 2019.

[31] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, 2018.

[32] R. Zhu, S. Zhang, X. Wang, L. Wen, H. Shi, L. Bo, and T. Mei, "Scratchdet: Training single-shot object detectors from scratch," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2268–2277, 2019.

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

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

[35] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.