Strip Steel Defect Detection Based on Improved YOLOv5s

Jie Wang and Qiu Fang
Xiamen University of Technology, Xiamen, China

Keywords: YOLOv5s, Defect Detection, Attention Mechanism, Multi-Scale Pyramidal Detection Head, Small Target.

Abstract: In response to the issues of low detection accuracy for surface defects in strip steel and difficulty in detecting small target defects in modern steel production processes, this study presents an improved algorithm based on YOLOv5s is proposed for detecting strip-steel surface defects. Firstly, an improved C3 module combining large-kernel depth separable convolution and Squeeze-and-Excitation (SE) attention mechanism is proposed, which increases the global receptive field of the network while adaptively adjusting the weight relationships among different channels in order to enhance the fusion of tiny features in the model. Secondly, A multi-scale pyramidal detection head is introduced as a means to enhance the model's proficiency in detecting small targets. Finally, the SIoU loss function is employed to more accurately calculate the regression loss and improve the model's detection accuracy. The results indicate that the mean average precision (mAP) of the proposed algorithm on the NEU-DET dataset reaches 80.6%, an increase of 3.4% compared to the baseline algorithm; moreover, the detection speed reaches 79.3f/s, which meets the real-time requirement of industrialised defect detection.

1 INTRODUCTION

As industrialization continues to advance, the demand for strip steel, a pivotal industrial raw material, is on the rise. However, several factors, including the inherent instability of raw materials, the intricacies of the rolling process, and the challenges associated with system control, have given rise to surface defects on strip steel. These defects encompass issues such as burrs, scratches, and patches. Notably, these surface imperfections exert a substantial impact on the critical properties of strip steel, including corrosion resistance and strength. Consequently, they significantly diminish the overall performance and service life of strip steel when deployed in practical applications. Therefore, the detection of surface defects on strip steel assumes paramount importance in ensuring the quality and reliability of this essential material.

In recent years, the prevailing approach in strip steel defect detection has shifted towards deep learning, driven by the remarkable advancements in convolutional neural networks. Yang et al.0 introduced a pixel-level deep segmentation network for automated defect detection and successfully built an end-to-end defect segmentation model. This model exhibited outstanding performance in terms of defect recognition and localization. Akhya et al.0 introduced a powerful defect detector (FDD) based on the Cascade R-CNN algorithm for surface defect detection in steel materials. Yu et al.0 proposed a bidirectional feature fusion network combining channel attention and FCOS detector to achieve fast detection of steel strips. Tang et al.0 introduced an innovative steel plate surface defect detection approach based on deep learning, incorporating the Transformer architecture. Specifically, they utilized the Swin Transformer module to extract features from strip steel images, thereby enhancing the network's feature extraction capabilities but less real-time. Jian et al.0 proposed a multi-scale cascaded attention network based on ResNet34 to enhance the extraction of high-level semantic features. However, the size of the model is large and not easy to deploy for edge devices. Zhao et al.0 introduced an enhanced steel detection method based on YOLOv5, leveraging the Res2Net module to expand the receptive field. Additionally, incorporated the Dual Feature Pyramidal Network to bolster the extraction of critical neck features in the process.

In view of above problems, this paper chooses YOLOv5s as the baseline model and proposes improvements on it. To tackle the challenge of distinguishing between the background and targets in strip steel surface defects, we introduce the large kernel depth separable convolution (Dsconv)0 to

Wang, J. and Fang, Q.
Strip Steel Defect Detection Based on Improved YOLOv5s.
DOI: 10.5220/0012284300003807

In *Proceedings of the 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (ANIT 2023)*, pages 375-380
ISBN: 978-989-758-677-4

enhance the network's receptive field. Additionally, we incorporate the SE attention mechanism to enhance the network's feature extraction capability. We also have introduced a multi-scale pyramidal detection head structure. This addition is intended to improve the model's ability to effectively detect smaller targets. Additionally, to address the issue of diverse defect shapes and significant variations in size within the dataset, we have replaced the original CIoU loss function in the network with an SIoU loss function. This modification serves to expedite the model's convergence speed, particularly in scenarios where defect shapes and sizes vary considerably.

2 METHODS

2.1 Improved YOLOv5s Network Structure

In this paper, we introduce three significant improvements to enhance the YOLOv5s network. First, we enhance the C3 module of the network by replacing it with the improved DsSE_C3 module. To strike a balance between model size and inference speed, the Ds_C3 component within the Neck structure does not employ the SE attention mechanism. Secondly, we make improvements to the Head of the network. The first detection head of the is substituted with a multi-scale pyramidal detection head. The overall architecture of the improved YOLOv5s network is visualized in Fig. 1. These enhancements collectively contribute to improving the network's performance in detecting surface defects on strip steel.

1) Introduction of SE attention mechanism

The SE attention mechanism module that can adaptively learn the weight relationship between different channels0, as shown in Fig. 2. It can be divided into the following steps: the first step is the transformation operation, for which the input X is mapped by any given F_{tr} transformation to a feature map U. The second step is the squeezing operation, which generates channel descriptors and thus global distributions embedded in the channel feature responses through the feature map U, so that the

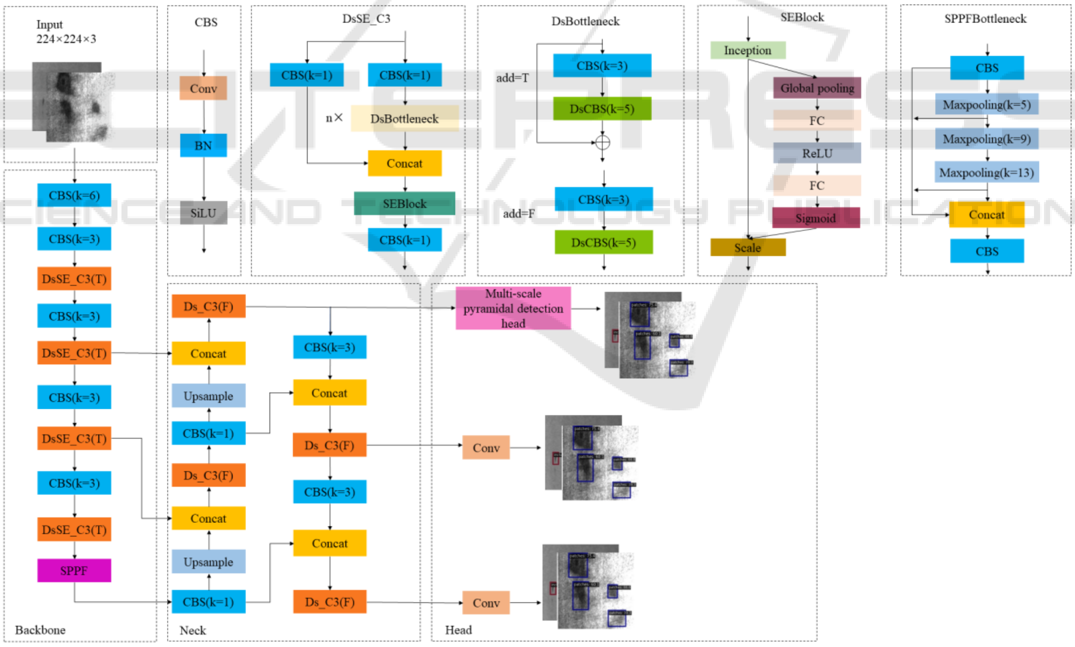

Figure 1. Improved network structure of YOLOv5s

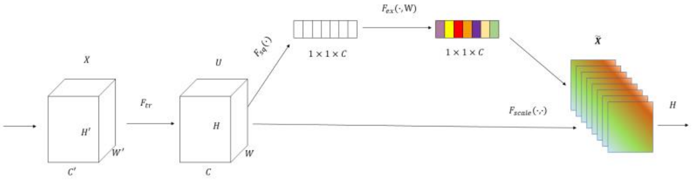

Figure 2. Structure diagram of SE attention mechanism.

network learns the information in the global sensory domain. The third step is the excitation operation, which employs a self-gating mechanism to generate the set of modulation weights for each channel. The fourth step is the fusion operation, which multiplies the modulated weight set and the feature map U on a channel-by-channel basis to adjust the weights between the channels, to mine the links between the features and to improve the performance of the model.

2) Improved C3 module

In order to strengthen the feature extraction capability of the model, we improve the C3 module in this paper, as shown in Fig. 3. Inspired by RTMDet0, we introduce a large kernel 5×5 depth-separable convolution into the basic C3 construction block of YOLOv5s to increase the effective receptive field and capture and model image semantic information more comprehensively. The DsBottleneck structure of the depth separable convolution module with the introduction of larger convolution kernels is shown in Fig. 3 (a). In addition, we introduce the SE channel attention mechanism module in Backbone's C3 module, which enables the model to focus on the feature relationships in the channel dimension to improve defect detection, and the improved DsSE_C3 is visually depicted in Fig. 3 (b).

3) Introduction of multi-scale pyramidal detection head

The Pyramidal Convolution (PyConv)0 structure, which performs multi-scale decomposition of the feature maps by multiple convolution operations with different convolution kernel sizes and depths, and cascades the decomposed feature maps to enhance the perceptual range of the network, which in turn improves the model performance and performance. In this paper, we combine PyConv with the first output detection header of YOLOv5s model, and add residual connectivity by borrowing the idea of Resnet0. A multi-scale pyramidal detection head is proposed, as shown in Fig. 4. The Pyramidal Convolutional Kernels (PyConv Kernels) in the figure is a double oriented pyramidal convolutional kernel. On one side the kernel size is increasing and on the other side the kernel depth (connectivity) is decreasing. It allows the network to explore from large receptive fields with low connectivity to small receptive fields with high connectivity, which brings about complementary image semantic information and enhances the network's multi-scale perception. In addition residual connectivity is added to enhance the expressive capability of the network without degrading the performance. The multi-scale pyramidal detection head effectively enhances the model for small target defect detection.

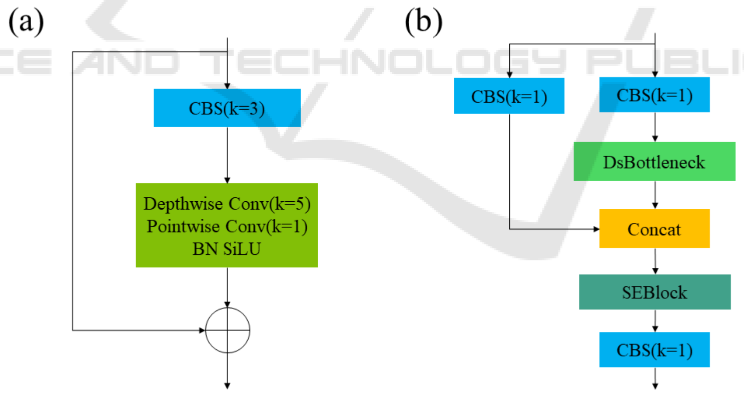

Figure 3. Improved C3 module. (a) DsBottleneck. (b) DsSE_C3.

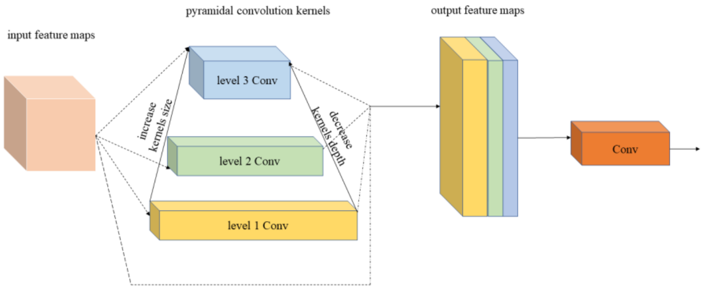

Figure 4. Structure diagram of multi-scale pyramidal detection head.

4) Modifying the regression loss function

The CIoU loss function0 was used in YOLOv5 (V6.1) version to calculate the regression loss of the prediction bounding box. In order to improve the detection performance, the SIoU loss function is used in this paper. It consists of four components: angular loss, distance loss, shape loss and IoU loss0. The angular loss is achieved by reducing the values of the distance-related variables and can be defined as follows:

$$\Lambda = 1 - 2 \times \sin^2(\arcsin(\frac{c_h}{\sigma}) - \frac{\pi}{4}) \tag{1}$$

Where c_h Indicates the height difference between the central point of the real bounding box and the predicted bounding box, as shown in equation (2). σ denotes the distance between the central points, as shown in equation (3). $(b_{c_x}^{gt}, b_{c_y}^{gt})$ is the coordinate of the central point of the real bounding box, and (b_{c_x}, b_{c_y}) is the coordinate of the central point of the predicted bounding box.

$$c_h = \max\left(b_{c_y}^{gt}, b_{c_y}\right) - \min\left(b_{c_y}^{gt}, b_{c_y}\right) \tag{2}$$

$$\sigma = \sqrt{\left(b_{c_x}^{gt} - b_{c_x}\right)^2 + \left(b_{c_y}^{gt} - b_{c_y}\right)^2} \tag{3}$$

The angular loss is mainly used to assist in calculating the distance between two bounding boxes to further approximation of the centers of the two bounding boxes. The distance loss is defined as follows:

$$\Delta = \sum_{t=x,y} (1 - e^{-\gamma\rho t}) \tag{4}$$

Where $\rho_x = (\frac{b_{c_x}^{gt} - b_{c_x}}{c_w})^2$, $\rho_y = (\frac{b_{c_y}^{gt} - b_{c_y}}{c_h})^2$, $\gamma = 2 - \Lambda$, (c_w, c_h) is the width and height of the smallest outer rectangle of the true and predicted bounding boxes.

The shape loss between the two bounding boxes is defined as follows:

$$\Omega = \sum_{t=w,h} (1 - e^{-w_t})^\theta = (1 - e^{-w_w})^\theta + (1 - e^{-w_h})^\theta \tag{5}$$

Where $w_w = \frac{|w - w^{gt}|}{\max(w, w^{gt})}$, $w_h = \frac{|h - h^{gt}|}{\max(h, h^{gt})}$, (w, h), (w^{gt}, h^{gt}) denote the width and height of the predicted and real boxes, respectively.

In summary, the SIoU loss function is defined as follows:

$$Loss_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{6}$$

Where $IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|}$ denotes the intersection and concurrency ratio between the true and predicted bounding boxes.

3 EXPERIMENTS AND RESULT ANALYSIS

3.1 Datasets

The NEU-DET dataset comprises six distinct defect classes: crazing, inclusion, patches, pitted_surface, rolled-in_scale and scratches. Each defect class is comprised of 300 grayscale images, each having a resolution of 200 pixels, as visually presented in Fig. 5. To assess the model's training performance, we conducted a random split of the dataset, allocating 80% of the data to the training set and the remaining 20% to the test set, maintaining an 8:2 ratio.

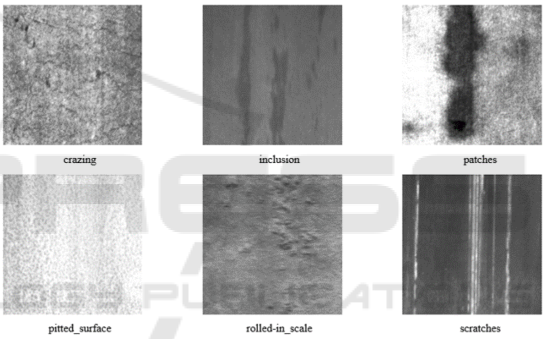

Figure 5. Examples of defect on the steel surface.

3.2 Implementation Details

The GPU utilized was an NVIDIA GeForce RTX 2080 Ti graphics card with 11GB of memory. The optimization algorithm employed was Stochastic Gradient Descent (SGD), with a batch size of 16. The initial learning rate was set to 0.01, momentum at 0.937, weight decay factor of 0.005, and the training process was conducted over 300 epochs. Our method was implemented with Pytorch. The input image size was consistently scaled to 224×224 pixels.

3.3 Evaluation Metric

We use average precision (AP), mean average precision (mAP), and frames per second (FPS) as the model evaluation metrics with the following formulas:

$$AP = \int_0^1 p(R)d(R) \tag{7}$$

$$p = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP(i) \tag{10}$$

where AP denotes precision of a single category. mAP denotes the mean of the average precision of all target detection categories. TP, FP, and FN denote the number of true positives, false positives, and false negatives, respectively.

3.4 Ablation Study

In order to assess the effectiveness of various improvement strategies, we conducted ablation experiments, and the outcomes are presented in Table 1.

Table 1. Results of ablation experiment.

Method	DsSE_C3	Ms_PD Head	SIoU	mAP @0.5/%	FPS/ Frame·s^{-1}	AP/%					
						Cr	In	Pa	PS	RS	Sc
1	—	—	—	77.2	93.4	47.6	81.6	96.1	80.3	63.3	94.2
2	√	—	—	78.9	82.9	52.3	82.6	96.6	81.6	65.5	94.6
3	√	√	—	80.0	79.6	51.4	85.2	96.3	81.8	69.6	95.5
4	√	√	√	80.6	79.3	53.2	87.2	95.7	83.2	67.9	96.3

The data presented in Table 1 clearly illustrates the effectiveness of the proposed DsSE_C3 module in enhancing the accuracy of strip steel defect detection. When compared to the original Method 1, the improved Model 2 exhibits a notable increase of 1.7% in mAP. Furthermore, the incorporation of the multi-scale pyramidal detection head results in a substantial 2.6% improvement in the AP for the "In" defect class in Method 3, specifically benefiting the detection of smaller targets. Finally, with the introduction of the SIoU loss function, Method 4 attains the highest detection accuracy at 80.6%. Apart from the "Pa" and "RS" classes, all other defect classes exhibit improved AP values when compared to Method 3. This underscores the efficacy of the SIoU loss function in enhancing the overall detection accuracy across different defect categories. Among them, for the Cr class, which has the lowest detection accuracy and is more difficult to detect, the detection accuracy is improved from 47.6% to 53.2%, which has a 5.6 percentage points improvement, enhancement effect is outstanding.

3.5 Comparison with State-of-the-Art Methods

To validate the efficacy of the algorithms introduced in this paper, a series of comparative experiments have been conducted. To ensure the fairness of the comparison experiments, the following classical algorithms: Cascade R-CNN, Faster R-CNN, and Retinanet are all trained in the same experimental environment and MMDetection0 open target detection toolbox. The experimental results are shown in Table 2.

Table 2. Performance comparison of mainstream algorithms.

Method	mAP/%	FPS/ Frame·s^{-1}	AP/%					
			Cr	In	Pa	Ps	RS	Sc
YOLOv3	69.1	55	44.7	60.8	84.4	74.5	61.1	87.2
YOLOv4	69.1	—	35.1	77.6	90.2	78.4	51.2	82.2
SSD	72.82	—	36.3	81.9	91.3	83.9	62.1	78.2
Faster R-CNN	78.7	21.7	49.5	84.8	93.7	78.1	71.1	94.9
Retinanet	75.1	32.3	49.6	78.4	94.7	78.7	70.3	78.7
Cascade R-CNN	79.4	18.6	47.3	86.2	92.7	83.0	73.4	94.0
YOLOv5s	77.8	92.4	46.6	84.8	96.1	80.3	63.8	95.1
Ours	80.6	79.3	53.2	87.2	95.7	83.2	67.9	96.3

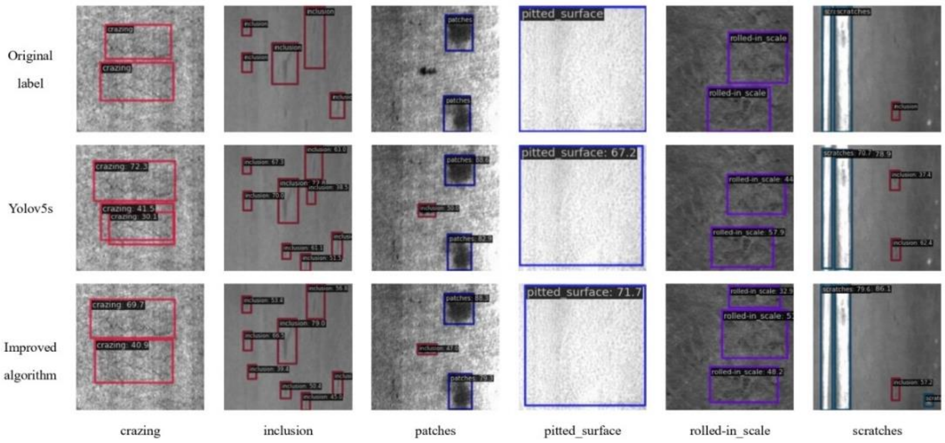

Figure 6. Comparison of detection effects.

Fig.6. illustrates a side-by-side comparison of the detection results for samples in each defect category. The results of crazing defect detection clearly demonstrate that the algorithm presented in this paper excels in accurately locating the detection bounding box when compared to YOLOv5s. In the case of inclusion defect detection, it is evident that our improved algorithm surpasses the baseline model, offering more robust detection capabilities and outstanding performance for small targets.

4 CONCLUSION

To tackle the challenges associated with detecting surface defects on strip steel within the context of automated production, this paper introduces an enhancement strategy based on the YOLOv5s algorithm. Firstly, we propose an enhanced C3 module aimed at augmenting the model's feature extraction capabilities. Secondly, we incorporate a multi-scale pyramidal detection head to bolster the model's proficiency in detecting small targets. Lastly, we adopt the SIoU loss function to expedite the model's convergence speed during training, thereby improving its overall performance in defect detection. The experimental results show that compared with the benchmark model, the method proposed in this paper effectively improves the detection accuracy and the detection effect for small targets is improved significantly. Furthermore, we plan to utilize techniques like pruning and distillation to reduce the model's size, facilitating its deployment on embedded edge devices, thereby expanding its practical applicability.

ACKNOWLEDGMENTS

This work was financially supported by Fujian Provincial Natural Science Foundation Project (2022J011247).

REFERENCES

Lei Yang, Shuai Xu, Junfeng Fan, En Li, Yanhong Liu. A pixel-level deep segmentation network for automatic defect detection [J].*Expert Systems with Applications*, 2023, 215: 119388. https://doi.org/10.1016/j.eswa.2022.119388

Fityanul Akhyar, Ying Liu, Chao-Yung Hsu, Timothy K. Shih, Chih-Yang Lin. FDD: a deep learning–based steel defect detectors [J].*The International Journal of Advanced Manufacturing Technology*, 2023: 1-15. https://doi.org/10.1007/s00170-023-11087-9

Jianbo Yu, Xun Cheng, Qingfeng Li. Surface defect detection of steel strips based on anchor-free network with channel attention and bidirectional feature fusion[J]. *IEEE Transactions on Instrumentation and Measurement,* 2021, 71: 1-10. https://doi.org/10.1109/TIM.2021.3136183

Bo Tang, Zi-Kai Song, Wei Sun, Xing-Dong Wang. An end-to-end steel surface defect detection approach via Swin transformer [J].*IET Image Processing*, 2023, 17(5): 1334-1345. https://doi.org/10.1049/ipr2.12715

Muwei Jian, Haodong Jin, Xiangyu Liu, Linsong Zhang. Multiscale Cascaded Attention Network for Saliency Detection Based on ResNet [J].*Sensors*, 2022, 22(24): 9950. https://doi.org/10.3390/s22249950

Chao Zhao, Xin Shu, Xi Yan, Xin Zuo, Feng Zhu. RDD-YOLO: A modified YOLO for detection of steel surface defects [J]. *Measurement*, 2023: 112776. https://doi.org/10.1016/j.measurement.2023.112776

Marcelo Gennari, Roger Fawcett, Victor Adrian Prisacariu. DSConv: Efficient Convolution Operator [J]. *arXiv preprint arXiv*:1901.01928, 2019.https://doi.org/10.48550/arXiv.1901.01928

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141. https://doi.org/10.48550/arXiv.1709.01507

Chengqi Lyu, Wenwei Zhang, Haian Huang, et al. RTMDet: An Empirical Study of Designing Real-Time Object Detectors [J]. *arXiv preprint arXiv*:2212.07784, 2022.https://doi.org/10.48550/arXiv.2212.07784

Ionut Cosmin Duta, Li Liu, Fan Zhu, Ling Shao. Pyramidl convolution: Rethinking convolutional neural networks for visual recognition [J]. *arXiv preprint arXiv*:2006.11538, 2020. https://doi.org/10.48550/arXiv.2006.11538

Sasha Targ, Diogo Almeida, Kevin Lyman. Resnet in resnet: Generalizing residual architectures [J]. *arXiv preprint arXiv*:1603.08029, 2016. https://doi.org/10.48550/arXiv.1603.08029

Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, Dongwei Ren. Distance-IoU loss: Faster and better learning for bounding box regression[C]*//Proceedings of the AAAI conference on artificial intelligence.* 2020, 34(07): 12993-13000. https://doi.org/10.48550/arXiv.1911.08287

Zhora Gevorgyan. SIoU loss: More powerful learning for bounding box regression [J]. *arXiv preprint arXiv:*2205.12740, 2022. https://doi.org/10.48550/arXiv.2205.12740

Kai Chen, Jiaqi Wang, Jiangmiao Pang, et al. MMDetection: Open MMLab Detection Toolbox and Benchmark [J]. *arXiv*, 2019. https://doi.org/10.48550/arXiv.1906.07155