Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 6, No. 3, 2023

Study on the Lightweighting of YOLOv5s Model for Precise Detection of Irregular-shaped Components

Hanpeng Ren [1], Lei Dong [1, *], Yangang Jin [2], Yuefei Zheng [1], Haojie Zhu [1], Beiping Zhao [2], Yushuang Feng [1]

[1] School of Mechanical Engineering, Tianjin University of Technology and Education, Tianjin 300222, China
[2] Citylong Technology Development Co., Ltd, Tianjin 300300, China

*** Corresponding author:** Lei Dong

Abstract: Traditional methods for target recognition face challenges in meeting both precision and speed requirements for precision-shaped parts. In this paper, we propose an enhanced algorithm for precision-shaped part recognition by integrating deep learning theory. To achieve this, we modify the YOLOv5 network. Specifically, we replace the C3 module of the original network's backbone with the C3_ghostnetv2 network, which incorporates Ghostnetv2. This modification results in a lighter network with reduced model parameters and size, thereby improving detection speed. Moreover, we replace the convolution in the original network's neck with Deformable Convolution v2 (DCNv2) to enhance feature extraction for precision-shaped parts. We conduct comparative experiments on a self-made dataset of precision-shaped parts. The experimental results demonstrate that our improved algorithm reduces parameters by 13.7% and model size by 12.5% compared to the original YOLOv5s algorithm, while achieving a 1.4% increase in detection accuracy. The proposed algorithm accurately identifies and classifies precision-shaped machining parts, providing valuable technical support for subsequent intelligent production.

Keywords: Target Identification; YOLOv5; Lightweighting; Irregular-shaped Parts; DCNv2.

1. Introduction

Many high-precision equipment companies specialize in customized products due to rapid technological advancements. These products exhibit diverse varieties, small batch sizes, numerous components, complex production processes, and the integration of production and research and development aspects. These characteristics heighten production challenges and significantly influence the stability, continuity, and efficiency of the manufacturing system. Furthermore, the need for higher production efficiency and quality has prompted increased demands on prevailing production methods.

Currently, component identification methods mainly consist of traditional machine methods and deep convolutional neural network object detection methods. Template matching is a classic method for identification and localization among various object recognition methods [1][2][3]. However, industrial component recognition algorithms based on features and fused features focus on the characteristics of the target to be tested. They have high requirements for the size and angle of the component. These algorithms cannot meet the real-time and accurate identification and positioning requirements of precise and complex-shaped components in relatively complex environments. Furthermore, in the process of detecting images, the image undergoes preprocessing before template matching is performed, which leads to poor real-time performance and limited detection target types. The above methods usually encounter difficulties in extracting feature information and recognition for components with complex structures and similar appearances. Object detection algorithms based on convolutional neural networks are superior to traditional machine vision technology in terms of detection and recognition due to the continuous development of modern information technology. However, in industrial applications, utilizing a large number of high-performance computers is not the most economically effective approach. Hence, there is a need for more efficient component recognition methods to enhance industrial inspection and assembly processes.

This paper introduces GD-YOLO, an improved YOLOv5s algorithm, as a precise detection and recognition method for complex-shaped parts. It enables fast detection and recognition of small parts with complex shapes in aerospace batch production. The paper's main contributions are summarized as:

1. Replacing the C3 module of the original network model's backbone network with C3_Ghostnet, which includes Ghost, both reduces the complexity of computation and network structure while maintaining sufficient accuracy.
2. The replacement of convolution in the C3 module of the neck network part of the original network model with DCNv2, named C3_DCNv2, enhances feature extraction to accommodate features of varying sizes and shapes, thereby improving detection accuracy.

2. Improve the YOLO Algorithm

The primary focus of this article is to improve the original algorithm model based on YOLOv5s. The Ghostnetv2 and C3 modules are combined in the Backbone network, resulting in the C3_ghostnetv2 module. This module reduces the network's weight, decreases the number of network parameters, and enhances detection speed. The deformable convolution DCNv2 replaces the convolution block in the C3 module of the Neck network. This enhancement improves the model's capability to extract features from irregular parts, resulting in improved detection accuracy. The upgraded

network model is depicted in Figure 1 below.

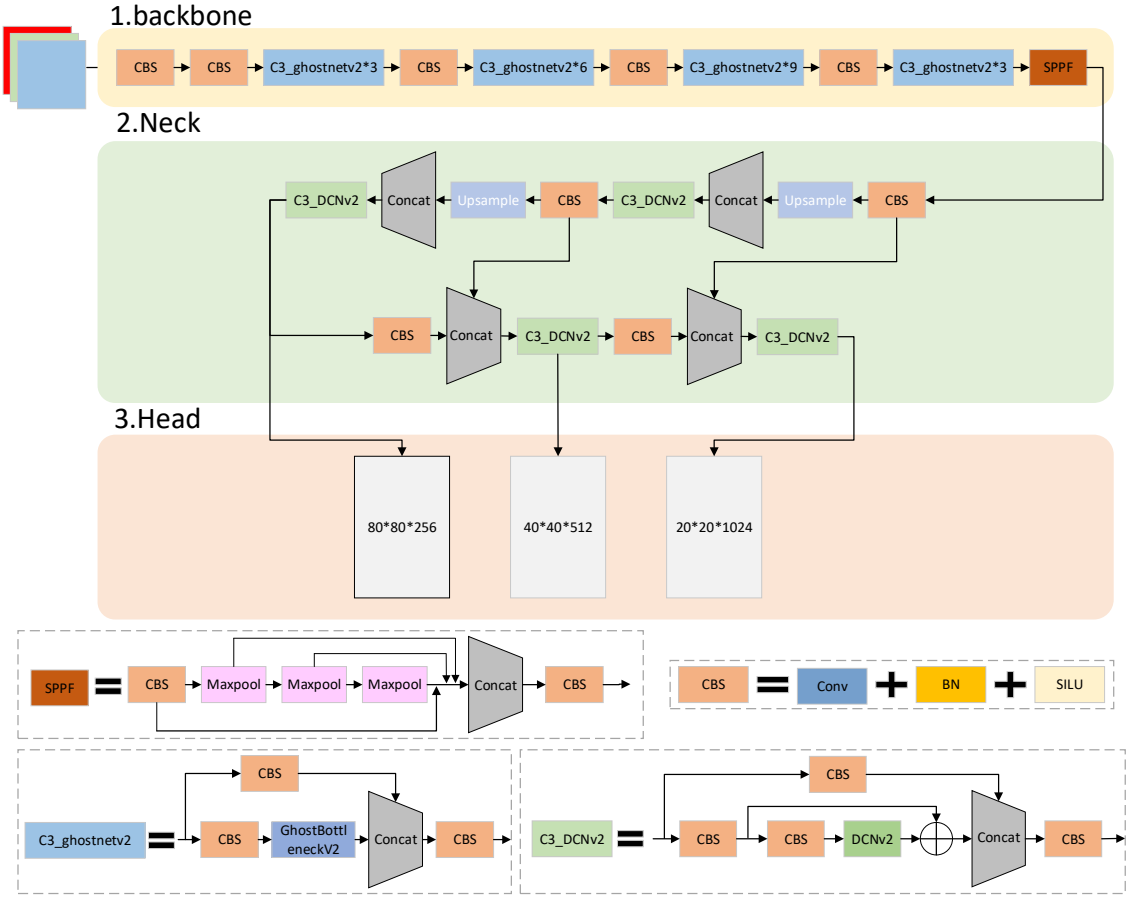

Figure 1. The YOLOv5s network has been enhanced

2.1. C3_ghostnetv2

Because of the constraints inherent in current embedded devices, applying large models to actual industrial production is challenging. Networks like Mobile Net and Shuffle Net employ depth wise separable convolution or shuffle operations to decrease model parameters and computational complexity. However, these networks still employ numerous dense 1x1 convolution operations for transforming dimensions or fusing information, which can use substantial computational resources. In contrast, the Ghost module initiates feature generation via ordinary convolution and subsequently augments the features and channels using simple linear operations. Ultimately, the augmented features are joined with the initial convolution-generated features to form the output, resulting in reduced model parameters and improved detection speed. The formula for the initial set of generated feature maps is as follows:

$$Y = X * f \tag{1}$$

$X \in \mathbb{R}^{c\times h\times w}$ represents the input data, c represents the number of input channels, and h and w represent the height and width of the input data. $Y \in \mathbb{R}^{h'\times w'\times n}$ represents the output data, with h' and w' representing the height and width, and n representing the number of channels. The convolution filter is denoted by $f \in \mathbb{R}^{c\times k\times k\times n}$, and k represents the size of the kernel. The generated set of feature maps undergoes a simple linear operation, with the following formula.

$$y_{ij} = \Phi_{ij}(y_i), \quad \forall i = 1,...,n, \ j=1,...,m \tag{2}$$

y_i represents the I intrinsic feature map in Y, and Φ_{ij} is utilized for generating the I Ghost feature map. The detailed process of generating Ghost feature maps is illustrated in Figure 2.

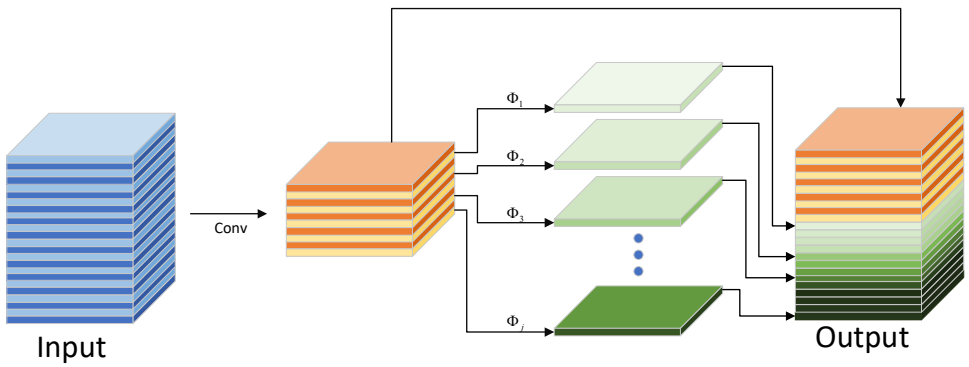

Figure 2. Diagram depicting the process of obtaining Ghost Features

Due to the limited ability of the Ghost network model in handling long-range dependencies, this paper proposes the adoption of Ghostnetv2 for network lightweighting. Ghostnetv2 incorporates the DFC attention module, which addresses the lack of long-range dependency in the original Ghost. To illustrate the structure of the GhostBottleneckV2, which consists of Ghostnetv2 modules, please refer to Figure 3.

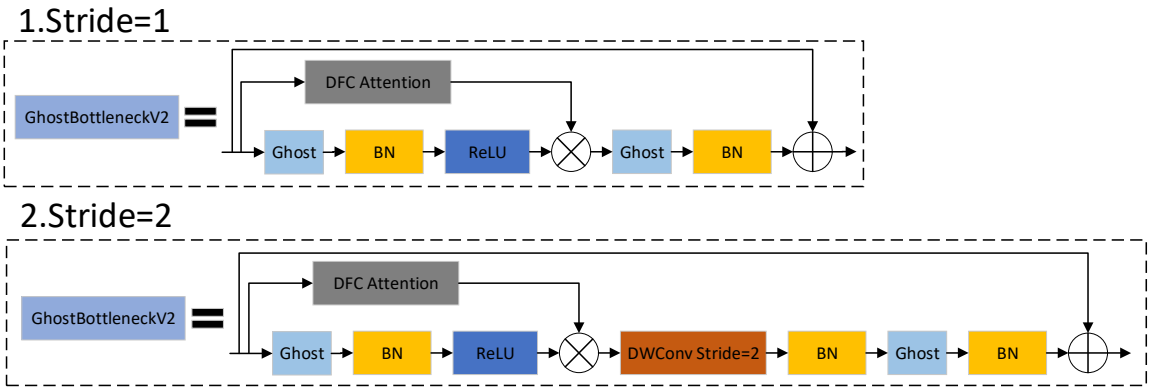

Figure 3. Ghost Bottleneck V structure diagram

The structure of Ghost Bottleneck V varies depending on whether the stride is set to 1 or 2. For a stride of 1, the structure primarily consists of the Ghost module, Batch Normalization (BN) module, Re LU function, and DFC attention module. Conversely, for a stride of 2, the structure incorporates an extra Depth wise Separable Conv (DW Conv) module and a BN module for normalization after DW Conv. n this paper, we merge the Ghost Bottleneck V with the C3 module of the original YOLOv5 network Backbone, resulting in C3_ghostnetv2.

2.2. C3_DCNv2

In the detection of precise irregular-shaped parts, the parts exhibit a variety of shapes and irregularities, which presents challenges in extracting target features and leads to a decrease in detection accuracy of the target parts. To address this issue, we utilize the deformable convolution DCNv2. In comparison to the previous generation DCNv1, DCNv2 not only maintains the offset of DCNv1 to accommodate various shapes and sizes of detection targets but also introduces point weights to mitigate the problem of DCNv1 easily sampling outside the target when performing offsets, assigning lower weights to sampled points outside the target. This helps reduce the influence of these sampled points on the output. The calculation formula is presented below:

$$y(p)=\sum_{k=1}^{k} w_k \bullet x(p+p_k+\Delta p_k)\bullet \Delta m_k \tag{3}$$

The weight coefficient, Δm_k, corresponds to the Kth position, $\Delta m_k \in (0,1)$. W_k represents the weight of P_k. P denotes the central point of the sampled data point. p_k indicates the relative position of the central point within the sampled data point. Δp_k represents the position offset and is of the floating-point data type.

In this article, the DCNv2 module is introduced to replace the convolution in the C3 module of the original YOLOV5 network, which is denoted as C3_DCNv2. The inclusion of the DCNv2 module in the C3 module enhances the capability of feature extraction for irregular components, leading to improved detection accuracy. Figure 4 depicts the structural diagram of C3_DCNv2.

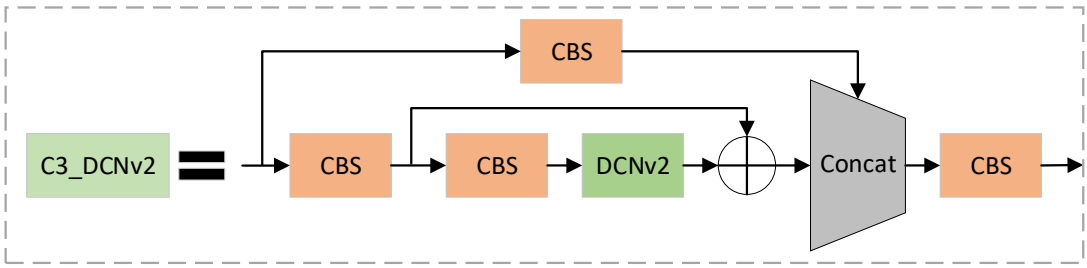

Figure 4. C3_DCNv2 architecture diagram

3. Experiment and Results

3.1. Experimental Dataset and Experimental Environment

To improve the model's generalization ability and assess the performance of the enhanced object detection algorithm, a precision irregular parts dataset created in-house is utilized. Five categories of processed parts, including Vibrator, Vertical Tower, Connector A, Connector B, and Connector, are selected for data collection. The dataset is taken under various conditions, including different environments, backgrounds, and occlusions. There are 1207 original photos with a resolution of 3000×4000 pixels. Additionally, to enhance the model's training speed, a script adjusts the resolution of the images to 640×640. The data is manually processed and annotated using LabelMe annotation software, yielding the final dataset of precision irregular parts. Figure 5 displays sample images from this dataset.

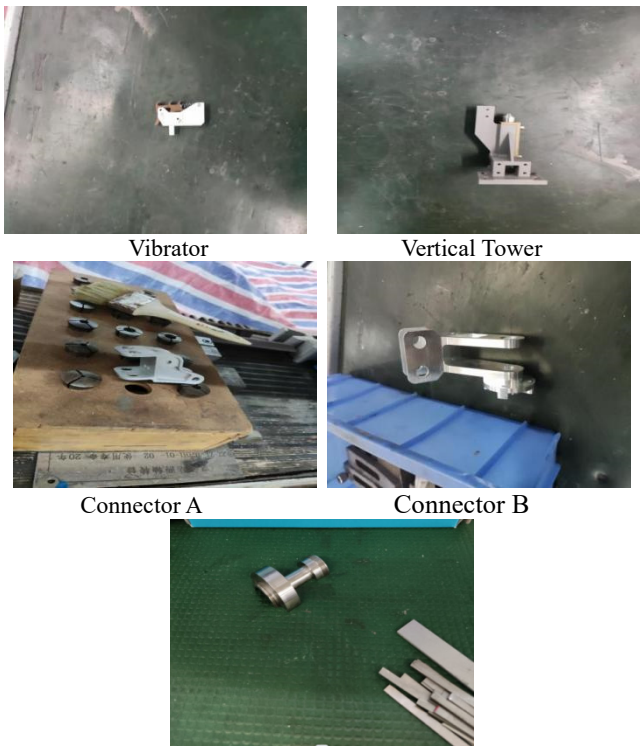

Figure 5. Precision custom-shaped part drawing

The experimental setup for this study consists of: Intel(R) Xeon(R) Platinum 8350C CPU @ 2.60GHz with 16 vCPUs, 42GB RAM, NVIDIA Ge Force RTX 3090 GPU with 24GB VRAM, Ubuntu 20.04 (64-bit), P y Torch framework version 1.10.0, CUDA version 11.3.

The YOLOv5s model is utilized as the fundamental training model for precision-shaped parts. The training set is enhanced using Mosaic data augmentation, employing an input size of 640×640 pixels. The training process includes 250 iterations, each with a batch size of 32 samples. The number of data loading threads is set to 4.

3.2. Evaluation Criteria

The evaluation of a model's performance primarily relies on two factors: detection accuracy and speed. Detection accuracy is measured using Mean Average Precision (Map). During the process of multiple object detection, a Precision-Recall (P-R) curve is generated for each object based on the recall rate and precision rate. AP represents the area under the curve, while Map is the average of AP values across multiple categories. The calculation method is depicted in the equation below:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$mAP = \frac{\sum AP}{N(Class)} \tag{6}$$

In this context, TP (True Positive) and TN (True Negative) denote the samples that the model accurately categorized as positive and negative, respectively. Conversely, FP (False Positive) and FN (False Negative) represent the samples that the model erroneously classified as positive and negative. Average Precision (AP) refers to the mean precision attained by averaging the precision scores of these samples.

This study measures speed using three key indicators: parameters, GFLOPs (floating point operations per second), and FPS (Frames Per Second). The parameter count is primarily determined by the model's structure, while GFLOPs can assess the model's complexity. A higher GFLOPs value indicates a higher number of floating-point operations required by the network, which may result in slower speed under similar hardware conditions. FPS (Frames Per Second) is a term commonly used in image processing, representing the number of frames transmitted per second. In the context of object recognition, it signifies the model's ability to recognize frames within a one-second interval. A higher FPS value indicates a faster speed of algorithm recognition.

3.3. Experimental Results

3.3.1. Model Performance Ablation Experiment

Revised sentence structure and phrasing. Simplified sentence structure and clarified the sequence of events. Restructured sentence and added clarity by specifying the purpose of comparative experiments. Revised sentence structure and phrasing. Restructured sentence and clarified the purpose of the notation "√". Reworded sentence to improve clarity and consistency.

Table 1. Ablation experiment

Network model	C3_Ghostnetv2	C3_DCNv2	Map@.5%	Rcall	Params (×106)	FPS(f/s)	Size(M)
a			91	0.89	7.03	138.89	13.7
b	√		90.9	0.87.7	5.92	142.86	11.6
c	√	√	92.4	0.89.8	6.07	111.21	12

The experimental data utilizes the YOLOv5 algorithm as the baseline. According to the experimental results in Table 1, when the C3_Ghostnetv2 is incorporated into network model b, the network parameters decrease by 15%, while increasing the network's fps by 4f/s. The mAP_0.5 shows a marginal decrease of only 0.1%. this suggests that the addition of C3_Ghostnetv2 to the network's backbone and obtaining sufficient feature maps through simple linear operations can reduce the number of parameters and improve computational speed, without compromising detection accuracy. When comparing model C with model b in the network, it is observed that incorporating C3_DCNv2 into the network's Neck simplifies the acquisition of features from targets of varying sizes and shapes, ultimately enhancing network detection accuracy. The network's mAP@0.5% increases to 92.4%, accompanied by a notable improvement in the recall rate; however, the model's detection speed decreases to 111.21f/s. Compared to the original model, the network's mAP_0.5% exhibits an improvement of 1.4%. Additionally, the recall rate increases from 0.89 to 0.898, while the number of parameters decreases to 6.07 ($\times 10^6$). Despite a decrease in detection speed by 27f/s, the model's performance remains suitable for real-time detection. In conclusion, the enhanced model (model C) surpasses the original YOLOv5 model in mAP, recall, and model size.

3.3.2. Algorithm Comparison Experiment

To further validate the effectiveness of the improved algorithm model, a comparative experiment was conducted using the dataset, comparing the algorithm model proposed in this paper with the current mainstream algorithm models. The main comparative algorithms included YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, YOLOv6, YOLOv6-tiny, and YOLOv7-tiny, as shown in Table 7. The main compared algorithms were YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, YOLOv6, YOLOv6-tiny, and YOLOv7-tiny. Table 2 shows the results.

Table 2. Comparative experiment

Method	Map_0.5/%	Fps(f/s)	Params(×106)	Size/M
YOLOv5n	86.6	136.99	1.77	3.7
YOLOv5s	91	138.89	7.02	13.7
YOLOv5m	91.4	107.53	20.89	40.2
YOLOv5l	91.7	83.33	46.13	88.5
YOLOv5x	91	60.6	86.24	165.1
YOLOv7-tiny	91.3	60.61	3.02	11.7
Ours	92.4	111.11	6.07	12

Based on the results in the comparative table, the algorithm proposed in this study exhibits the highest level of detection accuracy when compared to other commonly used detection models, while also maintaining a certain level of detection speed.

The enhanced algorithm, Map_0.5/%, compared to the original YOLOv5s algorithm, demonstrates a 1.4% enhancement in detection accuracy for the model. It achieves a 5.8% enhancement compared to YOLOv5n, a 0.7% enhancement compared to YOLOv5l, a 1% enhancement compared to YOLOv5m, and a 1.4% enhancement compared to YOLOv5x. After network improvement, the model's detection speed in FPS has experienced a slight decrease due to the addition of C3_DCNv2, resulting in increased post-processing time and a minor decrease in detection speed. However, it still satisfies the real-time detection requirements. Both the enhanced algorithm model and YOLOv7-tiny have achieved a 1.1% increase in detection accuracy, and the network parameters and detection speed have also shown significant improvement. In summary, the proposed GD-YOLO algorithm exhibits strong real-time performance in datasets containing precision-shaped parts, providing valuable insights for detecting and identifying precision-shaped parts as well as other industrial components. This validates the feasibility and superiority of the algorithm proposed in this paper.

4. Conclusion

This paper proposes improvements to the YOLO algorithm to overcome the limitations of traditional object detection methods in handling precision irregular parts. These improvements enable the application of the YOLO algorithm in aerospace precision parts inspection, enabling real-time detection of such parts. A lightweight feature extraction network is employed. Modifications are made to the C3 module of the Neck network. Experiments are performed on a dataset of precision irregular parts to validate the proposed approach.

The experimental results demonstrate that the HT-YOLO algorithm enhances the detection accuracy of the original algorithm by 1.4%. The network achieves a detection speed of 111.11f/s, meeting real-time detection requirements without compromising detection accuracy. Additionally, the algorithm demonstrates its effectiveness and superiority in precisely detecting non-standard parts due to its robustness in various dataset environments. Future research can explore network pruning and lightweight methods to further reduce network complexity.

Acknowledgments

This research was funded by the Development of special structure technology and simulation technology of high precision servo motor for aerospace application [Project No. 2022ZD027], as well as the Tianjin Research Innovation Project for Postgraduate Students [Project No. 2022 SKYZ 294].

References

[1] Zhang, Y., Liang, J., Lu, Q., et al. (2022). A Novel Efficient Convolutional Neural Algorithm for Multi-Category Aliasing Hardware Recognition. Sensors, 22(14), 5358.

[2] Girs hick, R., Donahue, J., Darrell, T., et al. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 580-587)

[3] Gir s hick, R. (2015). Fast R-CNN. In International Conference on Computer Vision (pp. 1440-1448).

[4] Ren, S., He, K., Gir s hick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In Neural Information Processing Systems (pp. 91-99)

[5] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., & Reed, S. (2016). SSD: Single Shot Multi Box Detector. In European Conference on Computer Vision (pp. 21-37). Springer International Publishing.

[6] Redmon, J., Div v ala, S., Gir s hick, R., et al. (2016). You Only Look Once: Unified, Real-Time Object Detection. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 779-788)

[7] Liu, W., Anguelov, D., Erhan, D., et al. (2016). SSD: Single Shot Multi Box Detector. In European Conference on Computer Vision (pp. 21-37). Springer International Publishing.

[8] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 6517-6525).

[9] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-6).

[10] He, K., Zhang, X., Ren, S., et al. (2015). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904-1916.

[11] Lin, T., Dollar, P., Gir s hick, R., et al. (2017). Feature Pyramid Networks for Object Detection. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 936-944).

[12] Liu, S., Qi, L., Qin, H. F., et al. (2018). Path aggregation network for instance segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8759-8768). Salt Lake City: IEEE.

[13] Howard, A. G., Zhu, M., Chen, B., et al. (2017). Mobile nets: Efficient convolutional neural networks for mobile vision applications. a r Xi v preprint arXiv:1704.04861.

[14] Sandler, M., Howard, A., Zhu, M., et al. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4510-4520).

[15] Howard, A., Sandler, M., Chu, G., et al. (2019). Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 1314-1324).

[16] Zhang, X., Zhou, X., Lin, M., et al. (2018). Shu ff le net: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6848-6856).

[17] Ma, N., Zhang, X., Zheng, H. T., et al. (2018). Shu ff le net v2: Practical guidelines for efficient c architecture design. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 116-131).

[18] Han, K., Wang, Y., Tian, Q., et al. (2020). Ghost net: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1580-1589).

[19] Liu, Y., Shao, Z., Hoffmann, N. (2021). Global attention mechanism: Retain information to enhance channel-spatial interactions. a r Xi v preprint arXiv:2112.05561.

[20] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132-7141).

[21] Woo, S., Park, J., Lee, J. Y., et al. (2018). CBAM: Convolutional Block Attention Module. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19).

[22] Zheng, Z., Wang, P., Liu, W., et al. (2020). Distance-Io U loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 07, pp. 12993-13000).

[23] Gevorgyan, Z. (2022). Sio U loss: More powerful learning for bounding box regression. A r X iv preprint arXiv:2205.12740.