2023 IEEE International Conference on Image Processing (ICIP) | 978-1-7281-9835-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIP49359.2023.10222202

TP-YOLO: A LIGHTWEIGHT ATTENTION-BASED ARCHITECTURE FOR TINY PEST DETECTION

Yang Di[*] *Son Lam Phung*[*] *Julian van den Berg*[†] *Jason Clissold*[†] *Abdesselam Bouzerdoum*[*‡]

[*]University of Wollongong [†]Intelligent System Design [‡]Hamad Bin Khalifa University

ABSTRACT

Automatic detection of agricultural pests is a challenging problem that is of great interest in biosecurity and precision agriculture. The detection model must cope well with the dense distribution of small-sized pests in complex backgrounds. This paper proposes a lightweight attention-based network, called TP-YOLO, for tiny pest detection. We introduce two attention-based components, namely Contextual Transformer and Omni-Dimensional Dynamic Convolution modules, to enhance feature extraction. The proposed modules are integrated into the YOLOv8 backbone, a state-of-the-art baseline for object detection. This paper also introduces a new benchmark dataset consisting of 1,600 images of Khapra beetles for objective evaluation of pest detection algorithms. Extensive experiments on two datasets indicate that TP-YOLO achieves competitive detection accuracy while having a significantly smaller model size and fast prediction time. We have made the code available to the public at: `https://github.com/yangdi-cv/TP-YOLO`.

Index Terms— Pest detection, attention mechanism, CNN, YOLO, vision transformers.

1. INTRODUCTION

Detection of pests such as insects, mites, and rodents is crucial to agricultural production because pests not only reduce yield but also damage the environment and natural resources. Effective pest detection methods enable farmers to locate harmful pests and prevent crop damage in a timely manner. Most current approaches rely on manual inspection [1], pheromone traps [2], and plant and soil analysis [3]. These methods are simple, but they are slow and labour-intensive.

With the advances in deep learning, various studies have been conducted to develop automated systems for detecting pests and insects in agriculture [4]. Such detection models can be categorized into two main types: two-stage detectors [5, 6] and one-stage detectors [7, 8]. For example, Li *et al.* employed two-stage methods, Faster R-CNN and Mask R-CNN, trained on the IP102 benchmark to detect 10 categories of insect pests [5]. However, the intensive computational cost limits these methods in real-time applications where prediction time is critical. To overcome this problem, one-stage pest detection methods have been proposed recently. They are mainly extended from the YOLOv4 [9] and YOLOv5 [10] algorithms. For example, Hu *et al.* utilized YOLOv5 to detect pests in cabbage fields using near-infrared images [7]. Although these one-stage methods yield promising detection performance, their large model sizes are not well suited for deployment on low-memory edge computers.

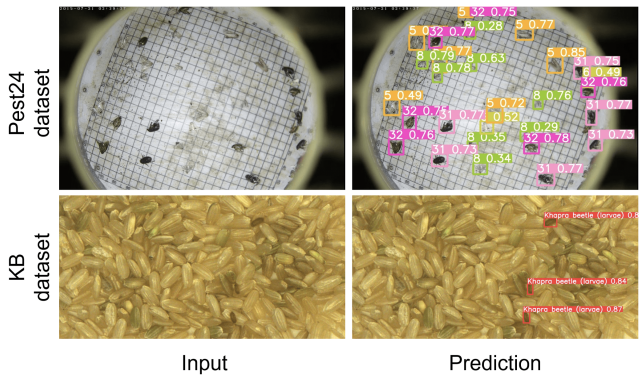

Fig. 1: Visual results of the proposed method for tiny pests detection. Row 1: Pest24 dataset; Row 2: Khapra beetle dataset. Column 1: Input images; Column 2: Detection outputs.

Recently, a few benchmark datasets have been introduced for pest detection research, but they include only some categories of field crop insects. Furthermore, the captured images contain large-sized insects, which may not reflect real-world variations. In this paper, we create a new benchmark dataset for tiny pest detection, which contains three Khapra beetle categories: adults, larvae, and skin. Khapra beetle (Trogoderma granarium) is a highly destructive pest for stored grains, including rice, wheat, barley, oats, and corn. It is considered one of the world's worst invasive species in terms of global economic losses [11].

This paper has two main contributions. *First*, we propose a lightweight attention-based method, called TP-YOLO for real-time detection of tiny pests. Our TP-YOLO outperforms the state-of-the-art (SOTA) methods in terms of detection accuracy and inference time. Fig. 1 shows the visual results of pest detection. *Second*, we introduce a new Khapra Beetle (KB) dataset for pest detection research.

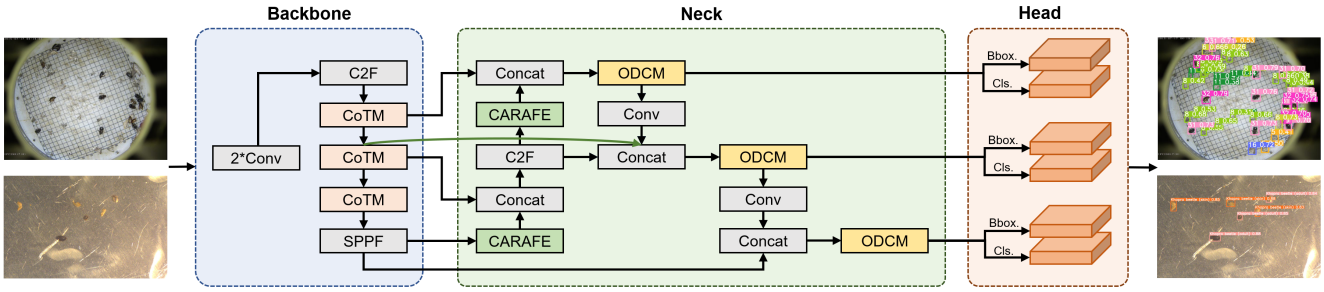

Fig. 2: Network architecture of TP-YOLO for pest detection.

2. PROPOSED METHOD

This section presents the proposed method TP-YOLO, including the network architecture (Section 2.1), the Contextual Transformer Module (Section 2.2), the Omni-Dimensional Dynamic Convolution Module (Section 2.3), and model training (Section 2.4).

2.1. Network architecture

The proposed TP-YOLO is illustrated in Fig. 2. For the backbone, we use two consecutive 3×3 convolutions followed by down-sampling. Inspired by YOLOv8, we employ the faster Cross Stage Partial Bottleneck with 2 Convolutions (C2F) block for feature extraction. Fig. 3 shows the structure of the C2F block. This block applies the cross-stage connection and feature reuse to provide rich gradient flow information and improve the network's efficiency. Meanwhile, the Bottleneck structure in C2F block uses the residual connection to maintain the original performance and reduces the number of parameters. We apply the Contextual Transformer (CoT) [12] in the backbone to better combine contextual information with local information. The Spatial Pyramid Pooling Fast (SPPF) block [13] is then used to extract multi-scale features via max pooling and the *concat* operation (Fig. 4).

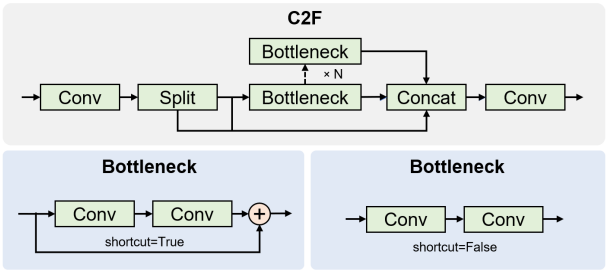

Fig. 3: The structure of the C2F block. The accumulation quantity N of Bottleneck blocks and the shortcut connections are adjustable.

For the neck, we apply the bottom-up fusion connection in Bidirectional Feature Pyramid Network (BiFPN) to preserve the underlying feature information [14]. The skip connection is added to fuse the low-level features with the high-level features. We apply the CARAFE operator [15] for feature upsampling. This operator employs spatial attention to improve the resolution of feature maps and dynamically adjusts the weights of different positions in the feature map to retain important information. Before each of the detection heads, we apply the Omni-Dimensional Dynamic Convolution (ODConv) [16] to enhance the ability of feature extraction for tiny objects. This method performs attention convolution operations from multiple kernel spatial dimensions, thereby effectively improving the detection performance.

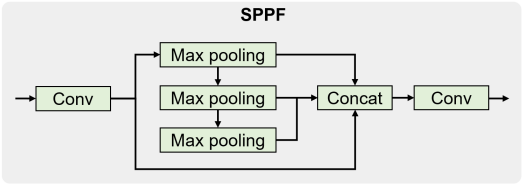

Fig. 4: The structure of the SPPF block.

For the heads, we use three scales of decouple-head [17, 13] to conduct object localization and classification separately. Decouple-head applies 'Conv2d + BatchNorm2d + SiLU' modules to improve the feature representation and one 1×1 Conv2d to adjust the number of channels.

2.2. Contextual Transformer Module

We propose Contextual Transformer Module (CoTM) based on CoT and C2F blocks. This module applies self-attention operations to obtain contextual semantic information. In the original Vision Transformer (ViT) [18], the attention matrix is produced by the dot product between keys and queries. However, this approach does not fully utilize the relationship between adjacent Keys. The CoT obtains the static context by encoding the key separately and fuses it with the dynamic context to improve the feature representation. Fig. 5 shows the structure of CoTM. Unlike ViT with the 1×1 convolution for encoding, CoT applies the $K \times K$ convolution to encode Key and extract local context K_1. Query and K_1 execute *concat* operation and perform two 1×1 convolution calculations. The obtained attention matrix is multiplied by the Value after the 1×1 convolution to acquire the dynamic context K_2. We

compute K_2 as follows:

$$K_2 = ([K_1, Q]W_\Theta W_\delta) \otimes V, \tag{1}$$

where K_1 denotes the local context extracted using the $K \times K$ convolution, Q denotes the Query calculated without convolution, and V denotes the Value obtained by the 1×1 convolution. Here, W_Θ and W_δ denote two 1×1 convolution calculation, and K_2 denotes the acquired global context information. Eventually, K_1 and K_2 are fused to produce feature maps.

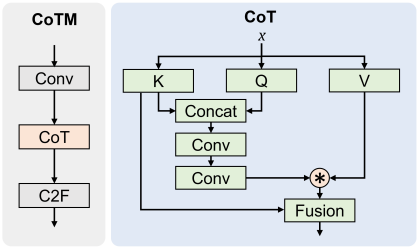

Fig. 5: The structure of the CoTM.

2.3. Omni-Dimensional Dynamic Convolution Module

We propose Omni-Dimensional Dynamic Convolution Module (ODCM) based on the ODConv method and C2F block. Traditional convolutional neural networks (CNN) use the same convolution kernels to learn over the entire dataset. To improve the prediction results, the CNN therefore needs to increase the width and depth of the network and adjust the parameters of the kernels. However, this will increase computational overhead and reduce inference speed. Dynamic convolution can apply multiple kernels and perform attention weighting to improve model performance. Moreover, the method can balance between the network size and the prediction speed.

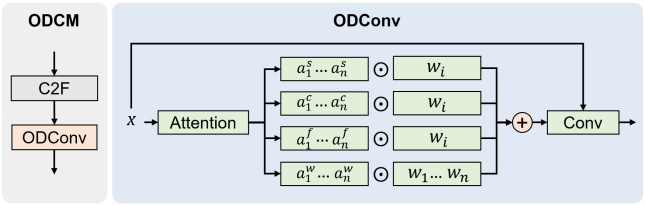

Fig. 6: The structure of the ODCM.

ODConv applies the attention mechanism for weighting operations from multiple spatial dimensions. Fig. 6 shows the structure of the ODCM. The Attention applied by ODConv consists of the global average pooling (GAP) layer and fully connected (FC) layer, followed by the ReLU, FC layer, and Softmax layer. The parameters included in the four dimensions addressed by ODConv are the number of convolution kernels n, the size of the spatial kernel $K \times K$, the number of input channels c_{in}, and the number of output channels c_{in}. The output of ODConv is computed as

$$y = \left(\sum_{i=1}^{n} a_i^{\mathrm{s}} \odot a_i^{\mathrm{c}} \odot a_i^{\mathrm{f}} \odot a_i^{\mathrm{w}} \odot W_i\right) * x, \tag{2}$$

where a_i^{s} denotes the attention weights on filters in the spatial domain, and a_i^{c} denotes the attention weights on the input channels of each convolutional filter. Here, a_i^{f} denotes the attention weights on the c_{out} filter, and a_i^{w} denotes the attention weights on the entire convolution kernel W_i. Symbol $\odot$ represents multiplication operations along different dimensions of the kernel.

2.4. Model Training

TP-YOLO applies the anchor alignment strategy in the Task-aligned One-stage Object Detection (TOOD) method to assign positive and negative samples for model training [19]. This strategy performs sample matching through the classification score and Intersection over Union (IoU) score of Anchor Box and Ground Truth (GT). The loss function of TP-YOLO includes a classification branch and a regression branch. The classification branch uses the Binary Cross Entropy (BCE) loss, and the regression branch uses a combination of the Distribution Focal (DF) loss and CIoU loss. The loss function is defined as

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\mathrm{BCE}} + \lambda_2 \mathcal{L}_{\mathrm{DF}} + \lambda_3 \mathcal{L}_{\mathrm{CIoU}}, \tag{3}$$

where $\mathcal{L}_{\mathrm{BCE}}$ denotes the BCE loss, $\mathcal{L}_{\mathrm{DF}}$ denotes the DF loss, $\mathcal{L}_{\mathrm{CIoU}}$ denotes CIoU loss, and λ_i represents the weights of each loss. Here, we set λ_1=0.5, λ_2=1.5, and λ_3=7.5.

3. EXPERIMENTS AND RESULTS

This section presents the experiments and the evaluations of the proposed method on two benchmark datasets. We also investigate the impact of the designed modules, including BiFPN, CARAFE, BoT, CoT, CoordAtt, and ODConv. Furthermore, we compare the proposed method with other SOTA networks, including Faster R-CNN, FCOS, DETR, YOLOv6, and YOLOv7. All experiments were conducted using an NVIDIA 4090 GPU with 24GB memory and the PyTorch framework. The Stochastic Gradient Descent (SGD) optimizer was used to train the models.

3.1. Datasets

In this paper, we use two pest datasets for model training and evaluation.

Khapra beetle dataset. We create a new dataset for Khapra beetle detection. This dataset contains 1,600 images and 4,885 bounding box annotations. We focus on three main categories: adult, larvae and skin. The images of this dataset are collected on six types of complex backgrounds, including

grain, carpet and metal. The size of Khapra beetle is tiny and around 3 mm in length. We use this dataset for the ablation study to develop a detection method for tiny pests. The dataset is divided with the ratios of 7:1:2 for model training, validation, and testing.

Pest24 dataset. Pest24 is a large-scale agricultural pest dataset [20]. This public dataset contains 25,378 images and 192,422 instance annotations. All images in this dataset are collected by an automated pest trap with a camera placed in the fields. The Pest24 dataset contains 24 categories of field crop pests. These pests are characterized by small sizes, similar shapes, and dense distributions. In Section 3.3, we apply the Pest24 dataset to further evaluate our TP-YOLO.

3.2. Ablation Study

We conduct an ablation study based on YOLOv8, which is the latest state-of-the-art real-time object detection algorithm [13]. We choose the nano version to conduct experiments on the Khapra beetle dataset. We first explore the effects of incorporating BiFPN and CARAFE to YOLOv8-N. Table 1 shows that BiFPN and CARAFE can effectively improve tiny-pest detection accuracy.

Table 1: Ablation study of the BiFPN structure and CARAFE operator on YOLOv8-N.

Method	AP	AP_{50}
Baseline YOLOv8-N method	75.8	97.9
+ BiFPN	77.0	98.4
+ BiFPN + CARAFE	77.2	98.5

Next, we introduce the attention mechanism modules to the improved YOLOv8-N. Here, we explore the effects of CoT and Bottleneck Transformer (BoT) in the backbone part [21]. Furthermore, we investigate the impacts of ODConv and Coordinate Attention (CoordAtt) on the neck part [22]. The results in Table 2 show that the combination of the CoT for the backbone and the ODConv for the neck achieves the best results.

Table 2: Ablation study of attention-based components on the improved YOLOv8-N.

Backbone part		*Neck part*		AP	AP_{50}
BoT	CoT	CoordAtt	ODConv		
✓		✓		77.4	98.9
✓			✓	78.0	98.6
	✓	✓		77.8	98.8
	✓		✓	78.0	98.9

The final network architecture for tiny pest detection that combines BiFPN, CARAFE, CoT and ODConv is referred to as TP-YOLO. By using the proposed attention-based method, TP-YOLO can better extract pest features from complex backgrounds. Compared with the baseline YOLOv8-N, TP-YOLO achieves a 2.2% improvement in the AP score, and a 1.0% improvement in the AP_{50} score.

3.3. Comparison of Detection Performances

We conduct experiments on the Khapra beetle dataset for other advanced object detection methods: Faster R-CNN, FCOS, DETR and YOLOv7. To comprehensively evaluate detection performances, we also record the inference speed, the model parameters, and the computation load. The experimental results in Table 3 show that the proposed TP-YOLO achieves the best detection performances in terms of accuracy, speed, model size, and computation load.

Table 3: Comparison of detection performances of different methods on the Khapra beetle dataset.

Method	*Higher is better*			*Lower is better*	
	AP	AP_{50}	FPS	Params	FLOPs
Faster R-CNN [23]	74.5	98.8	21.7	41.1M	91.0G
FCOS [24]	70.7	97.2	21.6	31.8M	78.7G
DETR [25]	63.9	98.7	22.7	41.3M	37.1G
YOLOv7 [26]	74.5	98.0	35.8	36.5M	103.2G
TP-YOLO [Ours]	78.0	98.9	80.0	4.3M	9.1G

We conduct further experiments on the Pest24 public dataset. Table 4 shows that our TP-YOLO network surpasses the performance of the SOTA real-time object detection algorithms YOLOv6 and YOLOv7. Compared with YOLOv6-S, which is 4.0 times larger, TP-YOLO achieves better and faster detection results.

Table 4: Comparison of TP-YOLO and other real-time object detection methods on the Pest24 dataset.

Method	*Higher is better*			*Lower is better*	
	AP	AP_{50}	FPS	Params	FLOPs
YOLOv6-S [17]	40.7	66.3	77.8	17.2M	44.1G
YOLOv7-tiny [26]	36.6	61.7	71.4	6.1M	13.4G
TP-YOLO [Ours]	42.0	66.8	81.3	4.3M	9.1G

4. CONCLUSION

In this paper, we propose a lightweight attention-based TP-YOLO method for tiny pest detection. We design CoTM to enhance the ability to extract network context information. In addition, we propose a dynamic convolution module to improve the detection performance for small objects. Experimental results show that our method outperforms the SOTA methods on two datasets and achieves the best real-time detection performance and the smallest model size. Our model can be deployed on edge devices for timely detection and control of pests in agricultural applications.

Acknowledgments: This work is funded by the Australian Research Council, Intelligent System Design, and the Australian Department of Agriculture, Fisheries and Forestry.

5. REFERENCES

[1] Vaclav Stejskal, Tomas Vendl, Zhihong Li, and Radek Aulicky, "Efficacy of visual evaluation of insect-damaged kernels of malting barley by sitophilus granarius from various observation perspectives," *Journal of Stored Products Research*, vol. 89, pp. 101711, 2020.

[2] Yu Sun, Xuanxin Liu, Mingshuai Yuan, Lili Ren, Jianxin Wang, and Zhibo Chen, "Automatic in-trap pest detection using deep learning for pheromone-based dendroctonus valens monitoring," *Biosystems Engineering*, vol. 176, pp. 140–150, 2018.

[3] Shaoqing Cui, Peter Ling, Heping Zhu, and Harold M. Keener, "Plant pest detection using an artificial nose system: a review," *Sensors*, vol. 18, no. 2, pp. 378, 2018.

[4] Wenyong Li, Tengfei Zheng, Zhankui Yang, Ming Li, Chuanheng Sun, and Xinting Yang, "Classification and detection of insects from field images using deep learning for smart pest management: A systematic review," *Ecological Informatics*, vol. 66, pp. 101460, 2021.

[5] Wei Li, Tengfei Zhu, Xiaoyu Li, Jianzhang Dong, and Jun Liu, "Recommending advanced deep learning models for efficient insect pest detection," *Agriculture*, vol. 12, no. 7, pp. 1065, 2022.

[6] Lin Jiao, Chengjun Xie, Peng Chen, Jianming Du, Rui Li, and Jie Zhang, "Adaptive feature fusion pyramid network for multiclasses agricultural pest detection," *Computers and Electronics in Agriculture*, vol. 195, pp. 106827, 2022.

[7] Zhengfang Hu, Yang Xiang, Yajun Li, Zhenhuan Long, Anwen Liu, Xiufeng Dai, Xiangming Lei, and Zhenhui Tang, "Research on identification technology of field pests with protective color characteristics," *Applied Sciences*, vol. 12, no. 8, pp. 3810, 2022.

[8] Chao Chen, Yundong Liang, Le Zhou, Xiuying Tang, and Mengchu Dai, "An automatic inspection system for pest detection in granaries using yolov4," *Computers and Electronics in Agriculture*, vol. 201, pp. 107302, 2022.

[9] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[10] Glenn Jocher, "YOLOv5," https://github.com/ultralytics/yolov5, 2020, Accessed on Feb. 23, 2023.

[11] Alexander Sutin, Timothy Flynn, Hady Salloum, Nikolay Sedunov, Yegor Sinelnikov, and Helen Hull-Sanders, "Vibroacoustic methods of insect detection in agricultural shipments and wood packing materials," in *IEEE International Symposium on Technologies for Homeland Security*, 2017, pp. 1–6.

[12] Yehao Li, Ting Yao, Yingwei Pan, and Tao Mei, "Contextual transformer networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 1489–1500, 2022.

[13] Glenn Jocher and Ayush Chaurasia, "YOLOv8," https://github.com/ultralytics/ultralytics, 2023, Accessed on Feb. 19, 2023.

[14] Mingxing Tan, Ruoming Pang, and Quoc V Le, "EfficientDet: Scalable and efficient object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10781–10790.

[15] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin, "CARAFE: Content-aware reassembly of features," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3007–3016.

[16] Chao Li, Aojun Zhou, and Anbang Yao, "Omni-dimensional dynamic convolution," in *International Conference on Learning Representations (ICLR)*, 2022, pp. 1–12.

[17] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al., "YOLOv6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021, pp. 1–12.

[19] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R Scott, and Weilin Huang, "TOOD: Task-aligned one-stage object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3490–3499.

[20] Qi-Jin Wang, Sheng-Yu Zhang, Shi-Feng Dong, Guang-Cai Zhang, Jin Yang, Rui Li, and Hong-Qiang Wang, "Pest24: A large-scale very small object data set of agricultural pests for multi-target detection," *Computers and Electronics in Agriculture*, vol. 175, pp. 105585, 2020.

[21] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani, "Bottleneck transformers for visual recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16519–16529.

[22] Qibin Hou, Daquan Zhou, and Jiashi Feng, "Coordinate attention for efficient mobile network design," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13713–13722.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 1–9.

[24] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He, "FCOS: Fully convolutional one-stage object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9627–9636.

[25] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 213–229.

[26] Chien-Yao Wang, Alexey Bochkovskiy, and Liao Hong-Yuan Mark, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.