A comprehensive survey on object detection YOLO

Xiangheng Wang[1], Hengyi Li[2], Xuebin Yue[2] and Lin Meng[3,*]

[1]Graduate School of Science and Engineering, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga, 525-8577, Japan

[2]Research Organization of Science and Technology, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga, 525-8577, Japan

[3]College of Science and Engineering, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga, 525-8577, Japan

Abstract

As a single-stage object detection framework, the YOLO (You Only Look Once) technique has emerged as a prominent technique for various object detection tasks owing to its impressive balance between speed and precision. This research article presents a comprehensive review of the YOLO family of algorithms. This review covers the evolutionary journey of YOLO from its initial release to the latest versions, encompassing an in-depth analysis of the performance and critical characteristics exhibited by each iteration. Particular emphasis is given to exploring the applications of YOLO in diverse domains, focusing on its role in real-time object detection on embedded systems. Furthermore, the paper delves into the latest advancements in compressing algorithms for optimizing the cumbersome YOLO models and practical implementation examples. The potential of deploying YOLO on resource-constrained devices is further unlocked by addressing the challenge of model size reduction. Finally, this study outlines potential research trends and improvements for the YOLO family of algorithms, including novel architectural designs and innovative training strategies. Overall, the thorough investigation presented in this review is a valuable reference for researchers seeking to explore the YOLO framework and its evolving landscape in object detection.

Keywords

YOLO, object detection, deep learning, application, compressing

1. Introduction

Object detection is the main task of computer vision which is to locate the object of interest from the input image target and then accurately judge the class of each target of interest. In recent years, object detection as a popular task in computer vision due to its wide range of applications and recent technological breakthroughs.

Traditional target detection algorithm uses sliding window or image segmentation technology to generate a large number of candidate regions and then extracts image features for each

The 5th International Symposium on Advanced Technologies and Applications in the Internet of Things (ATAIT 2023), August 28-29, 2023, Kusatsu, Japan

*Corresponding author.

†These authors contributed equally.

ri0095fs@ed.ritsumei.ac.jp (X. Wang); lihengyi@fc.ritsumei.ac.jp (H. Li); yue-xb@fc.ritsumei.ac.jp (X. Yue); menglin@fc.ritsumei.ac.jp (L. Meng)

0009-0005-6213-2126 (X. Wang); 0000-0003-4112-7297 (H. Li); 0000-0002-4356-9243 (X. Yue); 0000-0003-4351-6923 (L. Meng)

CEUR Workshop Proceedings (CEUR-WS.org)

candidate region such as Histograms of Oriented Gradients (HOG)[1]. These features are passed to a classifier like Support Vector Machine (SVM)[2] to judge the category of the candidate region. However, traditional target detection algorithm often needs to produce many candidate boxes, resulting in low efficiency. The speed and accuracy of detection can not meet the requirement of practical application, so developing a traditional target detection algorithm falls into the bottleneck. The deep convolutional neural network has been applied in various domains since AlexNet[3] was released in 2012. Deep learning provides a new method for object detection. Since then, a series of models based on deep learning have been proposed, and researchers have focused more on deep learning.

Object detection based on deep learning can be divided into two categories according to detection methods: Region-based and regression-based. The region-based system is known as a two-stage detector that first determines the candidate boxes of the samples and then classifies the samples through the convolutional neural networks (CNN), such as Regional CNN (R-CNN)[4].

The regression-based system is called a one-stage detector that does not generate candidate boxes during processing and directly realizes target detection based on a specific regression analysis. The comparative analysis shows that the characteristics of the two methods are different, the two-stage detectors were better than one-stage models in accuracy, but the real-time performance was slightly slower. Due to one-stage detectors' comprehensive performance and excellent operational efficiency, researchers have focused more on the one-stage detector. The most typical representative of one-stage detectors is YOLO[5].

Due to the superior performance of YOLO, researchers have made many improvements to it, and YOLO has been updated to YOLOv8[6]. In addition, there are YOLOR[7] and YOLOX[8].

Shao et al.[9] provide a detailed description of the YOLO algorithms, but the article is limited to YOLOv5[10] and does not include the latest version of the YOLO family. Terven et al.[11] make a summary of YOLO but need to be more intuitive. In this article, we introduce the features of YOLOv1-v8, YOLOR, and YOLOX to give researchers an understanding of the YOLO families. It provides help for researchers to choose models according to their own needs.

The rest of this paper is organized as follows. The architectures of the YOLO family are discussed in Section 2 and summarized according to structure, image size, AP, AP_{50}, FPS, and Parameters. Section 3 elaborates on the improvements and applications of YOLO in various domains. Section 4 introduces model compression and summarizes the applications of compressed YOLO models. Finally, section 5 summarizes the development trend of the YOLO framework and gives an outlook on the future of YOLO.

2. Systematic overview on the key features of YOLO families

The initial YOLO was presented by Joseph Redmon et al. in CVPR 2016. The YOLO means "You Only Look Once," which glances at an image like a human and offhandedly knows what object is in the image, what they are doing, and where they are. Unlike traditional object detection models based on deep learning, YOLO had excellent accuracy and speed as a regression-based object detection algorithm. YOLO is an advanced one-stage object detection framework that has evolved over the years and spawned several versions. This section introduces the differences between YOLOv1-YOLOv8, YOLOX, and YOLOR. YOLO algorithm forgoes the traditional sliding window technique, and it divides the input image into *S*×*S* grids, each of which predicts B bounding boxes of the same class and the confidence of each grid for *C* different classes. Each bounding box predicts five values: (*x*,*y*,*w*,*h*,*c*), representing the bounding box's position, size, and confidence, respectively. Each grid predicts (*B*×5+*C*) values, after which Non-Maximum Suppression (NMS) is used to remove duplicate detections.

2.1. YOLOv1

Accuracy, including classification accuracy and localization accuracy, and detection speed are essential criteria to judge the quality of image object detection model[12]. The YOLO model does not need to generate candidate boxes and classify them through the CNN network but directly regresses the candidate boxes and categories of the target in multiple positions of the image. YOLOv1 resizes the input image to 448×448 and trains it with the features extracted by CNN. And then, YOLOv1 processes the prediction results to achieve end-to-end object detection.

YOLOv1[5] uses a backbone network similar to GoogLeNet[13], with 24 convolutional layers and two fully connected layers, and 1×1 convolutional layers are used to reduce the number of feature maps. YOLOv1 is pre-trained on ImageNet[14] and then transferred to validate on VisualObject Classes (VOC) dataset. YOLOv1 divides the input image into 7×7 grids, and each grid predicts two bounding boxes, so there are 7×7×2 bounding boxes. A maximum of 49 targets were identified. YOLOv1 is not good at identifying dense targets and small targets.

Evaluated on the PASCAL VOC2007, YOLOv1 scores 63.4% average precision (AP).

2.2. YOLOv2

YOLOv2[15] inspires by the architecture of VGG[16] and constructs a Darknet-19 network that contains 19 convolutional layers and five max pooling layers. YOLOv1 uses the fully connected layer to predict the bounding box directly and loses more spatial information, causing inaccurate positioning. YOLOv2 introduces anchor boxes from Faster R-CNN[17] instead of fully connected layers to predict bounding boxes. Furthermore, YOLOv2 uses batch normalization to improve convergence—the authors through changing the input size to achieve robustness.

In addition, to achieve the goal of extending object detection to objects lacking detection samples, YOLOv2 uses various datasets to optimize the training jointly. The WordTree method is trained synchronously on the ImageNet classification dataset and MS COCO dataset to achieve real-time detection with more than 9000 object categories. The improved YOLOv2 is also known as YOLO9000.

Evaluated on the PASCAL VOC2007, YOLOv2 achieves 78.6% AP, which is 15.2% higher than YOLOv1.

2.3. YOLOv3

YOLOv3[18] uses the Darknet-53 as the backbone. The architecture of YOLOv3 draws on the residual structure of ResNet[19] to deepen the network structure, which solves the problem of network gradient explosion that makes the network difficult to converge. YOLOv3 utilizes a method similar to Feature Pyramid Network (FPN)[20] and performs multi-scale training to predict three boxes at three different scales. Moreover, YOLOv3 also uses k-means to determine the prior of the bounding box of the anchor box. Unlike YOLOv2, YOLOv3's architecture uses three prior boxes for the scales of the large, medium, and small objects. Furthermore, feature maps of three different scales are used for object detection by feature fusion, and logistics is used to replace softmax for category prediction to achieve multi-label object detection. The proposed network not only improves the performance of small objects but also achieves 3 to 4 times faster than previous YOLO models when the bounding box prediction is not strict, and the detection accuracy is similar.

Due to the rapid development of computer vision, evaluation datasets that can better reflect the comprehensive performance of detection algorithms are needed. When YOLOv3 was released, the evaluation benchmark for object detection changed from PASCAL VOC to MS COCO. Therefore, YOLOv3 and subsequent YOLO models are evaluated on the MS COCO dataset. The YOLOv3 algorithm meets the accuracy and speed requirements of real-time detection and has become one of the preferred target detection algorithms in the engineering field.

Evaluated on the MS COCO dataset test-dev 2017, YOLOv3 achieves 31.0% AP and 55.3% AP_{50} at 20 FPS.

2.4. YOLOv4

After YOLOv3, YOLOv4[21] adopts various improvement methods, which are divided into bag-of-freebies and bag-of-specials: the bag-of-freebies refer to modules that improve training without affecting inference speed, and the bag-of-specials refer to modules that have less impact on inference time and higher performance returns. The key features include Cross Stage Partial (CSP)[22], and Mish activation function[23] are adopted in the backbone network. CSPNet solves the bottleneck of the repetition of gradient information in other large neural network frameworks. It uses a unique method to integrate the gradient into the feature map fully, so this method can effectively reduce the number of parameters and FLOPs values of the model.

Therefore, compared with other YOLO models, YOLOv4 has higher accuracy while maintaining a higher inference speed. Furthermore, YOLOv4 is more suitable for training on a single GPU. The architecture of YOLOv4 operates CSPDarknet-53 as the backbone. For the neck, authors also use tricks from YOLOv3-SPP, including a modified version of spatial pyramid pooling (SPP)[24], multi-scale prediction, and a modified path aggregation network (PANet)[25] instead of FPN and improved Spatial Attention Module (SAM)[26]. Moreover, anchor boxes are used in the head to extract features.

Evaluated on the MS COCO dataset test-dev 2017, YOLOv4 achieves 41.2% AP and 62.8% AP_{50} at over 96 FPS by NVIDIA Tesla V100.

2.5. YOLOv5

The basic structure of YOLOv5[10] is similar to YOLOv4. The significant difference is the scaling based on different channels. YOLOv5 provides five scale models of YOLOv5-N (nano) /S (small) / M (medium) / L (large) / X (extra large) are constructed from small to large models. Pytorch develops YOLOv5, which is easier to deploy on hardware than YOLOv4. As of this writing, official papers have yet to be published for YOLOv5. According to the homepage, YOLOv5 has been updated to the seventh edition. In the latest version, it is capable of handling classification and instancing segmentation tasks and speeds up training.

Evaluated on the MS COCO dataset test-dev 2017, with image size is 1536×1536, YOLOv5x6 obtains 55.8%AP. In case the image size is 640×640, YOLOv5x achieves 50.7% AP and exceeds 200 FPS on NVIDIA Tesla V100.

2.6. YOLOR

The YOLOv4 team releases You Only Learn One Representation (YOLOR)[7] in 2021. They use implicit knowledge to address the fact that features extracted from previously trained convolutional neural networks are often less adaptable to other problems.

Human beings can acquire explicit knowledge through regular learning and implicit knowledge through subconscious learning. Even things that people have not seen can be judged by experience. YOLOR proposes a unified network combining implicit and explicit knowledge to give the network a learning ability similar to the human brain. YOLOR can be applied in many fields, such as segmentation and detection.

Evaluated on the MS COCO dataset test-dev 2017, YOLOR-D6 achieves 55.4% AP and 73.3% AP_{50} at 30 FPS by NVIDIA Tesla V100. The results demonstrate that the performance of all tasks improves after introducing implicit knowledge into the neural network.

2.7. YOLOX

YOLOX[8] is released by Megvii Technology in 2021. The model is based on YOLOv3 by Pytorch. Compared with other YOLO models, YOLOX has three main changes: decoupled head, anchor-free, and advanced label assigning strategy (SimOTA).

Decoupled head: The conflict between classification and regression tasks is an unavoidable problem[27][28]. According to the author's experimental analysis, the coupled detection head may damage the performance, so the head is replaced with a decoupled head. The decoupled head is divided into two parts, one for regression tasks and the other for classification tasks. The structure speeds up the convergence of the network.

Anchor-free: Although the anchor mechanism works well for specific domains, it increases the complexity of the detection head and may cause delays when deployed on edge hardware. Inspired by the target detection models of FCOS[29], YOLOX uses an anchor-free mechanism that reduces the number of parameters and GFLOPs of the detector and makes the model faster.

SimOTA: Based on the research of OTA[30], YOLOX reconsiders the tag assignment from a global perspective and proposes to formulate the assignment process as an Optimal Transport (OT) problem. The SimOTA technique proposed by YOLOX achieves better performance with reduced training time.

Evaluated on the MS COCO dataset test-dev 2017, YOLOX-L achieves the best performance is 50.1% AP on COCO at a speed of 68.9 FPS on NVIDIA Tesla V100.

2.8. YOLOv6

The Meituan Vision AI Department released YOLOv6[31] in 2022. The overall structure refers to YOLOv4 but uses a more advanced mechanism. Firstly, they use a new backbone efficiency developed based on RepVGG[32]. Secondly, the neck of YOLOv6 adopts the PANet[25], and RepPAN is obtained after the neck is enhanced. YOLOv6's architecture is similar to YOLOX. YOLOv6 also uses Efficient Decoupled Head and anchor-free technology. In addition, YOLOv6 also uses a self-distillation strategy to reduce the cost of reasoning. Similar to YOLOv5, YOLOv6 also provides multiple versions to facilitate model quantification and hardware deployment. It is also suitable for application in the industrial field.

Evaluated on the MS COCO dataset test-dev 2017, YOLOv6-L achieves an AP of 52.5% and AP_{50} of 70.0% on a NVIDIA Tesla T4 in the same environment with TensorRT.

2.9. YOLOv7

The research team of YOLOv4 and YOLOR propose YOLOv7[33] 2022. YOLOv7 outperforms all known object detectors in speed and accuracy, ranging from 5 FPS to 160 FPS. Like YOLOv4, YOLOv7 is trained from scratch on the MS COCO dataset.

The main changes in the architecture of YOLOv7 are Extended efficient layer aggregation networks (E-ELAN) by improved ELAN and Model scaling for concatenation-based models. If more computational blocks are stacked indefinitely, it may destroy the stable state of the network. ELAN[34] enables the deeper network to learn and converge efficiently by controlling the shortest and longest gradient path. E-ELAN proposed by YOLOv7 further takes expand, shuffles, and merges cardinality to achieve the ability to continuously enhance the learning ability of the network without destroying the original gradient path.

Unlike architectures such as ResNet, applying model scaling to a concatenation-based architecture results in a change in the ratio of input and output channels, which leads to the model's inefficiency for hardware. YOLOv7 proposed a composite scaling approach that maintains the characteristics of the model at the time of initial design and maintains the optimal architecture. The composite scaling method proposed by YOLOv7 maintains the model's properties at the initial design. It preserves the optimal structure by scaling the width factor on the transition layer with the same amount of variation.

Evaluated on the MS COCO dataset test-dev 2017, when input image size is 640×640, YOLOv7 achieves AP of 51.4% and AP_{50} of 69.7% at 161 FPS by NVIDIA Tesla V100.

2.10. YOLOv8

The YOLOv5 team releases YOLOv8[6] in January 2023, and an official article still needs to be published. The main improvement as follows:

Backbone: The backbone of YOLOv8 is CSPDraknet-53. As YOLOv5, YOLOv8's C3 module is replaced by the C2f module with richer gradient flow, and different channel numbers are adjusted for different scale models to achieve further lightweight. Moreover, YOLOv8 still uses the SPPF module used in YOLOv5.

Head: The Head part has two significant improvements compared with YOLOv5. Firstly, it is replaced with the current mainstream Decoupled-Head, which separates the classification and detection. Secondly, it is also changed from Anchor-Based to Anchor-Free.

Loss: YOLOv8 abandons the previous IOU matching or unilateral ratio distribution method

Table 1
Summary of YOLO families architecture

Model	Backbone	Neck	Anchor
YOLOv1	GoogLeNet,VGG-16	2×fully connected layers	No
YOLOv2	Darknet-19	fully connected layers	**Yes**
YOLOv3	Darknet-53	FPN	**Yes**
YOLOv4	**CSPDarknet-53**	SPP,PANet	**Yes**
YOLOv5	CSPDarknet-53	SPPF,CSP-PAN	**Yes**
YOLOR	CSPDarknet-53	FPN,SPP	**Yes**
YOLOX	Modified CSP v5	FPN	No
YOLOv6	EfficientRep	Rep-PAN	No
YOLOv7	Extended-ELAN	SPPCSPC	No
YOLOv8	Darknet-53	SPP,PAN	No

but takes the Task-Aligned Assigner positive and negative sample matching method. Furthermore, YOLOv8 introduced Distribution Focal Loss (DFL)[35].

Data augmentation can improve model performance, but introducing mosaic augmentation in training may have adverse effects. YOLOv8 turns off the Mosaic augmentation in the last ten epochs, improving accuracy. To meet the needs of different scenarios, YOLOv8 provides different size models of N / S / M / L / X scales.

Evaluated on MS COCO dataset test-dev 2017, YOLOv8X achieves an AP of 53.9% with 283 FPS on NVIDIA Tesla A100 and TensorRT.

2.11. Summary

The architecture of the YOLO families is shown in Table**??**, including the backbone, neck, and anchor. YOLOv2 first proposes darknet-19 as the backbone, and YOLOv3 deepened the 19 convolutional layers to 53 layers. Almost all subsequent YOLO models use Darknet-53 or an improved version of darknet-53 as the backbone architecture. The initial YOLO does not use anchors, which were used in YOLOv2, and improves the prediction accuracy until YOLOX takes an anchor-free approach and performs well. Since then, subsequent versions of YOLO dropped the use of anchors.

Table2 shows the performance, size, FPS, parameters, and GPU used for training mainstream YOLO versions. Among them, YOLOv1 and YOLOv2 were tested on PASCAL VOC2007. When YOLOv3 was released, the benchmark for object detection had changed from PASCAL VOC to Microsoft COCO. Therefore, the performance of subsequent versions is tested on Microsoft COCO. In addition, the YOLOv6 and YOLOv8x models are quantized by TensorRT. From the parameters in Table2, YOLO increasingly favors lighter models. In YOLOv8, the model YOLOv8v8m is only 3.7% less than the AP_{50} of YOLOv8x, but the parameters are only about 38% of that of YOLOv8x, which is 25.9M. In many versions of YOLO, researchers can choose models according to their needs to find a balance between accuracy and speed.

Table 2
Summary of YOLO families main indexes

Model	Size	AP(%)	AP_{50}(%)	GPU	Params(M)	FPS
YOLOv1	448	63.4	-	TitanX	-	-
YOLOv2	448	78.6	-	TitanX	-	-
YOLOv3	416	31.0	55.3	-	-	34.5
YOLOv4	416	41.2	62.8	Tesla V100	64.4	96
YOLOv5x	640	50.7	68.9	Tesla V100	86.7	208.3
YOLOR-D6	1280	**55.4**	**73.3**	Tesla V100	152.0	30
YOLOX-L	640	50.1	68.5	Tesla V100	54.2	69.0
YOLOv6	640	52.5	70.0	Tesla T4	58.5	121
YOLOv7	640	51.4	69.7	Tesla V100	**36.9**	161
YOLOv8x	640	-	53.9	Tesla V100	68.2	**283.3**

3. Improvements and Applications

YOLO models have been used for Industry, AIoT, Health Care, and the Protection of Cultural Heritage. In industry, many YOLO applications exist for robots, traffic, and personal protection equipment. In terms of the empty-dish recycling robots, Yue et al.[36][37] solve the problem that the traditional object detection model requires store parameters, and propose a lightweight dish detection model based on the YOLOX for an empty-dish recycling robot. Ge et al.[38] through lightweight YOLO-GG to enable the recycling of empty plates efficiently. For traffic, He et al.[39] design a flexible and efficient one-stage object detection network FE-YOLO for the rail transit scene. Li et al.[40] optimize the YOLOv5 model and propose a helmet detection system to ensure the safety of workers.

The field of AIOT has also attracted much attention. AIoT integrates artificial intelligence (AI) technology and the Internet of Things (IoT) in practical applications. Morioka et al.[41] propose a YOLO model-based Android system for ancient text recognition, implemented by communicating with a server equipped to recognize AI models. Building Smart City Traffic Management Systems based on artificial intelligence and big data has become a trend. Liu et al.[42] use YOLOv3 to detect vehicles and estimate their length by image processing.

YOLO is also widely used in other health care, [43] propose a system based on YOLOv4-tiny that was quantified and deployed on Jetson-nano at the beginning of COVID-19. The system identifies the wearing status of masks and measures social distance. It effectively protects people's health. Zhuang et al.[44] combine YOLO and the improved two-dimensional continuity equation for the cardiac Vector Flow Mapping (VFM) analysis and evaluation.

Besides, ancient books record much information, and decoding these classic books is favorable for studying history, politics, and culture. Liu[45] and Fujikawa[46] use YOLO to detect and identify Oracle Bone Inscription (OBI).

To sum up, YOLO models have been shown to play an essential role in various fields. It has become a trend to optimize the YOLO model for different applications.

4. Compression methods for YOLO models

The YOLO model has become more complex in pursuit of better performance. Model compression has become the focus of research to facilitate deployment on edge devices. This section introduces several techniques for model compression, such as model pruning, parameter quantization, knowledge distillation, and lightweight model design[47].

Model Pruning: Model pruning is achieved by searching for redundant layers/channels in the model and deleting them with little or no impact on performance. By pruning the YOLOv3-tiny network, Shi et al.[48] reduce the network computation by 68.7% Wu et al.[49] propose a real-time apple flower detection using the channel-pruned YOLOv4 model. The number of parameters is reduced by 96.74%.

Parameter Quantization: Parameter quantization converts floating-point calculations to low-bit-rate integer calculations, such as converting float32 to int8 or int4. We quantize the YOLOv4 model on 8-bit through TensorRT and deploy it to the Jetson Nano, and it achieves the purpose of real-time detection of dirty eggs[50]. Wang et al.[51] deploy the pruned YOLOv3 detection model on the FPGA, and the model size is reduced by 80% with little change in accuracy.

Knowledge Distillation: The teacher network is a complex pre-trained network, and the student network is a simple small network. By transferring knowledge, the student network that is more suitable for reasoning can be obtained through the teacher network. Chen et al.[52] propose a lightweight ship detector by knowledge distillation. Xing et al.[53] use ResNet101 as the teacher network and DD-YOLO as the student network, reducing the model complexity to 61.4%, which is more suitable for mobile deployment.

Lightweight Model Design: Lightweight DNN model design refers to the redesign based on the existing deep neural network structure to reduce the number of parameters and computational complexity. Liu et al.[54] replace the backbone of YOLOv3 with ShuffleNet to realize real-time vehicle detection. Liu et al.[55] uses the backbone of YOLOv4 with Mobilenetv3, which improves the accuracy of an extensive pedestrian detection network.

5. Conclusion

This paper summarizes the models of the YOLO series and provides a detailed analysis of the critical feature of each model.Afterward, the application of YOLO in different scenarios was introduced. Finally, the method of YOLO model compression was briefly described, and an illustration of the application of YOLO in model compression.

Although the YOLO series is the leader in the speed-accuracy balance in the field of target detection, its main work is for the computer side. In further work, how to make YOLO lighter and faster is worth pondering, especially embedded devices such as Nvidia Jetson Nano and Raspberry Pi. Moreover, it will become a trend to carry DNN models on FPGA to make the model run more efficiently. In addition, the technology combining AI and IoT will also bring more convenience to human life. Finally, since the release of YOLOv4, integrating various advanced algorithms has become an essential way to develop the YOLO algorithm. With the development of the YOLO framework, YOLO is more versatile and powerful and will be applied in a broader range of fields.

References

[1] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), volume 1, San Diego, CA, USA, 2005, pp. 886–893. doi:10.1109/CVPR.2005.177.

[2] N. Cristianini, J. Shawe-Taylor, An introduction to support vector machines and other kernel-based learning methods, Cambridge university press, 2000.

[3] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Communications of the ACM 60 (2017).

[4] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 580–587. doi:10.1109/CVPR.2014.81.

[5] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779–788. doi:10.1109/CVPR.2016.91.

[6] Glenn Jocher and Ayush Chaurasia and Jing Qiu, Yolo by ultralytics, 2023. URL: https://github.com/ultralytics/ultralytics.

[7] C.-Y. Wang, I.-H. Yeh, H.-Y. M. Liao, You only learn one representation: Unified network for multiple tasks, arXiv preprint arXiv:2105.04206 (2021).

[8] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, arXiv preprint arXiv:2107.08430 (2021).

[9] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du, X. Lan, A survey of deep learning-based object detection, Multimedia Tools and Applications 79 (2020). doi:10.1007/s11042-020-08976-6.

[10] Glenn Jocher, Yolov5 by ultralytics, 2020. URL: https://github.com/ultralytics/yolov5.

[11] J. Terven, D. Cordova-Esparza, A comprehensive review of yolo: From yolov1 to yolov8 and beyond, arXiv preprint arXiv:2304.00501 (2023).

[12] Z. Zou, K. Chen, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: A survey, Proceedings of the IEEE 111 (2023). doi:10.1109/JPROC.2023.3238524.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1–9. doi:10.1109/CVPR.2015.7298594.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (2015). doi:10.1007/s11263-015-0816-y.

[15] J. Redmon, A. Farhadi, Yolo9000: Better, faster, stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 6517–6525. doi:10.1109/CVPR.2017.690.

[16] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[17] R. Girshick, Fast r-cnn, in: 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1440–1448. doi:10.1109/ICCV.2015.169.

[18] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, CoRR abs/1804.02767 (2018). URL: http://arxiv.org/abs/1804.02767. arXiv:1804.02767.

[19] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 936–944. doi:10.1109/CVPR.2017.106.

[21] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934 (2020).

[22] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, I.-H. Yeh, Cspnet: A new backbone that can enhance learning capability of cnn, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2020, pp. 1571–1580. doi:10.1109/CVPRW50498.2020.00203.

[23] D. Misra, Mish: A self regularized non-monotonic activation function, arXiv preprint arXiv:1908.08681 (2019).

[24] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (2015). doi:10.1109/TPAMI.2015.2389824.

[25] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 8759–8768. doi:10.1109/CVPR.2018.00913.

[26] S. Woo, J. Park, J.-Y. Lee, I. S. Kweon, Cbam: Convolutional block attention module, in: Computer Vision – ECCV 2018, 2018, pp. 3–19. doi:10.1109/CVPR.2018.00913.

[27] G. Song, Y. Liu, X. Wang, Revisiting the sibling head in object detector, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 11560–11569. doi:10.1109/CVPR42600.2020.01158.

[28] Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, Y. Fu, Rethinking classification and localization for object detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 10183–10192. doi:10.1109/CVPR42600.2020.01020.

[29] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 9626–9635. doi:10.1109/ICCV.2019.00972.

[30] Z. Ge, S. Liu, Z. Li, O. Yoshie, J. Sun, Ota: Optimal transport assignment for object detection, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 303–312. doi:10.1109/CVPR46437.2021.00037.

[31] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, et al., Yolov6: A single-stage object detection framework for industrial applications, arXiv preprint arXiv:2209.02976 (2022).

[32] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, J. Sun, Repvgg: Making vgg-style convnets great again, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition

(CVPR), Nashville, TN, USA, 2021, pp. 13728–13737. doi:10.1109/CVPR46437.2021.01352.

[33] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, arXiv preprint arXiv:2207.02696 (2022).

[34] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Designing network design strategies through gradient path analysis, arXiv preprint arXiv:2211.04800 (2022).

[35] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, J. Yang, Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection, Advances in Neural Information Processing Systems 33 (2020).

[36] X. Yue, H. Li, M. Shimizu, S. Kawamura, L. Meng, Yolo-gd: A deep learning-based object detection algorithm for empty-dish recycling robots, Machines 10 (2022). doi:10.3390/machines10050294.

[37] X. Yue, H. Li, L. Meng, An ultralightweight object detection network for empty-dish recycling robots, IEEE Transactions on Instrumentation and Measurement 72 (2023). doi:10.1109/TIM.2023.3241078.

[38] Y. Ge, X. Yue, L. Meng, A high-efficiency dirty-egg detection system based on yolov4 and tensorrt, in: 2022 International Conference on Advanced Mechatronic Systems (ICAMechS), Toyama, Japan, 2022, pp. 59–63.

[39] D. He, Z. Zou, Y. Chen, B. Liu, J. Miao, Rail transit obstacle detection based on improved cnn, IEEE Transactions on Instrumentation and Measurement 70 (2021). doi:10.1109/TIM.2021.3116315.

[40] Z. Li, W. Xie, L. Zhang, S. Lu, L. Xie, H. Su, W. Du, W. Hou, Toward efficient safety helmet detection based on yolov5 with hierarchical positive sample selection and box density filtering, IEEE Transactions on Instrumentation and Measurement 71 (2022). doi:10.1109/TIM.2022.3169564.

[41] T. Morioka, C. Aravinda, L. Meng, An ai-based android application for ancient documents text recognition, in: Proceedings of the 2021 International Symposium on Advanced Technologies and Applications in the Internet of Things, Virtual, Kusatsu, Japan, 2021, pp. 91–98.

[42] Y. Liu, M. Reynolds, D. Huynh, G. Hassan, Study of accurate and fast estimation method of vehicle length based on yolos, in: 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS), Dalian, China, 2020, pp. 118–121. doi:10.1109/ICAIIS49377.2020.9194930.

[43] X. Yue, H. Li, L. Meng, Ai-based prevention embedded system against covid-19 in daily life, Procedia Computer Science 202 (2022).

[44] Z. Zhuang, G. Liu, W. Ding, A. N. J. Raj, S. Qiu, J. Guo, Y. Yuan, Cardiac vfm visualization and analysis based on yolo deep learning model and modified 2d continuity equation, Computerized Medical Imaging and Graphics 82 (2020).

[45] G. Liu, J. Xing, J. Xiong, Spatial pyramid block for oracle bone inscription detection, in: Proceedings of the 2020 9th International Conference on Software and Computer Applications, New York, NY, USA, 2020„ p. 133–140. doi:10.1145/3384544.3384561.

[46] Y. Fujikawa, H. Li, X. Yue, C. Aravinda, G. A. Prabhu, L. Meng, Recognition of oracle bone inscriptions by using two deep learning models, International Journal of Digital Humanities (2022). doi:10.1007/s42803-022-00044-9.

[47] Z. Li, H. Li, L. Meng, Model compression for deep neural networks: A survey, Computers

12 (2023). doi:10.3390/computers12030060.

[48] R. Shi, T. Li, Y. Yamaguchi, An attribution-based pruning method for real-time mango detection with yolo network, Computers and Electronics in Agriculture 169 (2020). doi:10.1007/s11263-014-0733-5.

[49] D. Wu, S. Lv, M. Jiang, H. Song, Using channel pruning-based yolo v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments, Computers and Electronics in Agriculture 178 (2022). doi:10.1016/j.compag.2020.105742.

[50] X. Wang, X. Yue, H. Li, L. Meng, A high-efficiency dirty-egg detection system based on yolov4 and tensorrt, in: 2021 International Conference on Advanced Mechatronic Systems (ICAMechS), Tokyo, Japan, 2021, pp. 75–80. doi:10.1109/ICAMechS54019.2021.9661509.

[51] Z. Wang, H. Li, X. Yue, L. Meng, Design and acceleration of field programmable gate array-based deep learning for empty-dish recycling robots, Applied Sciences 12 (2022). doi:10.3390/app12147337.

[52] S. Chen, R. Zhan, W. Wang, J. Zhang, Learning slimming sar ship object detector through network pruning and knowledge distillation, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14 (2021). doi:10.1109/JSTARS.2020.3041783.

[53] Z. Xing, X. Chen, F. Pang, Dd-yolo: An object detection method combining knowledge distillation and differentiable architecture search, IET Computer Vision 16 (2022). doi:10.1049/cvi2.12097.

[54] J. Liu, R. Zhang, Vehicle detection and ranging using two different focal length cameras, Journal of Sensors 14 (2020). doi:10.1155/2020/4372847.

[55] L. Liu, C. Ke, H. Lin, H. Xu, et al., Research on pedestrian detection algorithm based on mobilenet-yolo, Computational intelligence and neuroscience 2022 (2022). doi:10.1155/2022/8924027.