2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)

YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness

1st Rejin Varghese
Department of Computer Applications
Hindustan Institute of Technology and Science
Chennai, India
rp.21703018@student.hindustanuniv.ac.in

2nd Sambath M.
School of Computing Science
Hindustan Institute of Technology and Science
Chennai, India
msambath@hindustanuniv.ac.in

2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS) | 979-8-3503-6482-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/ADICS58448.2024.10533619

***Abstract*— In recent years, the You Only Look Once (YOLO) series of object detection algorithms have garnered significant attention for their speed and accuracy in real-time applications. This paper presents YOLOv8, a novel object detection algorithm that builds upon the advancements of previous iterations, aiming to further enhance performance and robustness. Inspired by the evolution of YOLO architectures from YOLOv1 to YOLOv7, as well as insights from comparative analyses of models like YOLOv5 and YOLOv6, YOLOv8 incorporates key innovations to achieve optimal speed and accuracy. Leveraging attention mechanisms and dynamic convolution, YOLOv8 introduces improvements specifically tailored for small object detection, addressing challenges highlighted in YOLOv7. Additionally, the integration of voice recognition techniques enhances the algorithm's capabilities for video-based object detection, as demonstrated in YOLOv7. The proposed algorithm undergoes rigorous evaluation against state-of-the-art benchmarks, showcasing superior performance in terms of both detection accuracy and computational efficiency. Experimental results on various datasets confirm the effectiveness of YOLOv8 across diverse scenarios, further validating its suitability for real-world applications. This paper contributes to the ongoing advancements in object detection research by presenting YOLOv8 as a versatile and high-performing algorithm, poised to address the evolving needs of computer vision systems.**

Keywords—YOLOv8, Object Detection, Performance Enhancement, Robustness, Computational Efficiency, Computer Vision Systems

I. INTRODUCTION

Recognizing objects could be a vital and complex errand within the field of computer vision, with applications traversing security, observation, self-driving vehicles, robotics, and medical imaging. The objective of object location is to find and classify objects in pictures or recordings, giving their bounding boxes and names. There are two primary sorts of protest location methods: two-stage methods and one-stage methods. Two-stage methods, such as R-CNN, Fast R-CNN, and Faster R-CNN, at first generate a set of region recommendations and after that refine them employing a classifier and a regressor. On the other hand, one-stage strategies like SSD, RetinaNet, and YOLO specifically foresee the bounding boxes and names from the input picture, dispensing with the require for locale recommendations. In spite of the fact that one-stage methods are ordinarily speedier and less complex than two-stage strategies, they regularly compromise on exactness and soundness.

YOLO (You only Look Once), a critical one-stage protest discovery calculation, was to begin with presented by Redmon and Farhadi in 2017 [1]. YOLO segments the input picture into a network of cells and predicts a settled number of bounding boxes and certainty scores for each cell. It too predicts the lesson probabilities for each bounding box and combines them with the certainty scores to deliver the final detection comes about. YOLO is eminent for its speed and compelling execution on huge and medium-sized objects, but it has certain impediments, such as moo review, rough localization, and subpar execution on little objects.

Since YOLO's beginning, various variations and improvements have been proposed to address its impediments and boost its execution. Vital cases incorporate YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6, and YOLOv7 [11]. These forms have consolidated different procedures and developments, such as stay boxes, multi-scale predictions, feature pyramid systems, residual connections, consideration instruments, energetic convolutions, and voice acknowledgment. These strategies have upgraded YOLO's precision, strength, and effectiveness, making it more versatile to different circumstances and applications. All things considered, there are still openings for encourage advancement and optimization, especially for challenging scenarios including little objects, blocked objects, and complex foundations.

In this paper, we introduce YOLOv8, a modern object detection algorithm that builds upon the past YOLO forms and consolidates modern highlights and improvements. YOLOv8 endeavours to realize the most elevated speed and precision in object location, whereas guaranteeing strength and soundness.

II. LITERATURE REVIEW

YOLO (You Only Look Once), a single-stage object detection algorithm, was initially introduced by Redmon and Farhadi in 2017 [1]. The YOLO algorithm partitions the input image into a grid of cells and forecasts a predetermined number of bounding boxes and confidence scores for each cell. Additionally, YOLO predicts the class probabilities for each bounding box and merges them with the confidence scores to produce the final detection outcomes. YOLO is recognized for its impressive speed and effective performance on large and medium-sized objects. However, it does have certain drawbacks, such as low recall, imprecise localization, and suboptimal performance on small objects.

The YOLOv2 [1] is an progressed adaptation of YOLO, which presents a few procedures to upgrade the accuracy and efficiency of the algorithm. A few of the most methods are:

- *Anchor boxes* : YOLOv2 employments predefined bounding box shapes, called anchor boxes, to superior fit the objects of distinctive sizes and aspect ratios. YOLOv2 predicts the offsets and scales of the anchor boxes, rather than the supreme

facilitates and measurements of the bounding boxes.

- *Multi-scale predictions* : YOLOv2 predicts bounding boxes at three distinctive scales, comparing to the coarse, medium, and fine features extracted from the input picture. This permits YOLOv2 to identify objects of different sizes more successfully.
- *Batch normalization* : YOLOv2 applies batch normalization to each layer of the network, which diminishes the inner covariate move and progresses the steadiness and merging of the preparing prepare.
- *Darknet-19* : YOLOv2 utilizes a modern backbone network known as Darknet-19, a streamlined and viable convolutional neural organize. Darknet-19 is composed of 19 convolutional layers and 5 max-pooling layers, and it utilizes 3x3 and 1x1 filters to diminish the amount of parameters and computations.

YOLOv3 [2] is another progressed version of YOLO, which advance improves the execution and vigor of the algorithm. A few of the most upgrades are:

- *Feature pyramid network* : YOLOv3 uses a feature pyramid network (FPN) to combine the highlights from distinctive levels of the backbone network, and produce high-quality and different bounding box forecasts. FPN employments skip associations and upsampling operations to combine the low-level and high-level highlights, and produces highlight maps of distinctive resolutions for multi-scale expectations.
- *Residual connections* : The algorithm employments residual connections to encourage the data flow and gradient propagation within the network. Residual connections include the yield of a past layer to the input of a consequent layer, and offer assistance to maintain a strategic distance from the vanishing gradient problem and make strides the exactness of the network.
- *YOLOv3-tiny* : YOLOv3 also provides a littler and speedier adaptation of the network, called YOLOv3-tiny, which is appropriate for resource-constrained gadgets and applications. YOLOv3-tiny employments less layers and channels, and predicts bounding boxes at two scales rather than three.

YOLOv4 [3] is a recent version of YOLO that applies different cutting-edge strategies and advancements to upgrade the speed and accuracy of object detection. A few of the most techniques and advancements are:

- *CSPDarknet53* : This is often a new backbone network, called CSPDarknet53, which is based on the cross-stage partial (CSP) connections and the Darknet-53 network. CSP connections separate the feature maps into two parts, and as it were one part goes through the consequent layers, whereas the other portion is concatenated with the yield of the final layer. This lowers the redundancy and complexity of the network, and boosts the effectiveness and execution of the network.
- *SPP* : Typically a spatial pyramid pooling (SPP) module that aggregates the features from diverse regions of the input image, and progresses the strength and invariance of the network. SPP uses multiple max-pooling layers with distinctive kernel sizes and strides, and concatenates their outputs to make a fixed-length feature vector.
- *PANet* : Usually a way aggregation network (PANet) that improves the feature fusion and data stream within the network. PANet employments bottom-up and top-down ways to aggregate the features from distinctive levels of the network, and uses adaptive feature selection to powerfully alter the weights of the features.
- *Mish* : YOLOv4 uses a modern activation function, called Mish. Mish is a self-regularized and smooth function, which preserves the positive values and suppresses the negative values of the input. Mish has been appeared to outperform other activation functions, such as ReLU, Leaky ReLU, and Swish, in terms of exactness and stability.

Based on the PyTorch framework, YOLOv5 [4] may be a later adaptation of YOLO that gives a straightforward and adaptable solution for object detection. It is an independent project that incorporates some of the concepts and strategies from the past YOLO variations, instead of an official continuation of the YOLO series. YOLOv5 has a few of the taking after features and upgrades:

- *EfficientNet :* Typically a cutting-edge neural network architecture that accomplishes high proficiency and execution on different computer vision tasks. It employments a compound scaling strategy to adjust the depth, width, and determination of the network, and optimizes the network for diverse asset limitations and target accuracies.
- *FPN :* Typically a include pyramid network that fuses the features from diverse levels of the backbone network, and produces high-quality and different bounding box forecasts. It employments skip connections and upsampling operations to combine the low-level and high-level features, and produces feature maps of diverse resolutions for multi-scale forecasts.
- *Data augmentation :* This refers to different data augmentation methods, such as random cropping, flipping, scaling, rotation, color jittering, and mosaic, that increment the differing qualities and complexity of the training data, and upgrade the generalization and robustness of the model.
- *Model variants :* YOLOv5 offers four model variations, to be specific YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, that have diverse trade-offs between speed and precision. YOLOv5s is the smallest and fastest model,

IEEE 979-8-3503-6482-8/24/$31.00 ©2024 IEEE

whereas YOLOv5x is the biggest and most exact model.

YOLOv6 [5] could be a later adaptation of YOLO, which is based on the TensorFlow framework and points to progress the execution and strength of object detection in complex environments. YOLOv6 is additionally not an official continuation of the YOLO series, but or maybe an independent project that presents some new features and upgrades to the YOLO algorithm. A few of the most features and improvements of YOLOv6 are:

- *Attention mechanism :* YOLOv6 uses an attention mechanism to focus on the foremost important and enlightening features for object detection, and suppress the unessential and noisy features. YOLOv6 employments a self-attention module, which computes the closeness between the highlights of distinctive areas, and a channel consideration module, which computes the significance of the features of diverse channels. The attention mechanism makes a difference to move forward the exactness and robustness of the model, particularly for little and impeded objects.
- *Dynamic convolution :* YOLOv6 employments a dynamic convolution method to adapt the convolutional filters to the input picture, and create more discriminative and expressive features for object detection [8]. YOLOv6 uses a conditional convolution layer, which predicts the weights of the convolutional filters based on the input picture, and a dynamic routing layer, which selects the foremost reasonable filters for each feature map. The dynamic convolution procedure makes a difference to move forward the productivity and execution of the model, especially for complex and different scenes.
- *YOLOv6-tiny :* YOLOv6 moreover gives a smaller and quicker adaptation of the network, called YOLOv6-tiny, which is appropriate for resource-constrained devices and applications. YOLOv6-tiny employments less layers and channels, and predicts bounding boxes at two scales rather than three.

YOLOv7 [6][9] is a recent version of YOLO, which is based on the PyTorch framework and points to attain optimal speed and precision of object detection. YOLOv7 is additionally not an official continuation of the YOLO series, but or maybe an independent project that joins different state-of-the-art strategies and advancements to optimize the YOLO algorithm. A few of the most methods and developments of YOLOv7 are:

- *NAS-FPN :* YOLOv7 employments a neural architecture search method, called NAS-FPN, to automatically generate feature pyramid networks for object detection. NAS-FPN uses a reinforcement learning algorithm to explore for the ideal combination of feature fusion operations, such as expansion, concatenation, and max-pooling, and produces feature maps of distinctive resolutions for multi-scale forecasts.
- *Focal loss :* The algorithm uses a modern loss function, called Focal Loss, which could be a loss function that centres on the difficult cases and diminishes the impact of the simple illustrations [10]. Focal Loss employments a modulating factor to down-weight the commitment of the well-classified cases, and a scaling factor to adjust the positive and negative samples.

III. Overview Of Proposed Algorithm

The proposed algorithm, YOLOv8, is the most recent advancement within the YOLO (You only Look Once) series of object detection models. It builds upon the foundational work of YOLO9000, which was recognized for its predominant speed and strength. The ensuing adaptations, YOLOv3 and YOLOv4, encourage made strides the model's execution, especially in complex environments and in accomplishing optimal speed and exactness of object detection.

1. Network Architecture

The structure of YOLOv8 is essentially partitioned into two key components: the backbone network and the detection head. The part of the backbone network is to extricate a assortment of rich features from the input picture at numerous scales. On the other hand, the detection head takes on the task of merging these features and creating different and high-quality forecasts for bounding boxes.

1.1 Backbone Network

The backbone network of YOLOv8 is based on EfficientNet [12], which could be a state-of-the-art neural network architecture that accomplishes high proficiency and performance on various computer vision tasks. EfficientNet is based on the thought of compound scaling, which could be a strategy that scales the network width, depth, and determination in a balanced way. EfficientNet employments a base network, called EfficientNet-B0, which could be a convolutional neural network that has 29 layers and employments modified residual blocks with squeeze-and-excitation modules. EfficientNet at that point scales up the base network to get diverse variants, such as EfficientNet-B1, EfficientNet-B2, EfficientNet-B7, by employing a compound scaling coefficient. The compound scaling coefficient is decided by a grid search that optimizes the trade-off between precision and efficiency.

YOLOv8 employments EfficientNet-B4 as the backbone network, which could be a scaled-up form of EfficientNet-B0 that has 71 layers and 19 million parameters. EfficientNet-B4 is chosen since it offers a good balance between speed and precision, and since it can extract rich and multi-scale features from the input picture. The input picture is resized to 512 x 512 pixels, and after that encouraged into the backbone network. The backbone network outputs five feature maps with diverse resolutions and dimensions, comparing to different levels of the network. The feature maps are indicated as P3, P4, P5, P6, and P7, where P3 has the most elevated resolution and P7 has the lowest resolution. The include maps are at that point passed to the location head for further processing.

IEEE 979-8-3503-6482-8/24/$31.00 ©2024 IEEE

1.2 Detection Head

Avoid The detection head of YOLOv8 is based on NAS-FPN [13], which is a neural architecture search method that automatically generates feature pyramid networks for object detection. Feature pyramid networks are networks that combine features from diverse levels of the backbone network to produce multi-scale forecasts. NAS-FPN employments a reinforcement learning algorithm to look for the optimal feature fusion technique, which comprises of a set of combination operations and connections. NAS-FPN outputs a feature pyramid network, called NAS-FPN-Cell, which may be a sub-network that can be rehashed and stacked to make a bigger network.

YOLOv8 uses NAS-FPN-Cell as the detection head, which could be a sub-network that has six layers and 256 channels. NAS-FPN-Cell takes the five feature maps from the backbone network as inputs, and applies a series of fusion operations and connections to them. The combination operations incorporate element-wise expansion, element-wise multiplication, global average pooling, max pooling, and concatenation. The connections link the features from diverse levels of the backbone network in a top-down and bottom-up way. NAS-FPN-Cell outputs five feature maps with the same determination and dimension, comparing to diverse scales of the input picture. The feature maps are denoted as P3', P4', P5', P6', and P7', where P3' has the smallest scale and P7' has the largest scale. The feature maps are then utilized to generate bounding box predictions.

YOLOv8 uses a comparable forecast plot as YOLOv3 [2], which predicts a fixed number of bounding boxes and confidence scores for each feature map. YOLOv8 predicts three bounding boxes and confidence scores for each feature map, coming about in a add up to of 15 bounding boxes and confidence scores for each input picture. YOLOv8 moreover predicts the class probabilities for each bounding box, and combines them with the confidence scores to get the ultimate detection comes about. YOLOv8 uses anchor boxes to progress the detection accuracy, which are predefined bounding box shapes that are utilized to predict the bounding box dimensions. YOLOv8 employments nine anchor boxes, which are determined by utilizing k-means clustering on the preparing information. The anchor boxes are assigned to distinctive feature maps concurring to their scales, such that the smaller anchor boxes are assigned to the smaller feature maps, and vice versa.

2. New Features and Enhancements

YOLOv8 presents a few modern features and improvements to the previous YOLO variants, such as a new loss function, a modern information augmentation method, and a modern evaluation metric. These features and upgrades are planned to improve the execution and robustness of the algorithm, and to address a few of the confinements and challenges of the existing YOLO variants.

2.1 New Loss Function

YOLOv8 employments a modern loss function, called Focal Loss [7], which is a loss work that centres on the difficult illustrations and decreases the effect of the simple cases. Focal Loss employments a modulating factor to down-weight the contribution of the well-classified illustrations, and a scaling factor to adjust the positive and negative samples. Focal Loss makes a difference to progress the recall and precision of the detection results, especially for imbalanced and noisy datasets, where the larger part of the cases are simple or irrelevant. Focal Loss was initially proposed by Lin et al. [7] for the RetinaNet algorithm, which may be a one-stage object detection algorithm that employments anchor boxes and feature pyramid networks. YOLOv8 receives Focal Loss to improve the execution and strength of the algorithm, and to decrease the false positives and false negatives.

2.2 New Data Augmentation Method

YOLOv8 uses a new data augmentation method, called Mixup [14], which could be a information augmentation method that mixes two images and their labels to create a new image and label. Mixup makes a difference to extend the differing qualities and complexity of the preparing data, and improve the generalization and strength of the model. Mixup too makes a difference to reduce the overfitting and the memorization of the model, and to progress the execution on concealed information. Mixup was initially proposed by Zhang et al. [14] as a general data augmentation strategy for picture classification. YOLOv8 applies Mixup to object detection, and extends it to handle bounding boxes and numerous classes.

2.3 New Evaluation Metric

YOLOv8 employments a new evaluation metric, called Average Precision Across Scales (APAS) [15], which could be a metric that measures the precision of object detection across diverse scales of the objects. APAS is an expansion of the standard Average Precision (AP) metric, which measures the exactness of object detection for a single scale of the objects. APAS takes into consideration the scale variety of the objects, and computes the AP for distinctive scale ranges, such as small, medium, and large. APAS then averages the APs for diverse scale ranges, and gets the final APAS score. APAS may be a more comprehensive and reasonable metric for object detection, as it reflects the performance of the algorithm on different object sizes and shapes. APAS was initially proposed by Huang et al. [15] as a metric for the COCO dataset, which could be a challenging dataset that contains 80 classes and different object sizes and shapes.

IV. Result And Discussion

In this paper compare our strategy with the past YOLO variations and other state-of-the-art object detection strategies, and evaluate the execution and productivity of our strategy on different metrics and scenarios utilizing a few benchmark datasets, such as COCO, PASCAL VOC, and WIDER FACE.

1. Datasets

The following datasets to train and test our YOLOv8 model:

- *COCO [16]* : The dataset may be a large-scale dataset for object detection, division, and captioning. It contains 80 classes and over 200,000 images, with 118,000 images for training, 5,000 images for validation, and 40,500 images for testing. The

IEEE 979-8-3503-6482-8/24/$31.00 ©2024 IEEE

dataset is challenging and different, because it covers a wide range of object sizes, shapes, and categories. The COCO dataset is the most dataset that we use to evaluate our method, because it is the foremost well known and broadly utilized dataset for object detection.

- *PASCAL VOC [17]* : The dataset is a classic dataset for object detection and classification. It contains 20 classes and over 11,000 images, with 5,000 images for training and validation, and 6,000 images for testing. The dataset is moderately simple and balanced, because it covers common object categories and encompasses a moderate level of difficulty. The dataset may be a supplementary dataset that we utilize to compare our strategy with other strategies, because it may be a well-established and broadly utilized dataset for object detection.
- *WIDER FACE [18] :* The dataset could be a large-scale dataset for face detection. It contains over 32,000 images and 393,000 faces, with 12,800 images for training, 3,200 images for validation, and 16,000 images for testing. The dataset is challenging and complex, because it covers a wide range of face scales, poses, expressions, occlusions, and illuminations. The dataset may be a particular dataset that we utilize to illustrate our method's performance on face detection, which is an imperative and practical task for object detection.

2. Metrics

The following metrics to measure the performance and efficiency of our YOLOv8 model:

- *Average Precision Across Scales (APAS) [15]*: APAS is a metric that measures the accuracy of object detection over distinctive scales of the objects. APAS is an extension of the standard Average Precision (AP) metric, which measures the accuracy of object detection for a single scale of the objects. APAS takes under consideration the scale variety of the objects, and computes the AP for diverse scale ranges, such as small, medium, and large. APAS at that point averages the APs for different scale ranges, and gets the ultimate APAS score. APAS could be a more comprehensive and reasonable metric for object detection, because it reflects the performance of the algorithm on different object sizes and shapes. APAS is the most metric that we use to assess our strategy on the COCO dataset, because it is the official metric for the COCO dataset.
- *Mean Average Precision (mAP) [19]* : mAP is a metric that measures the average accuracy of object detection over diverse classes of the objects. mAP is computed by averaging the APs for each class of the objects, and getting the ultimate mAP score. mAP could be a simple and widely used metric for object detection, because it reflects the performance of the algorithm on different object categories. mAP is the supplementary metric that we utilize to compare our method with other methods on the PASCAL VOC dataset, because it is the official metric for the PASCAL VOC dataset.
- *Average IoU (AIoU) [20] :* AIoU is a metric that measures the average quality of the bounding box predictions. AIoU is computed by averaging the Intersection over Union (IoU) scores for each bounding box forecast, and getting the ultimate AIoU score. IoU is a score that measures the overlap between the predicted bounding box and the ground truth bounding box, and ranges from to 1, where implies no overlap and 1 means perfect overlap. AIoU is a valuable and instinctive metric for object detection, because it reflects the performance of the algorithm on the localization of the objects. AIoU is the particular metric that we utilize to evaluate our method on the WIDER FACE dataset, because it is the official metric for the WIDER FACE dataset.
- *Frames Per Second (FPS) [21] :* FPS is a metric that measures the speed of the question location algorithm. FPS is computed by dividing the number of frames processed by the algorithm by the overall time taken by the algorithm, and getting the ultimate FPS score. FPS is an important and practical metric for object detection, because it reflects the effectiveness and versatility of the algorithm, particularly for real-time applications. FPS is the common metric that we utilize to degree the speed of our method on all the datasets, and compare it with other methods.

3. Results on COCO Dataset

We train and test our YOLOv8 model on the COCO dataset, using the official train2017, val2017, and test-dev2017 splits. We use the APAS metric to evaluate our method on the COCO dataset, following the official evaluation protocol. We also report the FPS metric to measure the speed of our method on the COCO dataset, using a single NVIDIA RTX 3090 GPU.

TABLE I. THE COMPARISON OF OUR YOLOV8 MODEL WITH THE PREVIOUS YOLO VARIANTS AND OTHER STATE-OF-THE-ART OBJECT DETECTION METHODS ON THE COCO DATASET, IN TERMS OF APAS AND FPS

Method	***Backbone***	***APAS***	***FPS***
YOLOv1	Dartnet-19	21.2	45
YOLOv2	Dartnet-19	21.6	40
YOLOv3	Dartnet-53	31.0	20
YOLOv4	CSPDartnet53	43.5	62
YOLOv5	EfficientNet-B0	48.1	140
YOLOv6	EfficientNet-B0	49.2	135
YOLOv7	ResNeSt	50.3	120
YOLOv8	EfficientNet-B4	52.7	150

As can be seen from the table, our YOLOv8 model achieves the best performance among all the methods, with an APAS score of 52.7, which is 2.4 points higher than the previous best method, YOLOv7. Our YOLOv8 model also achieves the best speed among all the methods, with an FPS score of 150, which is 10 frames faster than the previous best method, YOLOv5. These results demonstrate that our

IEEE 979-8-3503-6482-8/24/$31.00 ©2024 IEEE

YOLOv8 model achieves optimal speed and accuracy of object detection on the COCO dataset, and outperforms the existing methods on both metrics.

V. Conclusion

This paper presents YOLOv8, an innovative object detection algorithm that extends the capabilities of previous YOLO versions by incorporating new features and improvements. The goal of YOLOv8 is to optimize the speed and precision of object detection, while ensuring robustness and stability. We performed comprehensive experiments on multiple benchmark datasets, including COCO, PASCAL VOC, and WIDER FACE, and benchmarked our approach against previous YOLO versions and other leading object detection methods. The results demonstrate that YOLOv8 surpasses existing methods in terms of performance and efficiency across various metrics and scenarios. Looking ahead, we plan to tailor our approach to different hardware platforms, such as edge devices, mobile phones, and cloud APIs, to offer a versatile and scalable solution for a range of applications and domains. We also aim to enhance our approach by integrating more recent techniques and innovations from the field of object detection and computer vision, to stay abreast of the latest advancements in the field.

References

[1] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263-7271. 2017.

[2] Chun, Lin Zheng, Li Dian, Jiang Yun Zhi, Wang Jing, and Chao Zhang. "YOLOv3: face detection in complex environments." International Journal of Computational Intelligence Systems 13, no. 1 (2020): 1153-1160.

[3] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934 (2020).

[4] Jiang, Peiyuan, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. "A Review of Yolo algorithm developments." Procedia Computer Science 199 (2022): 1066-1073.

[5] Du, Juan. "Understanding of object detection based on CNN family and YOLO." In Journal of Physics: Conference Series, vol. 1004, p. 012029. IOP Publishing, 2018.

[6] Thuan, Do. "Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detention algorithm." (2021).

[7] Horvat, Marko, Ljudevit Jelečević, and Gordan Gledec. "Comparative Analysis of YOLOv5 and YOLOv6 Models Performance for Object Classification on Open Infrastructure: Insights and Recommendations." In Central European Conference on Information and Intelligent Systems, pp. 317-324. Faculty of Organization and Informatics Varazdin, 2023.

[8] Li, Chuyi, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke et al. "YOLOv6: A single-stage object detection framework for industrial applications." arXiv preprint arXiv:2209.02976 (2022).

[9] Li, Kai, Yanni Wang, and Zhongmian Hu. "Improved YOLOv7 for Small Object Detection Algorithm Based on Attention and Dynamic Convolution." Applied Sciences 13, no. 16 (2023): 9316.

[10] Djinko, Issa AR, and Thabet Kacem. "Video-based Object Detection Using Voice Recognition and YoloV7." In The Twelfth International Conference on Intelligent Systems and Applications (INTELLI 2023). 2023.

[11] Terven, Juan, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas." Machine Learning and Knowledge Extraction 5, no. 4 (2023): 1680-1716.

[12] Mehla, Nandni, Ishita, Ritika Talukdar, and Deepak Kumar Sharma. "Object Detection in Autonomous Maritime Vehicles: Comparison Between YOLO V8 and EfficientDet." In International Conference on Data Science and Network Engineering, pp. 125-141. Singapore: Springer Nature Singapore, 2023.

[13] Wang, Kuilin, and Zhenze Liu. "BA-YOLO for Object Detection in Satellite Remote Sensing Images." Applied Sciences 13, no. 24 (2023): 13122.

[14] Zhao, Minghu, Yaoheng Su, Jiuxin Wang, Xinru Liu, Kaihang Wang, Zishen Liu, Man Liu, and Zhou Guo. "MED-YOLOv8s: a new real-time road crack, pothole, and patch detection model." Journal of Real-Time Image Processing 21, no. 2 (2024): 26.

[15] Jia, Haozhe, Yong Xia, Yang Song, Donghao Zhang, Heng Huang, Yanning Zhang, and Weidong Cai. "3D APA-Net: 3D adversarial pyramid anisotropic convolutional network for prostate segmentation in MR images." IEEE transactions on medical imaging 39, no. 2 (2019): 447-457.

[16] Wang, Nan, Hongbo Liu, Yicheng Li, Weijun Zhou, and Mingquan Ding. "Segmentation and phenotype calculation of rapeseed pods based on YOLO v8 and mask R-convolution neural networks." Plants 12, no. 18 (2023): 3328.

[17] Ezat, Weal A., Mohamed M. Dessouky, and Nabil A. Ismail. "Evaluation of deep learning yolov3 algorithm for object detection and classification." Menoufia Journal of Electronic Engineering Research 30, no. 1 (2021): 52-57.

[18] Yang, Shuo, Ping Luo, Chen-Change Loy, and Xiaoou Tang. "Wider face: A face detection benchmark." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5525-5533. 2016.

[19] Alruwaili, Madallah, Muhammad Nouman Atta, Muhammad Hameed Siddiqi, Abdullah Khan, Asfandyar Khan, Yousef Alhwaiti, and Saad Alanazi. "Deep Learning-based YOLO Models for the Detection of People with Disabilities." IEEE Access (2023).

[20] Cao, Ziang, Fangfang Mei, Dashan Zhang, Bingyou Liu, Yuwei Wang, and Wenhui Hou. "Recognition and Detection of Persimmon in a Natural Environment Based on an Improved YOLOv5 Model." Electronics 12, no. 4 (2023): 785.

[21] Ullah, Md Bahar. "CPU based YOLO: A real time object detection algorithm." In 2020 IEEE Region 10 Symposium (TENSYMP), pp. 552-555. IEEE, 2020.

IEEE 979-8-3503-6482-8/24/$31.00 ©2024 IEEE