Title : The Developments and Improvements Made to the Different Versions of the Yolo Algorithms: A Review Paper

Volume : 7

مجلة العلوم والتطـبيقات الهـندسية

Journal of Science and Engineering Application

ISSN 2521-3911

The Developments and Improvements Made to the Different Versions of the Yolo Algorithms: A Review Paper

Hassan Muhammad Hassan Alqurayshi
Department of Medical Laboratories
Ahl -Albayt University
Karbala, Iraq
Hassan13101992@abu.edu.iq

Abstract

From the first YOLO version to the last version, it was considered the most common, acceptance and speed object detection algorithm. YOLO is a single-step object detection algorithm since it performs both detection and classification at the same step. This paper focuses on and highlights on the key contributions of each version and the modifications that happened in each one. Most modifications tend to improve either the accuracy or speed of the model or to improve both of them for example by manipulation of the number of layers or by the innovation of new technologies such as Bag of Freebies or to innovation a new type of backbones. So, this paper reviews these approaches and how much they effect on the model performance by making a comparison between the different versions of YOLO algorithms.

Keywords—Deep Learning, Object Detection, Single Step algorithm, You Look Only Once, Anchor Boxes.

1. Introduction

Before delving into the research on the developments of the YOLO algorithm, we must first know that it is a single-step algorithm. Basically, object detection algorithms can be subdivided into: single-step algorithms in which detection and classification are done at the same step. The other one is a double-step algorithm in which Region Proposal Generation to Identify regions of interest (ROIs) likely to contain objects at the first step. And then in the second step Region Classification & Refinement to Classify each proposal and refine bounding box coordinates. A single-step in comparison with double-step algorithms is faster in speed but slightly lower in accuracy than double-step algorithms. YOLO algorithms and SSD (single shot detection) are the most common examples of single-step algorithms whereas Faster RCNN and Mask RCNN are the most common examples of the double-step algorithms. In this paper, we will deal with some kind of single-step algorithm which is known as YOLO and review the different versions of it and review the developments, enhancements, and also performance of each version.in the following section of this paper, we will dive firstly into what is YOLO, and then we will review each version and compare the performance of each one to discover the modifications in each version.

2. A LOOK AT THE YOLO RELEASES

Deep learning is the driving force behind modern object detection systems, enabling them to achieve high accuracy, speed, and robustness. By leveraging deep neural networks, object detection has evolved

from relying on handcrafted features to end-to-end learning systems capable of handling complex real-world scenarios. The computer vision task of object detection entails locating and identifying objects in an image or video[1]. It goes beyond classification by identifying the specific locations of items, usually with the use of bounding boxes, in addition to identifying what objects are present. Coordinates (such as x, y, width, and height) define each bounding box, which is then linked to a confidence score and a class name [1]. To accomplish object detection the algorithm performs a series of procedures starting with finding the position of the object in the image or frame by localization process and after that, the algorithm will classify the object by determining the class label for example if the object was (cat, car, human, horse) and then outputs the results. Recently object detection can be utilized in several tasks such as autonomous driving, medical imagining, and surveillance cameras. The YOLO series is considered one of the most popular object detection frameworks due to its simplicity and efficiency in comparison with the double-step object detection algorithm. In 2016 Redmon et al [2] published a YOLO paper that was considered a Promising and new approach, it is worth noting YOLO came from the phrase (You Look Only Once) which means the algorithm directly predicts bounding boxes and class probabilities from full images in one evaluation [2]. The Key Contributions of YOLO v1 Includes the following:

- YOLO v1 combines several processes (such as region proposal, classification, and refinement) into a single neural network, in contrast to conventional object detection techniques that use numerous stages. As a result, the procedure is quicker and more effective.
- With its remarkable 45 frames per second and competitive precision, YOLO v1 was created for real-time detection of objects.
- YOLO v1 reduces false positives in background regions by implicitly encoding contextual information about classes and their appearances because it examines the full image during training and inference.
- YOLO v1 provides a simple pipeline because the model divides the input image into an S×S grid. Each grid cell predicts B bounding boxes, confidence scores for those boxes, and C class probabilities. The final detections are obtained by applying a threshold to the confidence scores.

The Architectural aspect can be summarized as follows:

- YOLO v1 uses a convolutional neural network (CNN) inspired by GoogLeNet, with 24 convolutional layers followed by 2 fully connected layers [2].
- The model outputs a tensor of size S×S×(B×5+C), where: S×S is the grid size. B is the number of bounding boxes per grid cell. Each bounding box prediction consists of 5 values: (x,y,w,h, confidence) .C is the number of classes.

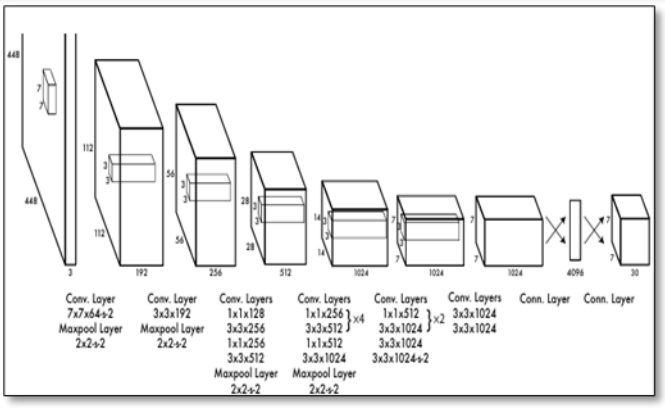

Fig. 1. Architecture of YOLO v1 [2]

Whereas YOLO v1 Performance can be listed as follows:

- YOLO v1 was designed for real-time object detection, achieving 45 frames per second (FPS) on a Titan X GPU[2]. A smaller version of YOLO called Fast YOLO, achieved an impressive 155 FPS but with reduced accuracy.
- YOLO v1 achieved a 63.4 mAP on the PASCAL VOC 2007 dataset [2]. On the PASCAL VOC 2012 dataset, it achieved 57.9 mAP[2].
- In comparison YOLO v1 was faster than other state-of-the-art detectors like R-CNN and Fast R-CNN, it had slightly lower accuracy compared to them. For example, Fast R-CNN achieved 70.0 mAP on PASCAL VOC 2007 but was significantly slower [2].

TABLE I. YOLO V1 IN COMPARISON WITH OTHER DETECTORS

Model	mAP (PASCAL VOC 2007)	Speed (FPS)	Strengths	Weaknesses
YOLO v1	63.4	45	Fast, simple, real-time	Lower accuracy struggles with small objects
Fast R-CNN	70.0	0.5	High accuracy	Slow, complex pipeline
Faster R-CNN	73.2	7	High accuracy, better localization	Slower than YOLO

Localization loss (for bounding box coordinates), confidence loss (for item existence), and classification loss (for class probabilities) are all combined into the loss function. The model is optimized by the use of sum-squared error [2].

One year after releasing the first version of the YOLO algorithm, the same authors release the second version of YOLO. Redmon et al [3] published the YOLO v2 or what they called (YOLO 9000) in 2017. The Improvements in YOLO v2 include: Added batch normalization to all convolutional layers, which improved convergence and reduced overfitting [3]. Fine-tuned the classification network on high-resolution images (448x448) before training on detection tasks. Introduced anchor boxes (inspired by Faster R-CNN) to predict bounding boxes relative to predefined priors, improving recall and localization. Used k-means clustering on training data to determine better anchor box priors, improving box predictions. Constrained bounding box predictions to prevent instability during training [3]. Added a passthrough layer to combine fine-grained features from earlier layers, improving detection of small objects. The second release of YOLO is faster than the previous one (YOLO v1) because YOLO v2 used a new backbone network called Darknet-19, a lighter and more efficient architecture compared to the one used in YOLO v1 [3]. Presented the YOLO9000 collaborative training system, which integrated datasets for classification and detection. Made it possible for the model to use hierarchical classification (using WordTree) to identify more than 9,000 item categories. The key contribution of YOLO v2 can be listed as follows :

- Anchor Boxes and Dimension Clusters: Improved localization and recall.
- Darknet-19: A lightweight and efficient backbone network.
- Joint Training (YOLO9000): Enabled detection of a large number of object categories by combining detection and classification datasets.
- Real-Time Performance: Maintained high speed while improving accuracy.

The Architectural aspect can be summarized as follows:

Darknet-19 Backbone:

- 19 convolutional layers with batch normalization and Leaky ReLU activations [3].
- 5 max-pooling layers to reduce spatial dimensions.
- Lightweight and efficient, designed for real-time performance [3].

Detection Head:

- Added convolutional layers on top of Darknet-19 to predict bounding boxes, objectness scores, and class probabilities [3].
- Used anchor boxes and dimension clusters for better localization.

Whereas YOLO v2 Performance can be listed as follows:

PASCAL VOC 2007:

- Achieved 76.8 mAP at 67 FPS.
- Outperformed YOLO v1 (63.4 mAP) and was competitive with Faster R-CNN (73.2 mAP) while being significantly faster [3].

COCO Dataset:

- Achieved 44.0 mAP on the COCO dataset, outperforming SSD and Faster R-CNN in terms of speed and accuracy trade-off [3].

Shown how to use the COCO (detection) and ImageNet (classification) datasets to discover more than 9,000 item categories. Despite lacking labeled detection data for numerous classes, achieved 19.7 mAP

on ImageNet detection tasks [3]. YOLO v2 set the stage for further advancements in the YOLO series, including YOLO v3, v4, and beyond. Its innovations, such as anchor boxes and joint training, became standard in object detection models [3].

Redmon et al [4] came back in 2018 to publish a new paper that was special for releasing YOLO v3 which is known for its speed and accuracy in detecting objects in images and videos. The authors of the study acknowledge in their relatively informal writing style that YOLOv3 is a collection of minor enhancements that add up to a big difference rather than a major breakthrough. Key points from the YOLOv3 paper are as follows:

- Architectural aspect: YOLOv3 makes use of Darknet-53, a Darknet variation with 53 convolutional layers. This network is more effective than ResNet-101 or ResNet-152 and more potent than the Darknet-19 that was previously employed in YOLOv2[4].
- YOLOv3, which resembles a feature pyramid network, predicts boxes at three different scales. This improves the model's ability to identify items of different sizes.
- Anchor boxes: YOLOv3 predicts bounding boxes by using dimension clusters as anchor boxes. The tensor for the four bounding box offsets, one objectness prediction, and eighty class predictions are N x N x [3*(4 + 1 + 80)] since it predicts three boxes at each scale [4].
- Class prediction: YOLOv3 uses independent logistic classifiers and binary cross-entropy loss for class prediction rather than softmax. This makes it possible to classify objects using multiple labels (one object might belong to numerous classes)[4].
- YOLOv3 offers several minor enhancements over YOLOv2, including enhanced feature extraction, multi-scale predictions, and a more effective network design.
- Performance: YOLOv3 is incredibly quick and precise. It is appropriate for real-time applications because it strikes a reasonable balance between speed and accuracy. With a mean average precision (mAP) of 33.0 at 51 ms on the COCO dataset, YOLOv3 is equivalent to SSD but three times quicker [4].

Bochkovskiy et al [5] in 2020 published a YOLO v4 paper. With an emphasis on attaining the best possible speed and accuracy for object identification tasks, the study introduces YOLOv4, an enhancement over the earlier YOLOv3 model. The paper comprises the following key contributions:

- Architectural aspect: CSPDarknet53 (Cross-Stage Partial connections) is used as the backbone for feature extraction.PANet (Path Aggregation Network) is employed for better feature aggregation [5]. YOLOv3 head is retained but optimized for better performance.
- Bag of Freebies (BoF): Techniques that improve accuracy without increasing inference time, such as Data augmentation (Mosaic, CutMix, etc.), Regularization (DropBlock, etc.), Loss function: CIoU loss [5].
- Bag of Specials (BoS): Techniques that slightly increase inference time but significantly improve accuracy, such as Mish activation function, Cross mini-Batch Normalization (CmBN), and Self-adversarial training (SAT) [5].
- Performance: YOLOv4 achieves state-of-the-art results on the MS COCO dataset with an AP of 43.5% (65.7% AP50) at a speed of 65 FPS on a Tesla V100 GPU [5].

YOLO v4 paper is a significant contribution to the field of object detection, providing a balance between speed and accuracy that is suitable for real-time applications. YOLOv4 is widely used in various fields such as autonomous driving, surveillance, and robotics due to its high speed and accuracy.

No formal YOLOv5 article has been presented at a conference or peer-reviewed journal. Ultralytics introduced YOLOv5, and the documentation, release notes, and blog articles of the repository are the main sources of information about the model. YOLOv5 has not adhered to the same academic publication process as YOLOv1 through YOLOv4, which was supported by official research publications. YOLO v5 consists of the following enhancements and updates:

- PyTorch, which is more popular and easier to use than Darknet (used in YOLOv4), is used to write YOLOv5.This facilitates integration with current workflows based on PyTorch.
- For the first time, YOLO v5 originates with different sizes small, medium, large, extra large, and also very light. Depending on their use case, users can choose between speed and accuracy with these variations.
- For improved generalization, YOLOv5 incorporates sophisticated data augmentation methods such as AutoAnchor, MixUp, and Mosaic augmentation.

Additionally, Hyperparameter Evolution is supported, which automatically adjusts hyperparameters for best results.

- With straightforward commands for training, validation, and inference, YOLOv5 is made to be easy to use. For platform deployment, it facilitates exporting models to multiple formats, such as ONNX, CoreML, and TensorRT.
- Due to variations in implementation and assessment parameters, YOLOv5 is not always directly comparable to YOLOv4, despite achieving competitive accuracy and speed on the COCO dataset[6].

YOLOv5 was developed by Ultralytics, a company focused on practical applications and open-source tools, rather than academic research.in comparison with the YOLO v4 Pytorch used in YOLO v5. In terms of speed and accuracy, YOLOv5 is comparable to YOLOv4, while the precise performance varies depending on the use case and implementation specifics [6].

A Chinese company called Meituan took the lead to release version six of YOLO algorithms in 2023, the following points are the most important features and improvements that the YOLO v6 has:

- YOLOv6 is designed to be used in the real world such as in robotics and autonomous cars where speed and accuracy are very important factors and necessary.
- A new backbone and a new neck design are comprised in YOLOv6 to guarantee better accuracy and speed. It uses RepVGG-style blocks for better speed-accuracy trade-offs [6].
- A new method used in YOLOv6 to make the detection process more simple that method known as Anchor-free and is also used to reduce the number of hyperparameters.
- Performance is enhanced and computational overhead is decreased by decoupling the detection head into classification and regression branches.
- To improve the accuracy without the need for any redundant labeled data, YOLOv6 follows self –distillations which means the model learns itself with its own predictions.
- YOLOv6 is made to be readily installed on a variety of hardware platforms, such as edge devices and GPUs.

For that reason, YOLOv6 outperforms the prior versions of YOLO algorithms in terms of accuracy and speed, and also in dealing with high-resolution images making it suitable for real-time applications [7]. With an emphasis on industrial applications, YOLOv6 is quicker and more precise. For improved performance, YOLOv6 provides architectural enhancements and takes an anchor-free approach. Like YOLOv5, YOLOv6 was created mostly for real-world uses rather than scholarly study. A strong and effective object detection model made for industrial use is called YOLOv6. With self-distillation

techniques, an anchor-free approach, and architectural enhancements, it expands upon the YOLO framework. Despite not having a formal research article, it is a useful tool for real-time object identification jobs due to its robust performance and open-source implementation.

Wang et al [8] publish in 2023 YOLOv7 that surpasses the other versions of YOLO. The features and contributions of YOLOv7 can be listed as follows:

- A novel backbone architecture that enhances feature learning and model scalability is the Extended Efficient Layer Aggregation Network (E-ELAN).Model Scaling: To optimize the model for varying computing budgets, YOLOv7 presents a compound scaling technique [8].
- The dynamic label assignment technique used by YOLOv7 during training enhances the model's capacity to handle objects of various forms and sizes.
- Bag of Freebies (BoF): YOLOv7 uses several training methods, including mosaic data augmentation, self-adversarial training (SAT), and CIoU loss for improved bounding box regression, that increase accuracy without lengthening inference time[8].
- With an AP of 56.8% for the YOLOv7-X model, YOLOv7 attains the highest accuracy of any real-time object detector (when it was first released) on the MS COCO dataset [8]. Compared to YOLOv4, YOLOv5, and YOLOv6, it is quicker and more accurate.
- YOLOv7 is appropriate for a variety of hardware platforms because it is optimized for both GPU and CPU inference. On high-resolution photos, it performs in real time.

In terms of speed and accuracy, YOLOv7 performs better than YOLOv6, emphasizing the advancement of real-time object identification. The innovative object detection model YOLOv7 raises the bar for accuracy and real-time performance. It makes it one of the most potent object detectors on the market by introducing several architectural and training advancements.

In the same year of YOLOv7 publishing, Ultralytics developed and released the YOLOv8 design to surpass the other ones, and the key feature involved the following:

- YOLOv8 provides a variety of pre-trained model sizes, including nano (n), small (s), medium (m), large (l), and extra-large (x). These models can be adjusted for specific tasks and are pre-trained using the COCO dataset [9].
- Advanced Training Features: Accommodates sophisticated methods of data augmentation.

Comprises functions such as mixed precision training, mosaic augmentation, and hyperparameter optimization.

- Options for Deployment: To be deployed on several platforms, Yolov8 models can be exported to many formats, including ONNX, TensorFlow, and TFLite [9].
- YOLOv8 expands on the developments of YOLOv5 and YOLOv7, even though the precise architectural elements are not documented in a formal paper. To increase feature extraction and prediction accuracy, it incorporates improvements to the head, neck, and backbone structures [9].
- Includes pre-trained models that may be adjusted using unique datasets.
- Supports tasks like picture categorization, instance segmentation, and object detection. Can be applied to many different tasks, such as video analysis and real-time object detection.
- YOLOv8 attains cutting-edge results in terms of inference speed and accuracy. Due to its real-time application optimization, it may be deployed on mobile platforms, drones, and edge devices.

Utilized in robotics, autonomous vehicles, and surveillance applications involving real-time object detection. Segmenting Instances for applications that call for object detection at the pixel level. Classification of Images for jobs that don't require object detection.

Analysis of Videos: for video object tracking.

Wang et al [10] released YOLOv9 in 2024, which included the following efforts and contributions: a theoretical examination of the architecture of deep neural networks from the standpoint of reversible function. Based on this analysis, the authors created Programmable Gradient Information (PGI) and auxiliary reversible branches, which produced outstanding outcomes [9]. The issue that deep supervision is limited to very deep neural network topologies is resolved by the planned PGI. As a result, it makes it possible to implement innovative lightweight architectures in everyday situations. To achieve a higher parameter utilization than the depth-wise convolution architecture, the generalized ELAN (GELAN) network just employs conventional convolution. Thus, it demonstrates the significant benefits of being precise, quick, and light. The YOLOv9's object identification performance on the MS COCO dataset, when combined with the suggested PGI and GELAN, significantly outperforms the current real-time object detectors in every way [9]. Introducing (PGI) to address the information bottleneck and adapt deep supervision to lightweight neural network topologies. Developing the useful and efficient neural network known as GELAN. In object detection tasks, GELAN has demonstrated robust and consistent performance over a range of convolution and depth parameters. It might be widely acknowledged as a model that works well with different configurations of inference. By merging GELAN and PGI, YOLOv9 has demonstrated a high level of competitiveness [9]. Compared to YOLOv9, the deep model's ingenious architecture enables it to lower the number of parameters by 49% and the number of calculations by 43%. Additionally, its Average Precision increase on the MS COCO dataset remains at 0.6% [9]. In terms of accuracy and efficiency, the created YOLOv9 model outperforms RT-DETR and YOLO-MS. Using conventional convolution to improve parameter consumption, it raises the bar for lightweight model performance [9].

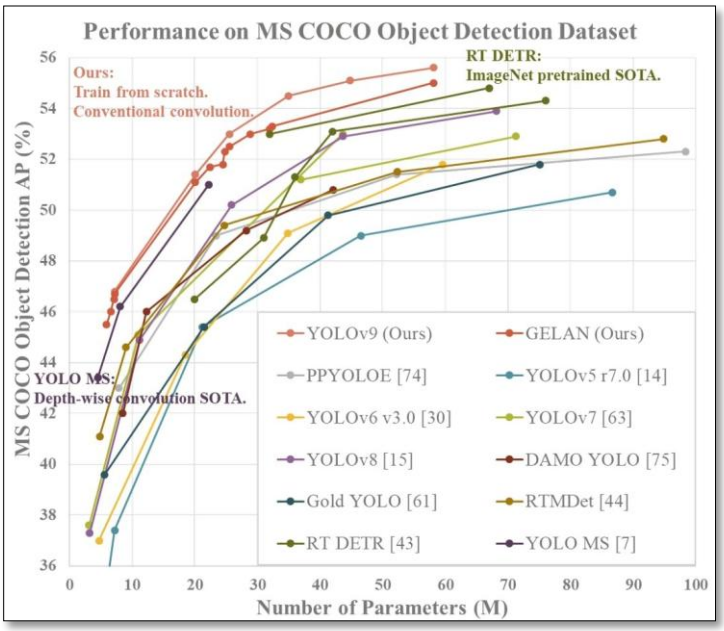

Fig. 2. Performance of YOLOv9 against other models [9]

YOLOv9 applications can be people counting, sports analytics, and Distribution and logistics since object detection can help determine product inventory levels to guarantee adequate stock levels and offer insights into customer behavior.

Wang et al[11] after a few months propose YOLOv10, because they strike a balance between detection performance and processing cost, YOLO models are widely used in real-time object identification. Although researchers have refined their designs, goals, and data techniques over time, end-to-end deployment is hampered and latency is increased when non-maximum suppression is used [11]. The capabilities of various YOLO components are limited by inefficiencies. With non-maximum suppression (NMS) free training for reduced latency and an efficiency-accuracy-driven design approach, YOLOv10 tackles these problems. Consistent dual assignments were presented by the authors for NMS-free training, which concurrently provides reduced inference latency and competitive performance [11]. Additionally, they suggested a comprehensive efficiency-accuracy-driven model design approach that optimized several YOLO components from an accuracy and efficiency standpoint. Performance is improved and computational overhead is decreased. According to experiments, YOLOv10 exhibits cutting-edge efficiency and performance. For instance, YOLOv10-S has fewer parameters and FLOPs than RT-DETR-R18 while being 1.8 times faster and having comparable accuracy. For the same performance, YOLOv10-B uses 25% fewer parameters and 46% less latency than YOLOv9-C [11]. Task Alignment Learning (TAL), which assigns several positive samples to each instance to improve optimization and performance, is commonly used for training YOLO models. Nevertheless, non-maximum suppression (NMS) post-processing is necessary, which lowers the efficiency of inference. Although one-to-one matching circumvents NMS, it results in inferior performance or inference overhead [11]. The authors present a consistent matching metric and dual label assignments as part of an NMS-free training approach. The conventional one-to-many head is combined with a secondary one-to-one head during training; both have the same optimization goals but employ distinct matching techniques. While the one-to-one head guarantees effective, NMS-free predictions during inference, the one-to-many head offers rich supervisory signals [11]. To save money, only the one-to-one head is used for inference. A consistent matching metric is employed to standardize the training procedure. This metric uses a consistent methodology that balances semantic prediction and location regression tasks to evaluate the concordance between predictions and examples for both one-to-many and one-to-one assignments [11]. The model optimizes both consistently by ensuring that the best positive samples for one head are also the best for the other by aligning the supervision from both heads. YOLO models' architecture, which might be computationally redundant and limited in capabilities, makes it difficult to strike a balance between accuracy and efficiency. To solve these problems, the authors suggest thorough model designs that prioritize accuracy and efficiency.

Model design influenced by efficiency:

- Spatial-channel decoupled down sampling: This technique separates spatial reduction and channel increase to lower computational expenses while preserving information.
- Lightweight classification head: This technique lowers computational overhead by employing a reduced architecture with depth-wise separable convolutions.
- Rank-guided block design: This technique replaces complex blocks with more effective structures, such as the compact inverted block, by using intrinsic rank analysis to find and eliminate redundancy in model stages.

Accuracy-based model construction:

- By expanding the receptive field in deep stages and selectively employing large-kernel depth wise convolutions to prevent overhead in shallow stages, large-kernel convolution improves model capability.
- Partial self-attention (PSA): This technique reduces memory use and computational complexity while improving global representation learning by dividing features and applying self-attention to a portion of the features.

YOLOv10 exhibits significant gains in AP when compared to the basic YOLOv8 models [10]. Additionally, YOLOv10 achieves notable latency reductions of 37% to 70% [10].

3. Conclusion

From the first version to the last one, the design of YOLO aims to make both detection and classification done with one step which makes it faster than RCNN, Faster-RCNN, or any other double-step models. From the aforementioned review will. YOLOv1 came with a novel approach which is an anchor box to improve the detection of the objects in the images this approach makes the YOLOv1 differ in terms of speed from other object detection algorithms. YOLOv1 achieved 63% mAP on the Pascal Voc 2007 Dataset with 45 frames per second and the number of parameters approximately 45 million parameters. For YOLOv2 The most notable improvements were adding batch normalization to all convolutional layers that enhance convergence and using the Darknet-19 backbone. YOLOv2 achieved 76.8% mAP on Pascal Voc 2007 and 44 % on the MS COCO dataset with 67 FPS and has 51 million parameters. YOLOv3 came after that with slight modifications such as using the Darknet-53 backbone and the multi-scale prediction that is just like a feature pyramid network (FPN). YOLOv3 achieved 78.1 % mAP on Pascal Voc 2007 and 33% mAP on the MS COCO dataset with 65.2 million parameters and 65 FPS. YOLOv4 came CSP Darknet-53 backbone that is used for feature extraction. YOLOv4 uses also a Bag of Freebies (BoF) and a Bag of Specials (BoS) to improve accuracy and inference time respectively 85.4% mAP on the Pascal Voc 2007 and 43.4% on MS COCO dataset with a number of parameters approximately 63.9 million and 62 FPS.YOLOv5 in turn doesn't differ so much from YOLOv4 but the important enhancement involves the implementation of a Pytorch frame to build the algorithm and YOLOv5 has come in different size range from small to large according to their weight. The following table illustrates the performance of YOLOv5.

TABLE II. YOLO V5 PERFORMANCE FOR THE THREE HIGHER WEIGHTS

model	mAP	Number of parameters	Frame per second
YOLO v5 m	45.5%	21 million	140
YOLO v5 L	49%	46.5 million	100
YOLO v5 X	50.7%	86.7 million	70

YOLOv6 involves a modification in its architecture the use of a RepVGG –style block for better accuracy. YOLOv6 employs a new technique known as self distillation which means the algorithm learns from itself predication. The following table illustrates the performance of the YOLOv6 algorithm.

TABLE III. YOLO V6 PERFORMANCE FOR THE THREE HIGHER WEIGHTS

model	mAP	Number of parameters	Frame per second
YOLO v6 m	49.5%	34.3 million	200
YOLO v6 L	52.8%	58.5 million	120
YOLO v6 X	54.1%	97.2 million	80

The Extended Efficient Layer Aggregation Network (E-ELAN) is a new backbone architecture that improves model scalability and feature learning used in YOLOv7. The following table illustrates the performance of the YOLOv7 algorithm:

TABLE IV. YOLO V7 PERFORMANCE FOR THE THREE HIGHER WEIGHTS

model	mAP	Number of parameters	Frame per second
YOLO v7 W	54.8%	97.2 million	84
YOLO v7 E	56%	151 million	56
YOLO v6 D	56.8%	212 million	42

YOLOv8 is designed for object detection and instance segmentation with small modifications to surpass the performance of YOLOv7. The following table illustrates the performance of the YOLOv8 algorithm:

TABLE V. YOLO V8 PERFORMANCE FOR THE THREE HIGHER WEIGHTS

model	mAP	Number of parameters	Frame per second
YOLO v8 m	50.2%	25.9 million	180
YOLO v8 L	52.9%	43.7 million	120
YOLO v8 X	53.9%	68.2 million	80

From Table (IV) and Table (V), we can conclude that YOLOv7 is more accurate than YOLOv8 but YOLOv8 is faster since it has less number of parameters or a number of layers.

The later versions such as YOLOv9 and YOLOv10 included modifications into architecture or path aggregation to improve either speed or accuracy or improve both of them. Its worth noting that YOLOv10 achieves notable latency reductions of 37% to 70%.

4. References:

[1] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, “Object Detection in 20 Years: A Survey,” Proc. IEEE, vol. 111, no. 3, 2023, doi: 10.1109/JPROC.2023.3238524.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "YOLOv1," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, 2016.K. Elissa, "Title of paper if known," unpublished.

[3] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017, vol. 2017-January. doi: 10.1109/CVPR.2017.690.Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[4]J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, [Online]. Available: https://arxiv.org/abs/1804.02767.

[5]A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020, [Online]. Available: https://arxiv.org/abs/2004.10934

[6]D. Yang, Y. Wang, H. Chen, E. Sun, and D. Zeng, "Feather Pecking Abnormal Behavior Identification and Individual Classification Method of Laying Hens Based on Improved YOLO v6 - tiny," Nongye Jixie Xuebao/Transactions Chinese Soc. Agric. Mach., vol. 54, no. 5, 2023, doi: 10.6041/j.issn.1000-1298.2023.05.028.

[7]M. Nain, S. Sharma, and S. Chaurasia, "Deep Learning-Based Safety Assurance of Construction Workers: Real-Time Safety Kit Detection," in Lecture Notes in Networks and Systems, 2023, vol. 680 LNNS. doi: 10.1007/978-981-99-2602-2_22.

[8]C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," 2023. doi: 10.1109/cvpr52729.2023.00721.

[9]G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, "UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios," Sensors, vol. 23, no. 16, 2023, doi: 10.3390/s23167190.

[10] C.-Y. Wang, I.-H. Yeh, and and H.-Y. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," vol. v1, 2024, [Online]. Available: https://arxiv.org/html/2402.13616v1

[11] A. Wang et al., "YOLOv10: Real-Time End-to-End Object Detection," in 38th Conference on Neural Information Processing Systems (NeurIPS 2024), 2024, vol. v2. [Online]. Available: 2405.14458v2

[12] A. Borji, M. M. Cheng, Q. Hou, H. Jiang, and J. Li, "Salient object detection: A survey," Computational Visual Media, vol. 5, no. 2. 2019. doi: 10.1007/s41095-019-0149-9.

[13] L. Jiao et al., "A survey of deep learning-based object detection," IEEE Access, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2939201.

[14] L. Liu et al., "Deep Learning for Generic Object Detection: A Survey," Int. J. Comput. Vis., vol. 128, no. 2, 2020, doi: 10.1007/s11263-019-01247-4.

[15] M. Nain, S. Sharma, and S. Chaurasia, "Deep Learning-Based Safety Assurance of Construction Workers: Real-Time Safety Kit Detection," in Lecture Notes in Networks and Systems, 2023, vol. 680 LNNS. doi: 10.1007/978-981-99-2602-2_22.

[16] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

Title : The Developments and Improvements Made to the Different Versions of the Yolo Algorithms: A Review Paper

Volume : 7

مجلة العلوم والتطـبيقات الهـندسية

Journal of Science and Engineering Application

ISSN 2521-3911

The Developments and Improvements Made to the Different Versions of the Yolo Algorithms: A Review Paper

Hassan Muhammad Hassan Alqurayshi
Department of Medical Laboratories
Ahl -Albayt University
Karbala, Iraq
Hassan١٣١٠١٩٩٢@abu.edu.iq
Hassan١٣١٠١٩٩٢@abu.edu.iq

Abstract

From the first YOLO version to the last version, it was considered the most common, acceptance and speed object detection algorithm. YOLO is a single-step object detection algorithm since it performs both detection and classification at the same step. This paper focuses on and highlights on the key contributions of each version and the modifications that happened in each one. Most modifications tend to improve either the accuracy or speed of the model or to improve both of them for example by manipulation of the number of layers or by the innovation of new technologies such as Bag of Freebies or to innovation a new type of backbones. So, this paper reviews these approaches and how much they effect on the model performance by making a comparison between the different versions of YOLO algorithms.