International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 12, ISSN No. 2455-2143, Pages 219-222
Published Online April 2024 in IJEAST (http://www.ijeast.com)

A COMPARATIVE STUDY ON DIFFERENT YOLO VERSIONS

Bhavana Baburaj, Aaron J Roy, Brite Sun George, Dr. Therese Yamuna Mahesh
Department of Electronics and
Communication
Amal Jyothi College of Engineering
Kottayam, India

***Abstract*— The last decade has seen a surge in the development of object detection techniques, particularly driven by the availability of vast amounts of data and advancements in Convolutional Neural Network (CNN) technology. Among the various CNN architectures, the You Only Look Once (YOLO) network has gained significant popularity due to its real-time object identification capabilities and simplicity. This review paper provides an overview of the YOLO algorithm and its subsequent versions, highlighting the ongoing improvements. We examine the differences and similarities among YOLO versions and between YOLO and other CNNs, evolution from the original YOLO to YOLOv8, YOLO-NAS, and YOLO with transformers. We discuss the major changes in network architecture and training techniques for each model, and summarize essential lessons learned from YOLO development. The applications of YOLO, particularly in robotics, driverless cars, video monitoring, UAV technology, medical object detection, and identify potential research direct object detection systems**

Keywords— YOLO, Bounding Box, IoU, SPPF

I. INTRODUCTION

In recent years, the field of computer vision has undergone a significant transformation, largely driven by the advent of Convolutional Neural Networks (CNNs) and groundbreaking algorithms such as You Only Look Once (YOLO). While humans possess an innate ability to instantaneously recognize and interact with visual stimuli, replicating this capability in computer systems has posed a significant challenge. However, advancements in CNNs, coupled with the development of algorithms like YOLO, have brought us closer to creating intelligent systems capable of performing complex tasks. These systems have the potential to predict driving crashes before they occur or identify specific objects in images in real-time, thereby enhancing safety and efficiency in various domains. This paper aims to delve into the development of YOLO and its diverse applications across a range of fields.

Computer vision systems, although adept at recognizing visuals, currently exhibit limitations in both accuracy and speed. Enhancements in efficiency and performance have the potential to elevate these systems to a level akin to human intelligence, facilitating the development of assistive technologies that enable humans to accomplish tasks effortlessly, with minimal conscious effort. Real-time object detection plays a pivotal role in automating or substituting human tasks, with computer vision and object detection emerging as prominent domains within machine learning. Recent technological progress has simplified the process of accessing and sharing data, with the proliferation of information and images on the internet/cloud becoming increasingly accessible. However, the recognition process remains challenging due to the unpredictable nature of factors such as item availability, location, size, and shape within each image. Additionally, variables like the complexity of backgrounds and variations in light intensity further complicate the recognition process. [1] [11].

II.KEY DEVELOPMENTS IN THE YOLO ALGORITHM

The YOLO (You Only Look Once) algorithm has undergone significant developments since its inception with YOLO V1. The original YOLO was distinguished by its focus on compact model size and rapid computational speed, facilitating real-time video detection. It directly produced bounding box positions and categories via the neural network, leveraging global image information for detection to encode overall context and minimize background detection errors. YOLO V1 showcased robust generalization capabilities by acquiring highly generalized features, thereby effectively transferring knowledge to other domains by framing target detection as a regression task.

With the introduction of YOLO V2, the algorithm addressed some of its limitations, particularly inaccuracies in positioning and a lower recall rate compared to area-based methods. YOLO V2 introduced improvements in convergence speed through batch normalization, a high-resolution classifier for enhanced accuracy, and the incorporation of fine features connecting feature maps of different scales. The implementation of multi-scale training further improved accuracy at different resolutions without sacrificing speed. YOLO V2 did not deepen or broaden the network but focused on simplifying it for efficiency.

YOLO V3 continued the evolution by adopting feature graphs of three scales for object detection. It used three prior boxes for each position, allowing for the detection of objects of varying sizes. The feature extraction network in YOLO V3 utilized a residual model with 53 convolution layers. The introduction of multi-scale features and adjustments to the basic network structure marked notable advancements in YOLO V3.

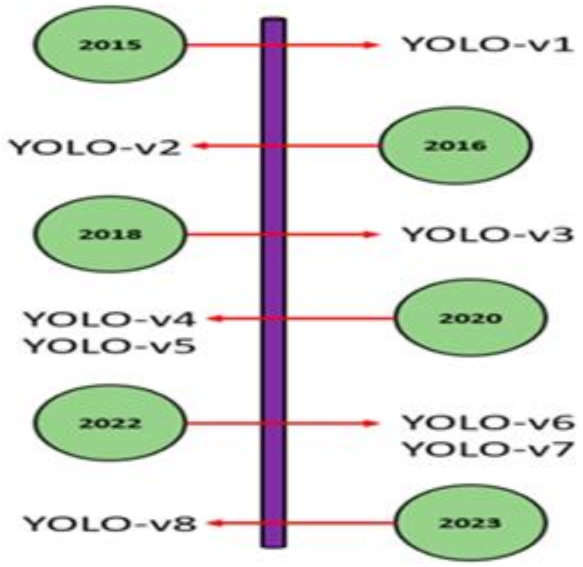

Fig 1.Evolution of YOLO Algorithms[13]

The YOLO V4 version marked a significant change, emphasizing a comprehensive exploration of optimization possibilities. It incorporated CSP Darknet53, SPP, Pan, and YOLO V3, addressing issues related to sample imbalance and introducing the CIOU loss function for bounding box regression. YOLO V4 demonstrated superior performance, running twice as fast as EfficientDet with comparable accuracy, showcasing its efficiency and advancements in the detection domain.

In the subsequent development of YOLO V5, the algorithm demonstrated enhanced flexibility, allowing for the control of model sizes from 10+M to 200+M. YOLO V5 leveraged the user-friendly PyTorch framework, making it easier to train datasets and integrate computer vision technologies. It introduced Mosaic enhancement to effectively address challenges in small object detection during model training.

YOLO V6 introduces key developments, including a single-stage object detection approach, a focus on hardware efficiency, and notable improvements in performance metrics compared to YOLOv5.

The YOLO v7 algorithm attains the utmost accuracy among all existing real-time object detection models, maintaining a frame rate of 30 FPS or more with a GPU V100. In contrast to the most efficient Cascade-Mask R-CNN models, YOLOv7 achieves a 2% boost in accuracy while significantly outpacing them in inference speed, showcasing a remarkable 509% improvement in speed.

YOLOV8's enhanced accuracy, when compared to earlier versions, is particularly valuable for precision-oriented tasks, such as medical imaging, where the precise detection of anomalies is critical. This improvement in accuracy contributes significantly to the algorithm's effectiveness in medical applications, supporting healthcare professionals in making accurate diagnoses based on detailed and reliable results. [3].

III. ARCHITECTURAL DIFFERENCES

The architecture of YOLOv1 comprises 24 convolutional layers succeeded by 2 fully connected layers, integrating leaky rectified linear unit activations, except for the final layer which employs a linear activation function. Drawing inspiration from Google Net and Network in Network, YOLO utilizes 1×1 convolutional layers for channel reduction. The output consists of a fully connected layer that produces a grid of 7×7 with 30 values assigned to each grid cell, accommodating ten bounding box coordinates (2 boxes) along with 20 categories.[10]

YOLOv2, also known as YOLO9000, introduces the Darknet-19 backbone with 19 convolutional layers and 5 max-pooling layers. Similar to YOLOv1, it utilizes 1×1 convolutions for parameter reduction and incorporates batch normalization for regularization.

The YOLOv3 architecture replaces max-pooling layers with strided convolutions and incorporates residual connections. It consists of 53 convolutional layers, each with batch normalization and Leaky ReLU activation. Additionally, YOLOv3 uses 1×1 convolutions with residual connections across the entire network.[9]

YOLOv4 introduces various modules, including CMB, CBL, UP, SPP, and PANet, for improved performance. It offers scaled-down (YOLOv4-tiny) and scaled-up (YOLOv4-large) versions, optimized for different hardware.

YOLOv5 adopts a modified CSPDarknet53 backbone and includes augmentations such as Mosaic, MixUp, and others. It offers five scaled versions (YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x) to suit specific applications and hardware requirements.[8]

YOLOv6 introduces a new backbone with RepVGG blocks, similar to YOLOv5 in terms of spatial pyramid pooling fast (SPPF) and Conv modules, but with a decoupled head for improved performance.

YOLOv7 introduces the Extended efficient layer aggregation network (E-ELAN) and a model scaling strategy for concatenation-based models. It utilizes planned re-parameterized convolution and includes a bag of freebies for enhanced performance. YOLOv8 features a modified

CSPDarknet53 backbone with a C2f module, incorporating a spatial pyramid pooling fast (SPPF) layer for accelerated computation. The head is decoupled to independently process objectness, classification, and regression tasks.

The architecture of YOLOv8 represents a progression from its predecessors, featuring a convolutional neural network divided into two primary components: the backbone and the head. The backbone is constructed on a modified version of the CSPDarknet53 architecture, incorporating 53 convolutional layers with enhanced cross-stage partial connections. Complementing this, the head encompasses multiple convolutional layers followed by fully connected layers, tasked with predicting bounding boxes, objectness scores, and class probabilities. Notably, YOLOv8 introduces a self-attention mechanism in the head of the network and integrates a feature pyramid network for multi-scaled object detection. This design enables YOLOv8 to selectively focus on different regions of an image and effectively detect objects of varying sizes and scales.[12]

The YOLO-NAS architecture is discovered through an automated neural architecture search (NAS) system, resulting in the generation of three distinct architectures (YOLO-NASS, YOLO-NASM, and YOLO-NASL), each differing in depth and the arrangement of QSP and QCI blocks.

ViT-YOLO integrates multi-head self-attention blocks with cross-stage partial-connection blocks within its MHSA-Darknet backbone. Feature aggregation in the neck is accomplished using BiFPN, while the head consists of five multi-scale detection heads. [4][5].

IV. HOW YOLO WORKS

The You Only Look Once (YOLO) algorithm revolutionizes object detection through its innovative grid-based approach. Initially, the input image undergoes partitioning into a grid with dimensions S × S. Each grid cell becomes the focal point for predicting a set of bounding boxes, ensuring a systematic examination of the entire image for potential object presence. This grid-based methodology lays the foundation for YOLO's efficiency and real-time capabilities.[7]

For each grid cell, YOLO predicts B bounding boxes, each characterized by critical parameters. The confidence score (Pc) reflects the algorithm's certainty regarding the presence of an object within the bounding box. Additionally, the coordinates of the box center (bx, by) and its dimensions (bh, bw) contribute to the comprehensive set of predictions. The output of YOLO takes the form of a tensor with dimensions S × S × (B × 5 + C), encapsulating confidence scores and bounding box parameters across different classes.

To refine its predictions and avoid redundancy, YOLO incorporates Non-Maximum Suppression (NMS) as a post-processing step. NMS selectively retains the most confident bounding boxes while suppressing those with significant overlap, resulting in a concise and accurate prediction set. This mechanism enhances the algorithm's precision in identifying and localizing objects within an image.

The confidence score calculation is a pivotal step in YOLO's workflow. Pc is determined by multiplying the probability (p) with the Intersection over Union (IOU) between the predicted and actual bounding boxes. This intricate calculation ensures that the confidence score not only considers the likelihood of object presence but also factors in the accuracy of the bounding box. The utilization of IOU contributes to the algorithm's reliability in assessing overlaps between predicted and ground truth bounding boxes.

YOLO operates on a regression-based algorithm, estimating probabilities and bounding boxes for all images. This regression approach allows the model to learn complex relationships between input features and output predictions, accommodating diverse object classes. Moreover, the distribution and placement of objects within the grid cells are carefully managed, with each cell associated with values indicating object presence, bounding box parameters, and category information.

The amalgamation of YOLO's grid-based methodology, bounding box predictions, NMS, confidence score calculation, regression-based approach, and grid distribution techniques collectively empowers the algorithm with efficiency and accuracy in real-time object detection tasks. This holistic approach has made YOLO a widely adopted solution across various domains requiring rapid and precise object detection capabilities.[2]

V. PERFORMANCE PARAMETERS

mAP (Mean Average Precision):

mAP serves as a prevalent metric for assessing the effectiveness of object detection algorithms. It amalgamates precision and recall values across various Intersection over Union (IoU) thresholds.

In YOLO, the model undergoes evaluation across different IoU thresholds to gauge its performance under diverse overlapping criteria.

Precision and Recall:

Precision denotes the ratio of correctly predicted positive instances to the total predicted positives, whereas recall represents the ratio of correctly predicted positive instances to the total actual positives. YOLO leverages precision and recall to analyze the balance between accurately detecting objects and mitigating false positives.

Intersection over Union (IoU):

IoU quantifies the overlap between the predicted bounding box and the ground truth bounding box. It is computed as the intersection area divided by the union area of the two boxes. YOLO employs IoU thresholds to ascertain whether a prediction qualifies as a true positive or false positive. Typically, this threshold is set to a value such as 0.5.

F1 Score:
The F1 score serves as the harmonic mean of precision and recall, offering a balanced evaluation of a model's performance. It is utilized to assess the overall effectiveness of the model, taking into account both precision and recall.

Confidence Score:
Confidence score represents the algorithm's confidence in its predictions. YOLO assigns confidence scores to each detected bounding box. The algorithm may consider bounding boxes with confidence scores above a certain threshold as positive detections

VI.CONCLUSION

In summary, the review provides a comprehensive exploration of YOLO's impact across diverse fields, emphasizing its real-time object detection capabilities and evolution through various versions. The advantages of YOLO, including its simplicity, parallel training, and superior generalization, position it as a leading algorithm in object detection. The future outlook is promising, with potential integrations of novel techniques and applications, showcasing YOLO's pivotal role in shaping the landscape of real-time object detection. Ethical considerations, particularly data privacy and algorithmic bias, are emphasized, indicating the importance of responsible development in healthcare settings. The integration of YOLO into healthcare applications is presented as a significant stride towards an AI-driven future that enhances medical processes and broadens access to quality healthcare. Ongoing research and development are identified as crucial for further optimizing YOLO's performance and addressing emerging challenges in real-time object detection within healthcare.[6].

VII. REFERENCE

[1]. Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, Bo Ma (2022)-“ A Review of Yolo Algorithm Developments ” ,pp(1066-1073)

[2]. Upulie,Lakshini Kuganandamurthy(2021)-“Real-Time Object Detection using YOLO: A review” , DOI:10.13140/RG.2.2.24367.66723

[3]. Juan Terven ,ORCID,Diana,Margarita Córdova,Esparza ORCID and Julio-Alejandro Romero-González (2023) -“ A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS”

[4]. Chunling Chen ,Ziyue Zheng ,Tongyu Xu ,Shuang Guo ,Shuai Feng ,Weixiang Yao ,ORCID and Yubin Lan (2023)-“ YOLO-Based UAV Technology: A Review of the Research and Its Applications”

[5]. Tausif Diwan, G. Anirudh & Jitendra V. Tembhurne(2022) -“ Object detection using YOLO: challenges, architectural successors, datasets and applications” ,pp(9243-9275)

[6]. Rizwan Qureshi , Mohammed Gamal Ragab, Said Jadid Abdulkader,Amgad muneer , Alawi Alqushaib , Ebrahim Hamid Sumiea , and Hitham Alhussian(2023)-,“ A Comprehensive Systematic Review of YOLO for Medical Object Detection”,DOI: 10.36227/techrxiv.23681679.v1

[7]. Eko Prasetyo, Nanik Suciati, Chastine Fatichah(2020)- A Comparison of YOLO and Mask R-CNN for Segmenting Head and Tail of Fish

[8]. Midhun P. Mathew,Therese Yamuna Mahesh(2021)-Leaf-based disease detection in bell pepper plant using YOLO v5

[9]. Therese Yamuna Mahesh ,Midhun P. Mathew (2023)-Detection of Bacterial Spot Disease in Bell Pepper Plant Using YOLOv3, DOI: 10.1080/03772063.2023.2176367

[10]. Joseph Redmon,Santosh Divvala,Ross Girshick(2016)- You Only Look Once: Unified, Real-Time Object Detection, DOI:10.1109/CVPR.2016.91

[11]. Tausif Diwan,G Anirudh,Jitendra V.Tembhurne (2022), Object detection using YOLO: challenges, architectural successors, datasets and applications,pp(9243–9275)

[12]. Muhammad Hussain (2023), YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection

[13]. Machines | Free Full-Text | YOLO-v1 to YOLO-v8, “the Rise of YOLO and Its Complementary Nature ,toward Digital Manufacturing and Industrial Defect Detection “.