Enhanced Real-Time Object Detection using YOLOv7 and MobileNetv3

Sara Ennaama
SIGL LAB., ENSA of Tetouan, Abdelmalek Essaadi University, Tetouan, Morocco
sara.ennaama@etu.uae.ac.ma (corresponding author)

Hassan Silkan
Department of Computer Science, Laboratory LAROSERI, Faculty of Sciences, University of Chouaib Doukkali, El Jadida, Morocco
silkan_h@yahoo.fr

Ahmed Bentajer
SIGL LAB., ENSA of Tetouan, Abdelmalek Essaadi University, Tetouan, Morocco
abentajer@uae.ac.ma

Abderrahim Tahiri
SIGL LAB., ENSA of Tetouan, Abdelmalek Essaadi University, Tetouan, Morocco
t.abderrahim@uae.ac.ma

Received: 20 August 2024 | Revised: 12 October 2024 and 13 October 2024 | Accepted: 16 November 2024

 DOI: https://doi.org/10.48084/etasr.8777

ABSTRACT

Object detection serves as a crucial element in computer vision, increasingly relying on deep learning techniques. Among various methods, the YOLO series has gained recognition as an effective solution. This research enhances object detection by merging YOLOv7 with MobileNetv3, known for its efficiency and feature extraction. The integrated model was tested using the COCO dataset, which contains over 164,000 images across 80 categories, achieving a mAP score of 0.61. Additionally, confusion matrix analysis confirmed its accuracy, especially in detecting common objects such as 'person' and 'car' with minimal misclassifications. The results demonstrate the potential of the proposed model to address the complexities of real-world scenarios, highlighting its applicability in various scientific and industrial domains.

Keywords-real-time object detection; deep learning; YOLOv7; MobileNetv3; computer vision

I. INTRODUCTION

Detecting objects is a key function in computer vision, supporting various applications such as autonomous driving, healthcare, security systems, and facial recognition. These scenarios often require fast and accurate detection in real-time, especially in environments with limited computational resources, including mobile and embedded systems. Initial methods such as the Viola-Jones algorithm [1] and the Histogram of Oriented Gradients (HOG) technique [2] laid the foundation for object detection. However, these approaches faced challenges with handling scale variations and delivering real-time performance, which made them less suitable for applications requiring rapid detection. The rise of deep learning brought about a major transformation in object detection through the use of Convolutional Neural Networks (CNNs). The R-CNN (Region-based CNN) family [3] significantly improved detection accuracy by employing a dual-phase approach which initially identifies regions of interest before classification. Despite additional enhancements in approaches such as Fast R-CNN [4] and Faster R-CNN [5], they remained computationally intensive, making real-time applications on devices with limited processing power difficult.

To address the demand for quicker and more efficient detection, one-stage detection models such as the Single Shot Detector (SSD) [6] and RetinaNet [7] were introduced. These models bypassed the region proposal step, resulting in faster detection times. However, they faced challenges in balancing speed with accuracy, particularly when detecting smaller objects in complex environments. The YOLO (You Only Look Once) series [8] introduced an innovative technique by analyzing the full image in one forward sweep through the network, which enabled real-time detection capabilities. Although the initial version, YOLOv1, achieved processing speeds of up to 45 fps, it faced challenges in accurately detecting smaller objects. Subsequent versions, YOLOv2 [9] and YOLOv3 [10], introduced multiscale predictions and more sophisticated backbone architectures such as Darknet-53, which improved detection performance. YOLOv4 [11] and YOLOv5 [12] brought in additional strategies, such as mosaic augmentation and Cross-Stage Partial (CSP) connections, to further enhance both speed and accuracy. However, the significant computational resources required by these models continued to restrict their usability on devices with limited processing capabilities. YOLOv7 [13] marked a notable advancement by incorporating features such as convolution reparameterization and efficient long-range attention mechanisms (E-ELAN), resulting in an improved trade-off between speed and detection precision. Despite these enhancements, its computational demands still make it difficult to deploy in mobile and embedded systems.

To overcome these limitations, MobileNet architectures emerged as lightweight alternatives suitable for real-time applications. MobileNetV1 [14] introduced depthwise separable convolutions, effectively reducing computational complexity, while MobileNetV2 [15] improved efficiency with the addition of inverted residuals and linear bottlenecks. MobileNetV3 [16], optimized through Neural Architecture Search (NAS), further enhanced performance using Squeeze-and-Excitation (SE) blocks and the H-swish activation function, making it particularly suitable for integration into systems that require real-time detection.

Previous efforts to combine YOLO models with MobileNet architectures yielded efficiency improvements but did not achieve high accuracy in real-time applications. In contrast, MobileNetv3 strikes an ideal balance between lightweight design and effective feature extraction, making it an excellent candidate for integration with YOLOv7 to address challenges in computational performance and precision. This study introduces the integration of YOLOv7 with MobileNetV3, aiming to develop an object detection model that is tailored for real-time use on mobile and embedded platforms. By merging YOLOv7's advanced detection capabilities with MobileNetV3's efficient architecture, the proposed model delivers strong accuracy without compromising speed and resource efficiency. The key innovation of this study lies in achieving cutting-edge performance with lower computational demands, rendering it a practical solution for real-time object detection applications.

II. MATERIALS AND METHODS

A. Coco Dataset

The COCO (Common Objects in Context) dataset [17], a well-known and extensive resource for tasks involving object detection, image segmentation, and caption generation, was used in this study. Developed by Microsoft, this dataset provides a diverse collection of realistic images representing various real-world scenarios, making it an excellent tool for evaluating the effectiveness of object detection models. This study employed the 2017 version of the COCO dataset, which contains more than 164,000 images, all annotated with 80 distinct object categories. These categories cover a broad spectrum of objects, from everyday household objects and animals to vehicles. The dataset is organized into several subsets, with approximately 118,000 images allocated for training, 5,000 for validation, and another 20,000 for testing. These detailed annotations include bounding boxes and instance segmentation masks, which facilitate accurate object localization. What sets the COCO dataset apart is its complexity and diversity, as it contains images with multiple objects of varying sizes and degrees of occlusion. This diversity makes it an ideal benchmark for assessing the adaptability and effectiveness of object detection algorithms, such as the integrated YOLOv7-MobileNetV3 model used in this study. The COCO dataset is offered under the CC BY 4.0 license [18], which allows reuse and modification, provided appropriate credit is given. This study used the COCO dataset to train and validate the proposed object detection model.

B. MobileNetv3

MobileNetv3 [16] represents a pivotal development in the evolution of deep learning architectures, emerging from a comprehensive Network Architecture Search (NAS). This model incorporates key features from its predecessors, utilizing depth-wise separable convolutions from MobileNetv1 [14] while embracing the linear bottleneck and residual setups found in MobileNetv2 [15]. A notable enhancement in MobileNetv3 is the integration of Squeeze-and-Excitation (SE) blocks within its bottleneck structures, elevating both its operational efficiency and effectiveness. Furthermore, it introduces an improvement by substituting the conventional swish activation with the h-swish activation function, showcasing a crucial advancement in refining neural network designs. The switch from swish to h-swish is motivated by practical considerations. Sigmoid computations, which are essential to the swish function, tend to be computationally demanding, especially on mobile devices with limited resources. In contrast, h-swish serves as a more efficient alternative to sigmoid, making it ideal in situations where fast processing is essential. Additionally, MobileNetv3 incorporates the ReLU activation function, recognized for its adaptability. ReLU is widely supported across various software and hardware platforms, maintains accuracy during quantization, and performs effectively within deep neural network architectures. These attributes make ReLU a reliable option in MobileNetv3's design, enhancing its overall efficiency and robustness.

MobileNetv3 is specifically designed for computer vision applications, emphasizing streamlined object detection and image classification. Its design strategically balances computational efficiency with accuracy, enabling it to perform at a high level while consuming fewer computational resources. This efficiency makes MobileNetv3 exceptionally suitable for applications in environments with limited computational resources. Equation (1) provides the details of the swish function, while (2) outlines the h-swish formula, emphasizing the network's adaptability to diverse computational environments and its commitment to maintaining high precision in challenging scenarios.

$$swish\, x = x \cdot \delta(x) \tag{1}$$

where x represents the function's input, and $\delta(x)$ denotes the sigmoid function applied to x.

$$h - swish = x \cdot [\frac{ReLU6(x+3)}{6}] \tag{2}$$

where x is the argument of the function. In this equation, $ReLU6(x)$ represents a rectified linear unit function with a maximum output capped at 6, ensuring boundedness within a specific range.

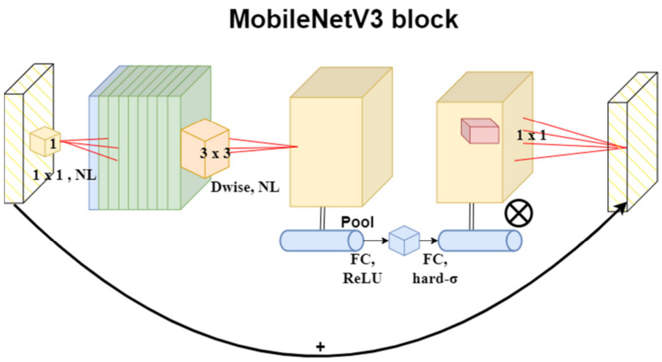

Fig. 1. Principal architectural structure of MobileNetv3.

TABLE I. MOBILENETV3 NETWORK PARAMETER INFORMATION

Input size	Operation	Expansion size	Output Channels	SE Module	Activation	Stride
50176 ×3	Convolution 2D	Not Applicable	16	No	HS	2
12544 ×16	Bottleneck, 3x repeats	16	16	No	RE	1
1254× 16	Bottleneck, 3x repeats	64	24	No	RE	2
3136× 24	Bottleneck, 3x repeats	72	24	No	RE	1
3136× 24	Bottleneck, 5x repeats	72	40	Yes	RE	2
784× 40	Bottleneck, 5x repeats	120	40	Yes	RE	1
784× 40	Bottleneck, 5x repeats	120	40	Yes	RE	1
784× 40	Bottleneck, 3x repeats	240	80	No	HS	2
196× 80	Bottleneck, 3x repeats	200	80	No	HS	1
196× 80	Bottleneck, 3x repeats	184	80	No	HS	1
196× 80	Bottleneck, 3x repeats	184	80	No	HS	1
196× 80	Bottleneck, 3x repeats	480	112	Yes	HS	1
196× 112	Bottleneck, 3x repeats	672	112	Yes	HS	1
196× 112	Bottleneck, 5x repeats	672	160	Yes	HS	2
49× 160	Bottleneck, 5x repeats	960	160	Yes	HS	1
49× 160	Bottleneck, 5x repeats	960	160	Yes	HS	1
49× 160	Convolution 2D, 1x1	Not Applicable	960	No	HS	1
49× 960	Pooling 7x7	Not Applicable	Not Applicabl e	No	None	1
1×960	Convolution 2D, 1x1, NBN	Not Applicable	1280	No	HS	1
1× 1280	Convolution 2D, 1x1, NBN	Not Applicable	k	No	HS	1

The MobileNetv3 block diagram in Figure 1 illustrates the sequence of operations that contribute to the network's efficiency. Beginning with a 1×1 convolution for channel-wise feature recalibration, the process then moves to a depthwise 3×3 convolution (Dwise), which is crucial for spatial feature extraction while maintaining low computational cost. The SE blocks, indicated where applicable, further refine the feature maps by dynamically adjusting features across channels, greatly enhancing the network's representational power. In addition to the diagram, Table I offers a detailed summary of the MobileNetV3 architecture. It presents each stage of the network, specifying the input size, type of operation, expansion size for bottleneck layers, the number of output channels, along with the presence of the SE block, the activation function used, and the stride. This table effectively encapsulates the design elements of MobileNetV3, highlighting its aim to achieve both computational efficiency and model accuracy.

C. YOLOv7

The YOLOv7 model [13], an improved version of the YOLO object detection algorithm series, introduces several advanced techniques to achieve an optimal equilibrium between detection precision and operational performance. This equilibrium is reached by incorporating innovative elements such as convolution reparameterization [19], scaling using concatenation-based models, and the Extended Efficient Long-range Attention Network (E-ELAN) [20]. Figure 2 illustrates how YOLOv7 retains the fundamental principles of YOLO detection while building on the groundwork established by earlier versions, YOLOv4 and YOLOv5.

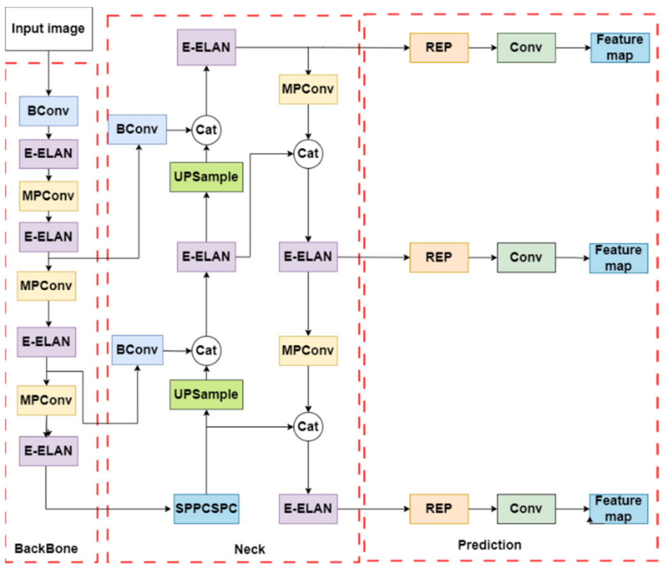

Fig. 2. The architecture of the original YOLOv7 network.

YOLOv7 is organized into four primary components: input, backbone, head, and prediction, each meticulously crafted to deliver optimal performance. The input component adjusts the size of incoming images to fit the requirements of the backbone, which comprises convolutional layers, including E-

ELAN and MPConv [13], with particular emphasis on the BConv [13] segment. This section combines a convolution layer with Batch Normalization (BN) and the LeakyReLU activation function [21], which is essential for effectively capturing features at multiple scales.

At the core of YOLOv7, the head module utilizes a Feature Pyramid Network with Path Aggregation (PAFPN) [22] to combine features from different depths, improving the model's capacity to analyze images at various scales. The prediction module, the final component, is responsible for refining the channel counts for features P3, P4, and P5, optimized with the REP block [19]. This results in the use of a 1×1 convolution layer specifically designed to accurately predict confidence scores, object categories, and anchor box details.

The detection of an input image requires a model that can deliver both real-time processing and high accuracy. Recognizing the impressive equilibrium between detection precision and speed, YOLOv7 was chosen as the foundational model to meet these demanding requirements.

D. Improving Yolov7 Using MobileNetv3

YOLO is widely recognized for its outstanding speed, accuracy, and high-quality performance, making it one of the leading approaches in object detection. YOLOv7, an advanced deep-learning version, builds on this reputation by performing object detection on images in just one forward pass, enabling it to identify multiple objects simultaneously while providing precise bounding-box coordinates. This blend of speed and accuracy sets YOLOv7 apart from many other object detection models, making it the model of choice for a wide range of real-time applications [13].

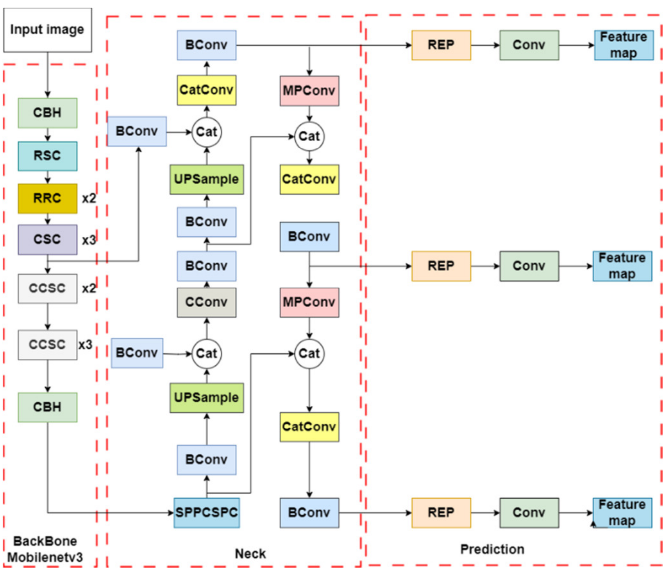

Fig. 3. The network structure of the MobileNetv3-YOLOv7 model.

Incorporating MobileNetv3 into YOLOv7 represents a crucial modification at the heart of the object detection framework. Within deep learning for object detection, the backbone structure is a key component in determining the model's performance. MobileNetv3 was selected to replace YOLOv7's initial backbone, tasked with the critical roles of processing incoming images and extracting vital features. This decision to replace the original backbone with MobileNetv3 is driven by several compelling reasons. MobileNetv3 is known for its distinct advantages, including computational efficiency, minimal memory footprint, and exceptional suitability for real-time and resource-constrained scenarios. This efficiency is achieved through the utilization of depth-wise separable convolutions, which significantly minimize the computational load while preserving feature extraction capabilities. Moreover, MobileNetv3 uses efficient building blocks, allowing an optimal balance between model accuracy and computational resources, a crucial consideration for real-time applications.

By adopting MobileNetv3 as the new backbone, the YOLOv7 model harnesses the benefits of both YOLOv7's object detection prowess and MobileNetv3's efficiency. The result is a hybrid model that achieves exceptional speed and accuracy in object detection while optimizing resource utilization. This integration exemplifies the dynamic nature of deep learning, where innovative combinations of architectures enable the development of models that meet the diverse demands of modern computer vision applications. Whether it is in autonomous vehicles, surveillance systems, or robotics, the potential for achieving outstanding performance in real-time object detection tasks while maintaining compatibility with resource-constrained platforms, such as embedded devices, is demonstrated.

III. RESULTS AND DISCUSSION

A. Model Evaluation Criteria

This study evaluated the effectiveness of the MobileNetv3-YOLOv7 architecture for object detection using several key metrics, including precision (P), F1 score, mean average precision (mAP), and recall (R) [23]. The mAP is determined through an Intersection over Union (IoU) threshold of 0.5 as the main criterion for evaluation. Precision is measured according to (3), recall is determined by (4), F1 score is computed using (5), and the method for calculating mAP is described in (6).

$$Precision = \frac{T_p}{(T_p + F_p)} \tag{3}$$

$$Precall = \frac{T_p}{(T_p + F_N)} \tag{4}$$

$$F_1 = \frac{2T_p}{(2T_p + F_N + F_P)} \tag{5}$$

$$mAP = \frac{\Sigma AP}{n} \tag{6}$$

where T_p (True Positives) denotes correct detections, F_p (False Positives) refers to instances where non-objects are mistakenly classified as objects, and F_N (False Negatives) describe instances where actual objects fail to be recognized. Calculating the AP (Average Precision) involves determining the region beneath the curve generated by mapping precision (P) values against recall (R), with n representing the number of unique object categories.

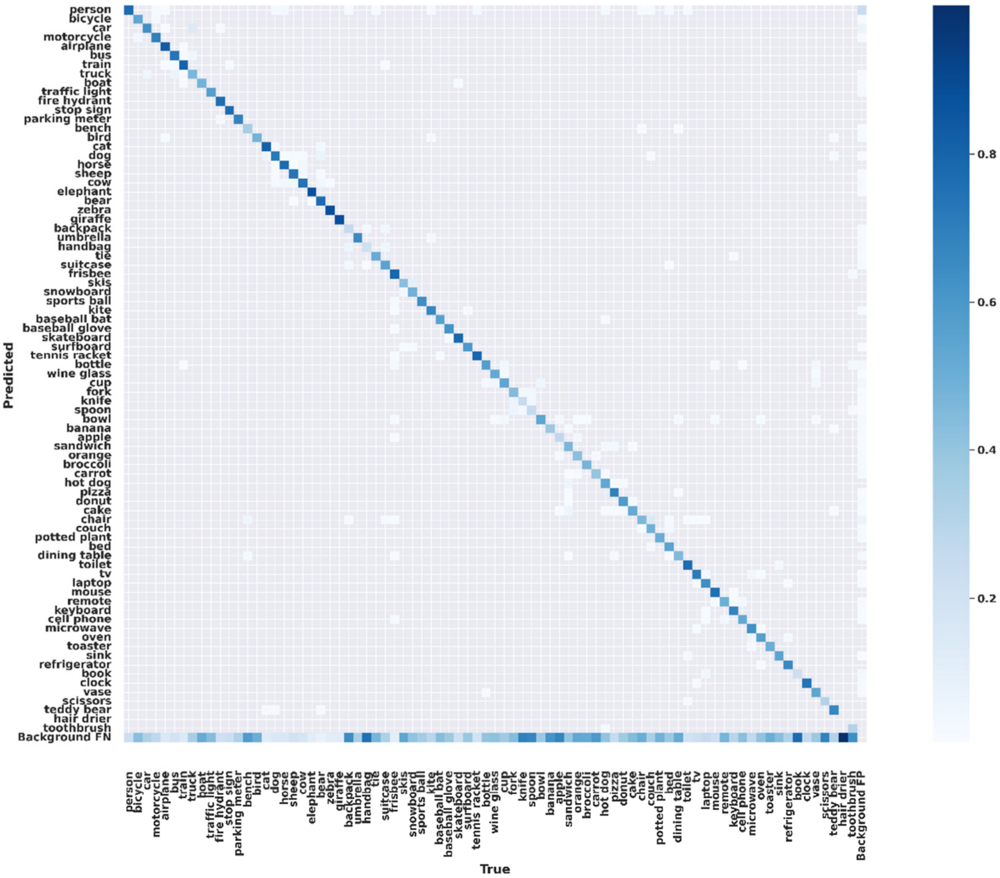

Fig. 4. Confusion matrix.

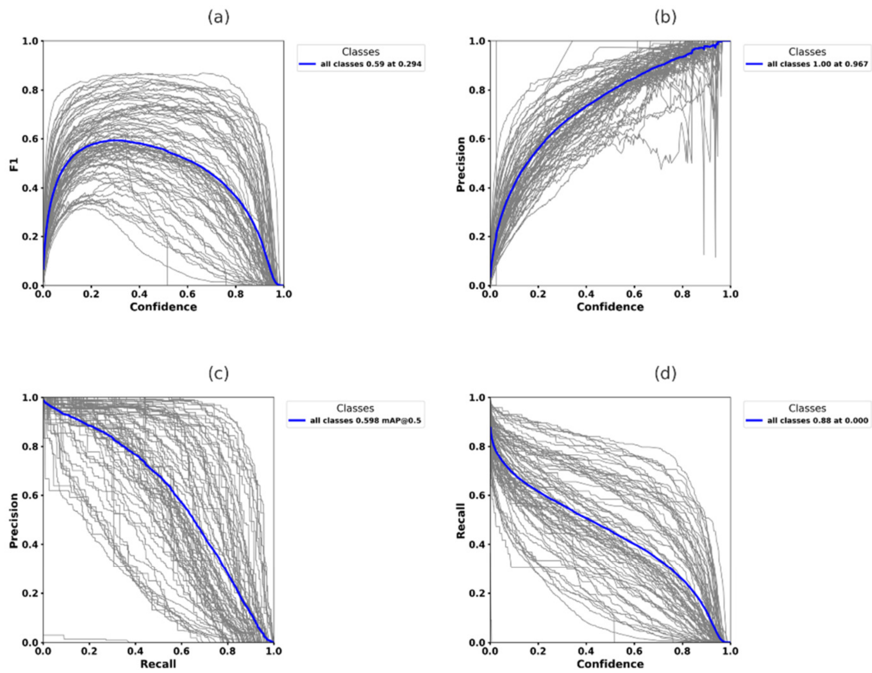

Fig. 5. (a) F1 score curve, (b) Precision plot, (c) Precision-Recall Graph, (d) Recall plot.

B. Results and Analysis

The MobileNetv3-YOLOv7 model was trained and tested on an online platform using the PyTorch framework on a server powered by an NVIDIA RTX 3090 GPU with 24 GB of RAM and CUDA 11.2 for parallel computing. Image inputs were standardized to an image resolution of 640×640 pixels, paired with a batch size of 16, carefully selected to balance training efficiency with memory limitations. A minimal confidence threshold of 0.001 was applied to filter out detections, while an IoU threshold of 0.65 was used to ensure accurate bounding box predictions, optimizing the equilibrium between accuracy and processing speed. The effectiveness of the proposed model was evaluated on the COCO dataset, and a detailed examination of the results was performed to evaluate the influence of multiple factors, including object size and scene complexity, on detection accuracy and efficiency. The confusion matrix (Figure 4) confirmed that the model accurately detected frequently occurring objects such as 'person,' 'car,' and 'bicycle,' with minimal misclassifications, showcasing robustness in real-world applications.

The MobileNetv3-YOLOv7 model exhibited strong adaptability in handling varying object sizes. It achieved high precision for larger objects while also showing improved detection rates for smaller objects compared to earlier YOLO versions. This enhanced accuracy for smaller objects can be attributed to MobileNetv3's efficient feature extraction capabilities, which helped the model differentiate finer details even in challenging situations. The model also maintained robust detection accuracy in scenes with high object density or complex backgrounds, effectively distinguishing objects even in cases of overlap or occlusion. Further analysis using Precision-Recall curves (Figure 5) indicated that the MobileNetv3-YOLOv7 model achieved a peak F1 score of 0.59, demonstrating its ability to maintain high precision as recall increased. The model attained a mAP of 0.599 at an IoU threshold of 0.5, closely aligning with the final reported mAP, which verifies the model's detection performance across multiple object categories. For a comprehensive comparison, Table II outlines the effectiveness of the MobileNetv3-YOLOv7 model alongside other leading models on the COCO dataset.

TABLE II. COMPARATIVE EFFECTIVENESS OF OBJECT DETECTION MODELS ON THE COCO DATASET BASED ON PUBLISHED BENCHMARKS AND EXPERIMENTAL RESULTS

Model	mAP @0.5	mAP @0.5:0.95	Unique characteristics
YOLOv4	0.495	0.33	Balanced accuracy and speed; relatively heavy model
YOLOv5	0.50	0.35	Improved accuracy and speed; more lightweight than YOLOv4
YOLOv7	0.556	0.39	State-of-the-art accuracy with efficient architecture
MobileNetv3 -YOLOv7	0.607	0.435	High accuracy and efficiency; lightweight architecture for embedded devices

The comparison shows that the MobileNetv3-YOLOv7 model outperformed other models in accuracy, achieving a mAP@0.5 of 0.607 while maintaining efficiency. This performance demonstrates the model's potential for deployment across various practical real-time scenarios, especially in resource-constrained environments. The integration of MobileNetv3 into YOLOv7 provided a balance between high precision and lightweight architecture, making it a highly effective solution for object detection tasks across varying object sizes and complex scenes.

Although the MobileNetv3-YOLOv7 model demonstrated strong overall performance, it faced challenges in detecting smaller objects within densely populated scenes, which is a common limitation for object detection models. Future research could explore incorporating multiscale feature fusion techniques or advanced attention mechanisms to enhance the model's accuracy in such scenarios. Additionally, further experiments involving more diverse datasets or extended training periods could potentially improve detection accuracy and robustness, expanding the model's applicability across different real-world environments.

IV. CONCLUSION

The combination of YOLOv7 with MobileNetv3 constitutes notable progress in real-time object detection, overcoming limitations from previous YOLO models. This combination achieved an mAP of 0.61 on the challenging COCO dataset in only 120 training epochs, demonstrating both high accuracy and efficiency. This result emphasizes the potential of the proposed model in the deployment of real-world applications, particularly in resource-constrained environments such as mobile and embedded systems. MobileNetv3's lightweight architecture played a key part in boosting YOLOv7's efficiency, allowing the model to maintain high detection accuracy while minimizing computational demands. The MobileNetv3-YOLOv7 model outperformed established models, such as the YOLOv4, YOLOv5, and even the standalone YOLOv7 model, both in terms of accuracy and efficiency. Additionally, the analysis revealed that the model performed robustly across different object sizes and scene complexities, further emphasizing its adaptability to diverse real-world scenarios. This flexibility makes it a suitable solution for real-time object detection tasks, where balancing accuracy and efficiency is critical. In summary, the integration of YOLOv7 and MobileNetv3 offers a well-balanced model that combines accuracy, efficiency, and adaptability, representing a step forward in real-time object detection. As the model continues to evolve, it holds the potential for broader applications across various domains, reinforcing its role in the advancement of computer vision technology.

Future research could explore experimenting with additional lightweight architectures or incorporating more advanced attention mechanisms to further improve detection accuracy, particularly for smaller objects in complex scenes. Additionally, testing the model on different datasets or applying it in real-world scenarios beyond COCO could provide important perspectives on the model's generalizability and resilience, setting the stage for further optimizations and broader applicability.

REFERENCES

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Kauai, HI, USA, 2001, vol. 1, pp. I-511-I–518, https://doi.org/10.1109/CVPR.2001.990517.

[2] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, vol. 1, pp. 886–893, https://doi.org/10.1109/CVPR.2005.177.

[3] R. Girshick, J. Donahue, T. Darrell, and 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, Jun. 2014, pp. 580–587, https://doi.org/10.1109/CVPR.2014.81.

[4] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448, https://doi.org/10.1109/ICCV.2015.169.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." arXiv, 2015, https://doi.org/10.48550/ARXIV.1506.01497.

[6] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, 2016, vol. 9905, pp. 21–37, https://doi.org/10.1007/978-3-319-46448-0_2.

[7] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 2999–3007, https://doi.org/10.1109/ICCV.2017.324.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, https://doi.org/10.1109/CVPR.2016.91.

[9] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, Jul. 2017, pp. 6517–6525, https://doi.org/10.1109/CVPR.2017.690.

[10] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement." arXiv, 2018, https://doi.org/10.48550/ARXIV.1804.02767.

[11] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection." arXiv, 2020, https://doi.org/10.48550/ARXIV.2004.10934.

[12] E. Iren, "Comparison of YOLOv5 and YOLOv6 Models for Plant Leaf Disease Detection," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13714–13719, Apr. 2024, https://doi.org/10.48084/etasr.7033.

[13] 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." arXiv, 2022, https://doi.org/10.48550/ARXIV.2207.02696.

[14] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv, 2017, https://doi.org/10.48550/ARXIV.1704.04861.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, Jun. 2018, pp. 4510–4520, https://doi.org/10.1109/CVPR.2018.00474.

[16] A. Howard *et al.*, "Searching for MobileNetV3," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), Oct. 2019, pp. 1314–1324, https://doi.org/10.1109/ICCV.2019.00140.

[17] T. Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, Zurich, Switzerland, 2014, pp. 740–755, https://doi.org/10.1007/978-3-319-10602-1_48.

[18] "Deed - Attribution 4.0 International - Creative Commons." https://creativecommons.org/licenses/by/4.0/.

[19] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets Great Again," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 13728–13737, https://doi.org/10.1109/CVPR46437.2021.01352.

[20] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 13024–13033, https://doi.org/10.1109/CVPR46437.2021.01283.

[21] T. Jiang and J. Cheng, "Target Recognition Based on CNN with LeakyReLU and PReLU Activation Functions," in *2019 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, Beijing, China, Aug. 2019, pp. 718–722, https://doi.org/10.1109/SDPC.2019.00136.

[22] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021." arXiv, 2021, https://doi.org/10.48550/ARXIV.2107.08430.

[23] A. M. Roy and J. Bhaduri, "A Deep Learning Enabled Multi-Class Plant Disease Detection Model Based on Computer Vision," *AI*, vol. 2, no. 3, pp. 413–428, Aug. 2021, https://doi.org/10.3390/ai2030026.