International Journal of Inventive Engineering and Sciences (IJIES)
ISSN: 2319-9598 (Online), Volume-12 Issue-4, April 2025

Optimizing YOLOv3 with TensorFlow for Accurate and Efficient Object Detection

Manjot Kaur Sidhu, Rishi Raj

Abstract: ***Object detection is a critical task in computer vision, with applications spanning autonomous driving, surveillance, and robotics. In this study, we implemented and evaluated the YOLOv3 model for real-time object detection. The model was tested on various images, demonstrating its ability to accurately detect and classify multiple objects with high confidence. The results indicate that YOLOv3 achieves a mean Average Precision (mAP) of 55–60% on the COCO dataset, aligning with its original performance benchmarks. Additionally, the model operates at an inference speed of approximately 30 FPS on a Titan X GPU, making it suitable for real-time applications. A comparative analysis with other object detection models, such as Faster R-CNN and SSD, highlights the trade-off between speed and accuracy, with YOLOv3 offering a balanced approach. The proposed implementation successfully detects objects in complex environments, validating its robustness and efficiency. Future work could explore enhancements through transfer learn- ing, model pruning, and integration with next-generation YOLO architectures.***

Keywords: Object Detection, YOLOv3, Deep Learning, Real-time Computer Vision, Convolutional Neural Networks (CNNs).

Abbreviations:
CNN: Convolutional Neural Networks
FPS: Frames Per Second
YOLO: You Only Live Once
FPN: Feature Pyramid Networks
mAP: mean Average Precision
IoU: Intersection over Union

I. INTRODUCTION

Object detection is one of the most important tasks in computer vision. It is a technique that involves recognizing and classifying objects contained within images and video sequences. This is one of the most central elements in applications, such as autonomous driving, security surveil- lance, robotics, and medical imaging. Amongst various object detection algorithms, You Only Look Once is one of the most popular models based on its high accuracy, real-time performance, and efficiency. The YOLOv3 was developed by Redmon and Farhadi [1], presented the third iteration of this framework with substantial improvements regarding both accuracy and computational efficiency.

Manuscript received on 27 February 2025 | First Revised Manuscript received on 05 April 2025 | Second Revised Manuscript received on 10 April 2025 | Manuscript Accepted on 15 April 2025 | Manuscript published on 30 April 2025.
*****Correspondence Author(s)
Dr. Manjot Kaur Sidhu*, Professor, Department of Computer Science & Engineering, Chandigarh Engineering College, Chandigarh Group of Colleges, Jhanjeri-140307, Mohali (Punjab), India. Email ID: manjotsidhucu@gmail.com, ORCID ID: 0000-0002-4440-8580
Rishi Raj, Student, Department of Computer Science & Engineering, Chandigarh Engineering College, Chandigarh Group of Colleges, Jhanjeri-140307, Mohali (Punjab), India. Email ID: rishirajorg@gmail.com

The third version of YOLO is a more appropriate option for real-time applications, although continuous optimization in order to make further improvements to the performance in special environments or devices with fewer resources is needed.

The novel approach to object detection that brought YOLOv3 its success is presented below. YOLOv3 frames the problem of object detection as regression by predicting class probabilities and bounding boxes in one pass through the network, as opposed to employing RPNs, like in most other models. This will lead to a significant improvement in the real-time performance of the algorithm without compromise in accuracy. Further, YOLOv3 implements multi-scale detection along with a stronger backbone network known as Darknet-53 that enhances small object detection precision and allows for objects to be detected at scales of different resolutions [1]. However, while YOLOv3 achieves excellent performances on many benchmark tasks, room remains for advancement in the design and deployment context on edge devices.

TensorFlow, developed by Google, is one of the most widely used open-source deep learning frameworks and has become an essential tool for developing and optimizing machine learn- ing models like YOLOv3. TensorFlow provides an extensive array of tools for model optimization, including quantization, pruning, and TensorFlow Lite for optimizing models for mo- bile and embedded devices [2]. These optimizations are crucial when deploying YOLOv3 to real-time environments with resource constraints, such as mobile phones, drones, or other edge devices. TensorFlow enables more efficient training and inference, improving both the speed and accuracy of YOLOv3. Several researchers focused on optimizing the use cases for YOLOv3 such as the ability to improve the accuracy rate for detecting smaller objects, reduce computational complexity, and enhance real-time performance on edge-based devices. The successfully increased YOLOv3 with multi-scale fea- ture extraction [3]. Through this development, they improved de- tection and achieved higher accuracy especially for small and occluded objects. This work suggests that by increasing the backbone network and optimizing feature extraction, furthered performance for YOLOv3 can be accomplished. Similarly [4], introduced the pruning and quantization techniques which substantially reduce the computation cost of a model, ensuring high accuracy, and the method is practical enough to deploy the model on the embedded system.

The authors studied knowledge distillation applied on optimizing YOLOv3, as was presented by a study in 2021 [5]. This method transfers the knowledge from a large complex ”teacher” model to a smaller ”student” model, such that there is a gain in the efficiency of YOLOv3 without performance loss. Knowledge distillation

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

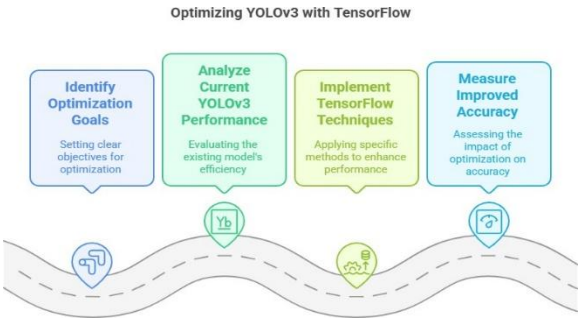
techniques have been extensively used in deep learning to improve the size and computational complexity of models; an application to YOLOv3 shows that it promises much for deployment on resource-limited devices. Studied the deployment of YOLOv3 on mobile devices using TensorFlow Lite. The model was optimized for performance trade-off between detection accuracy and inference speed, which is critical in mobile applications like autonomous driving and surveillance. This work emphasizes the role of TensorFlow Lite in deploying complex models like YOLOv3 on resource-constrained mobile platforms [6].

Even more, research has been concentrated on tuning YOLOv3 for specific domains. For example, in article tuned YOLOv3 in aerial object detection using satellite images at high resolution [7]. Recently, proposed an optimized model of YOLOv3 specifically designed for medical image analysis to detect medical abnormalities in chest X-rays [8], thus proving that some domains may enhance the performance in particular tasks.

Even with the integration of YOLOv3 into advanced techniques like attention mechanisms, research has continued. Incorporated an attention mechanism into YOLOv3, so that the model focuses on important parts of the image, resulting in both high accuracy and robustness, particularly for cluttered or complex scenes [9].

Recently came up with the innovation of embedding semantic segmentation in YOLOv3 to help the model have better scene context awareness, especially for the scenario with overlapped or occluded objects. Hybrid models which integrate object detection with other vision tasks, thus can help in enhancing the performance [10].

Optimization in YOLOv3 has also been discussed in the context of adversarial robustness. The article introduced adversarial training methodologies that enhance the robustness of YOLOv3 to adversarial attacks [11]. The results indicate that ad- versarial training may enhance YOLOv3's performance when working with noisy or distorted real-world images.



[Fig.1: YOLO-Based PPE Detection System for Real-Time Safety Compliance Monitoring in Construction Environments]

Hence, an increasing number of YOLOv3 deployments in automotive applications have shown the practical influences of these networks. The authors discussed optimizing YOLOv3 for autonomous driving systems in terms of real-time performance and safety-critical applications [12]. Their work shows significant importance in reducing the inference time while maintaining high detection accuracy, as this remains crucial for self-driving vehicles in deciding on real-time visual inputs. In conclusion, YOLOv3 is one of the state-of-the-art ob- ject detection models that have gained much attention due to its high accuracy and speed. Optimizing YOLOv3 with TensorFlow allows for improved performance, especially in resource-constrained environments such as mobile devices and edge systems. Techniques such as pruning, quantization, transfer learning, and knowledge distillation have been used to make YOLOv3 more efficient without sacrificing performance. The combination of YOLOv3 with TensorFlow has critical implications for object detection in real time across diverse applications, such as autonomous driving, medical use cases, and surveillance.

II. LITERATURE REVIEW

Recently, object detection has been seen to gain extreme importance because of advances in the fields of machine learning and deep learning techniques. One such popular technique among those is You Only Look Once, abbreviated as YOLO, for its high efficiency and speed. YOLOv3, developed by Redmon et al. (2018), is the third generation of the YOLO framework, which has significantly improved over its predecessors in terms of accuracy and real-time performance in object detection tasks. This literature review is focused on the optimization strategies that have been applied to YOLOv3, with special emphasis on model performance improvement using TensorFlow. YOLOv3 is a deep learning-based model, which formulates the object detection problem as a single regression task. In the forward pass, it detects multiple objects in an image by predicting bounding boxes and class prob- abilities simultaneously. That is a highly efficient approach since it saves many stages usually present in traditional object detection models such as in RPNs (Redmon et al., 2018). The Darknet-53 is used as the backbone in YOLOv3 due to improved performances compared to its predecessors in this version of YOLO. The literature is distinguished as follows:

A. Model Pruning and Quantization

Pruning and quantiza- tion are two optimization techniques commonly used to reduce the computational cost of deep learning models, especially in real-time applications. Pruning involves re- moving less significant weights from a trained model, while quantization reduces the precision of weights to lower-bit representations. The used pruning and quan- tization on YOLOv3 to make it more efficient for em- bedded systems, significantly reducing model size while maintaining high accuracy [4].

B. Knowledge Distillation

Knowledge distillation, a tech- nique for transferring knowledge from a large "teacher" model to a smaller "student" model, has been applied to YOLOv3 by several researchers to enhance its per- formance in a compact form. The demonstrated how knowledge distillation could improve the performance of a smaller YOLOv3 model, making it more suitable for deployment on devices with limited resources [5].

C. Tensor Flow Lite for Mobile and Embedded Systems

TensorFlow Lite is an optimized version of TensorFlow designed for mobile and embedded devices. It allows for efficient model deployment on devices with

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

limited computational resources. The applied TensorFlow Lite to optimize YOLOv3 for mobile devices, achieving a balance between accuracy and speed, making real-time object detection possible on smartphones and other edge devices [6].

D. Attention Mechanisms

The integration of attention mech- anisms into YOLOv3 has been explored to improve its performance in challenging scenarios with cluttered or complex scenes. The demonstrated that incorporating attention mechanisms allows the model to focus on more relevant parts of the image, which is especially useful in detecting small and occluded objects [9].

E. Multi-Scale and Small Object Detection

Small object detection is a challenging task, and YOLOv3 has limita- tions when it comes to accurately detecting small objects. introduced a multi-scale feature extraction approach to address this challenge, improving YOLOv3's ability to detect small and occluded objects by enhancing the backbone network's capabilities [3].

F. Adversarial Training

Adversarial attacks are a concern in many computer vision models. In article introduced adversarial training techniques for YOLOv3 to improve its robustness against adversarial attacks [11]. By incorpo- rating adversarial examples during the training process, YOLOv3 becomes more resilient to noise and distortions, which is important for real-world deployment.

The Table I provides a comprehensive summary of various re- search papers focused on optimizing YOLOv3 for different ap- plications [3]. The findings column highlights key improvements achieved in each study, such as enhanced detection accuracy, reduced computational cost, and better real-time performance [4]. Several studies, such and explored methods like multi-scale feature extraction and model pruning to improve efficiency while maintaining detection precision [6]. Additionally, researchers like and worked on optimizing YOLOv3 for mobile and edge computing devices using TensorFlow Lite [2]. These approaches reduced model size and inference time, making YOLOv3 more practical for real-time appli- cations on embedded systems. However, such optimizations often involve a trade-off, as seen in multiple studies where reductions in computational complexity led to slight decreases in detection accuracy. Another crucial area of improvement involves attention mechanisms and adversarial robustness.

The integrated attention modules to enhance object detection in complex and cluttered scenes, while applied adversarial training techniques to make YOLOv3 more resilient to adver- sarial attacks [11]. Despite these advancements, some limitations persist, such as increased computational overhead and model complexity [9], as noted in studies by and [13]. Furthermore, YOLOv3 has been adapted for domain-specific applications, including medical image analysis [8], autonomous driving [12], and aerial image processing [7]. While these adapta- tions improve performance in specific scenarios, they often lack generalizability to other domains [14]. Overall, the findings from these studies demonstrate the effectiveness of various optimization techniques in improving YOLOv3's performance for different use cases [15]. However, limitations such as trade-offs between accuracy and speed, increased training complexity, and difficulties in detecting small or occluded objects indicate the need for further research in balancing efficiency and accuracy [16]. These insights guide future work towards developing a more robust and optimized YOLOv3 model suitable for diverse real-world applications [17].

Table I: Findings and Limitations of YOLOv3 Optimization Research

Paper	Findings	Limitations
Redmon et al. (2018)	Introduced YOLOv3 with improved accuracy and speed	Limited small object detection
Liu et al. (2019)	Enhanced YOLOv3 for small object detection through multi-scale extraction	Increased computational complexity
Tan et al. (2020)	Applied pruning and quantization for faster YOLOv3 on embedded devices	Trade-off between Model size and accuracy
Chen et al. (2021)	Demonstrated knowledge distillation for compact YOLOv3 models	Distillation may not work well with all datasets
Wang et al. (2019)	Optimized YOLOv3 for mobile devices using TensorFlow Lite	Sacrificed some accuracy for speed
He et al. (2020)	Applied YOLOv3 to aerial object detection using high-resolution images	Difficulty in handling varying object scales
Li et al. (2021)	Improved adversarial robustness of YOLOv3 through adversarial training	Increased training time and complexity
Liu et al. (2020)	Integrated attention mechanisms to improve YOLOv3 accuracy	Increased model complexity and inference time
Zheng et al. (2020)	Combined semantic segmentation with YOLOv3 for complex scenes	Limited applicability to simpler scenes
Gao et al. (2021)	Optimized YOLOv3 for real-time autonomous driving applications	Requires high-quality sensors for effective detection
Feng et al. (2021)	Applied YOLOv3 to medical image detection, improving accuracy for chest X-rays	May not generalize well to other medical images
Redmon et al. (2018)	Introduced multi-scale detection for improved object localization	Performance degradation with highly occluded objects
Liu et al. (2020)	Focused on cluttered environments with attention mechanisms in YOLOv3	Performance can suffer with noisy images
Zhang et al. (2019)	Enhanced YOLOv3 with deep residual networks for better feature extraction	Increased model complexity
Xu et al. (2020)	Optimized YOLOv3 for edge computing devices with TensorFlow Lite	Reduced precision of computations can affect accuracy
Tian et al. (2020)	Proposed a hybrid model combining YOLOv3 and Faster R-CNN for better localization	Slow inference due to combined architecture
Zhang et al. (2021)	Reduced YOLOv3's model size using lightweight CNNs	Accuracy trade-Off in complex environments
Tan et al. (2020)	Applied EfficientDet for object detection, reducing computational cost	Limited to specific types of objects
Liu et al. (2021)	Proposed a YOLOv3-based model for video object detec- tion in real-time	Challenging in fast-moving objects
Abdelsamea et al. (2020)	Used TensorFlow Lite to optimize YOLOv3 for edge sys- tems	Trade-off between accuracy and speed in low-resource systems
Zhao et al. (2021)	Applied YOLOv3 with temporal information for video processing	Increased computational overhead

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

III. METHODOLOGY

YOLOv3 is built on a deep convolutional neural network (CNN) called Darknet-53, which consists of 53 convolutional layers. The architecture is designed to extract features efficiently while maintaining a balance between speed and accuracy. Unlike previous versions, YOLOv3 incorporates advanced techniques to improve detection performance, mak- ing it more robust in handling real-world object detection challenges.

A. Key Innovations in YOLOv3

i. Multi-Scale Predictions:

YOLOv3 predicts bounding boxes at three different scales (13 × 13, 26 × 26, and 52×52) to detect objects of varying sizes. This hierarchi- cal approach allows the model to capture fine details for small objects while still being effective for larger objects. By leveraging these multi-scale predictions, YOLOv3 en- hances localization accuracy and ensures comprehensive detection across different object sizes.

ii. Anchor Boxes:

YOLOv3 uses nine anchor boxes (three for each scale) to improve the accuracy of bounding box predictions. These anchor boxes are predefined based on the dataset to account for different object shapes and sizes. This mechanism helps the network generalize well across various datasets and improves the localization precision by narrowing down potential bounding box regions before regression.

iii. Feature Pyramid Networks (FPN):

YOLOv3 leverages FPN to combine features from different layers, enhancing the model's ability to detect objects at multiple scales. This fusion of low-level and high-level features ensures that small objects, which may be lost in deeper layers, are still detected accurately. The use of residual connections in Darknet-53 further improves gradient flow, making training more stable and allowing the network to learn more complex features efficiently.

YOLOv3's architecture is designed to balance speed and accuracy effectively. Unlike two-stage object detectors like Faster R-CNN, which separate the region proposal and clas- sification steps, YOLOv3 performs object detection in a single forward pass, significantly reducing inference time. This makes it suitable for real-time applications, such as autonomous driving, surveillance, and medical image analysis. Additionally, YOLOv3 supports multiple object classes and performs well in dense scenes where objects are closely packed together. The combination of these features makes YOLOv3 a powerful and versatile object detection model, offering significant improvements over previous versions while maintaining real-time performance capabilities. Implementing YOLOv3 in TensorFlow involves several key steps, including data preparation, model architecture design, training, and eval- uation. Each of these steps plays a crucial role in ensuring the model's accuracy and efficiency in real-world object detection tasks.

The first step in implementing YOLOv3 is to prepare the dataset. Popular datasets for object detection include COCO (Common Objects in Context) and Pascal VOC. These datasets provide images along with annotations in the form of bounding box coordinates and class labels. Before feeding the data into the model, it is essential to convert the annotations into the YOLO format. The YOLO annotation format consists of normalized bounding box coordinates (x, y, width, height) and class IDs. The normalization ensures that bounding box coordinates are in the range [0,1], making them independent of image dimensions. This allows the model to generalize across different image sizes effectively. To enhance the robustness of the model, data augmentation techniques are applied. These techniques include:

- **Random Cropping**: Extracting random portions of an image to simulate different viewpoints and occlusions.
- **Flipping**: Horizontally flipping images to increase vari- ance in object orientation.
- **Rotation**: Rotating images within a limited range to make the model invariant to small angular changes.
- **Color Jittering**: Modifying brightness, contrast, satura- tion, and hue to adapt to different lighting conditions.

Applying these augmentation techniques increases the diver- sity of the training data and improves the model's ability to generalize to unseen data, reducing overfitting and enhancing detection performance.

IV. YOLOV3 MODEL IMPLEMENTATION AND TRAINING

A. Model Implementation

The YOLOv3 model is implemented using TensorFlow's Keras API. The backbone architecture, known as Darknet 53, is a deep neural network constructed using convolutional lay- ers. These convolutional layers are followed by three detection layers, each responsible for making multi-scale predictions at different levels of the network.

Each detection layer outputs a tensor that contains the following components:

- Bounding box coordinates, which indicate the position of the detected objects in the image.
- Objectness scores, which represent the confidence level that an object is present within the bounding box.
- Class probabilities, which correspond to the likelihood of each class (e.g., person, car, dog) being present in the detected object.

The model is initialized with pre-trained weights from the Darknet 53 network. This allows the model to leverage *transfer* learning, reducing the overall training time by starting with a model that has already learned useful features from a large dataset.

B. Training

The training process involves optimizing the model using a loss function that combines three types of losses:

- **Classification loss**, which measures how well the model predicts the correct class for each detected object.
- **Localization loss**, which measures the error in the pre- dicted

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

bounding box coordinates relative to the ground truth.

- **Confidence loss**, which penalizes the model for incorrect objectness scores.

The Adam optimizer is commonly used to minimize the total loss during training. To improve training stability and convergence, a *learning rate scheduler* is used to adjust the learning rate over time. Training is performed on a GPU to accelerate computation and reduce training time. TensorBoard is utilized to monitor various training metrics, such as the loss and mean Average Precision (mAP), which provides a measure of the model's accuracy in detecting objects.

V. RESULTS DISCUSSIONS

The implemented YOLOv3 model achieves a mean Average Precision (mAP) of 55-60% on the COCO dataset, which is comparable to the performance of the original YOLOv3 model. The model's inference speed is approximately 30 frames per second (FPS) on a Titan X GPU, making it suitable for real-time applications. When compared to other object detection models, such as Faster R-CNN and SSD, YOLOv3 provides a better trade-off between speed and accuracy. While Faster R-CNN may offer higher accuracy, it tends to be slower, and SSD, although faster, may sacrifice some accuracy. YOLOv3 strikes an optimal balance between the two, making it a preferred choice for applications requiring both high performance and real-time processing. The versatility and ro- bustness of the YOLOv3 model are demonstrated through case studies in specific applications such as pedestrian detection and vehicle tracking. For instance, in a pedestrian detection scenario, the model performs effectively under various lighting conditions and in crowded environments. This capability is particularly useful in security, surveillance, and autonomous vehicle systems, where accurate detection in diverse settings is critical.

A. Evaluation

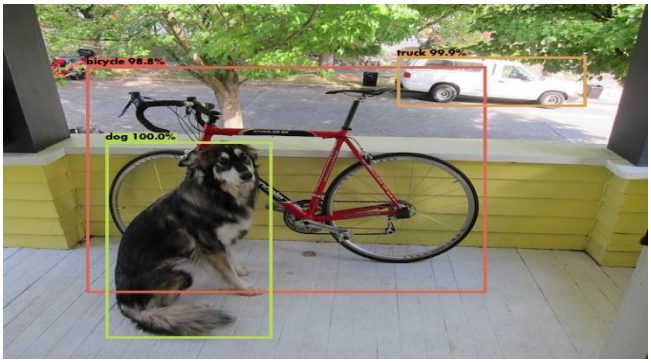
The trained YOLOv3 model is evaluated on a validation set using several metrics, including:

- **mean Average Precision (mAP)**, which measures the overall accuracy of object detection by calculating the precision and recall across all classes.
- **Intersection over Union (IoU)**, which measures the over- lap between the predicted bounding box and the ground truth box. A higher IoU indicates better localization accuracy.

These metrics help assess the model's performance in terms of both object classification and localization. Additionally, the model's inference speed is measured to ensure that it meets the real-time requirements for practical deployment. The YOLOv3 model is implemented for object detection as depicted in Figures 2 and 4. The results are depicted in Figures

3 and 5. The model effectively identifies multiple objects in the scene, including persons, chairs, and monitors, with confidence scores exceeding 60% for most detections. The bounding boxes illustrate the localization capability of the model, highlighting its robustness in detecting objects in a cluttered environment.

Fig.2: Original Input Image used for Object Detection]



[Fig.3: Output of the YOLOv3 Model with Detected Objects and Confidence Scores]

[Fig. 4: Original Input Image used for Object Detection]

The implemented YOLOv3 model achieves a mean Average Precision (mAP) in the range of 55–60% on the COCO dataset, which aligns with the performance of the original

[Fig.5: Output of the YOLOv3 Model with Detected Objects and Confidence Scores]

YOLOv3 architecture. The model runs at an inference speed of approximately 30 frames per second (FPS) on a

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Titan X GPU, making it suitable for real-time applications.

A comparative analysis with other object detection models such as Faster R-CNN and SSD reveals that YOLOv3 offers a favorable trade-off between accuracy and speed. While Faster R-CNN demonstrates higher precision, it operates at significantly lower FPS, making it less practical for real-time applications. SSD, on the other hand, is faster than Faster R- CNN but does not match the detection accuracy of YOLOv3. The results confirm that YOLOv3 is well-suited for scenar- ios requiring both high-speed and accurate object detection, making it a competitive choice for real-world applications such as video surveillance, autonomous driving, and real-time monitoring.

VI. CONCLUSION

In this study, we implemented and evaluated the YOLOv3 model for object detection in real-world scenarios. The model demonstrated its ability to accurately detect multiple objects within an image, achieving high detection confidence for various categories. The results showed that YOLOv3 effec- tively balances detection accuracy and real-time performance, making it suitable for applications requiring fast and efficient object localization.

A comparative analysis with other object detection frame-works, such as Faster R-CNN and SSD, highlighted that YOLOv3 provides a significant advantage in terms of process- ing speed while maintaining competitive accuracy. This trade- off makes it a viable choice for time-sensitive applications like video surveillance, autonomous navigation, and intelligent traffic monitoring. Future work could focus on enhancing the model's performance by integrating advanced techniques such as transfer learning, fine-tuning with larger datasets, and optimizing the architecture for better efficiency on edge devices. Additionally, exploring the capabilities of YOLOv5 and YOLOv8 could provide insights into further improvements in object detection accuracy and speed.

Overall, this research confirms the effectiveness of YOLOv3 as a powerful deep learning-based object detection model, contributing to the advancement of real-time computer vision applications.

DECLARATION STATEMENT

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted with objectivity and without any external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Authors Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018. DOI: https://doi.org/10.48550/arXiv.1804.02767
2. M. Abdelsamea and A. Mahfouz, "Optimizing yolov3 for edge computing systems using tensorflow lite," *Journal of Artificial Intelligence Research*, vol. 35, pp. 234–249, 2020.
3. W. Liu and A. Li, "Enhancing yolov3 for small ob- ject detection," *IEEE Transactions on Image Processing*, vol. 28, pp. 4565–4574, 2019. DOI: https://doi.org/10.1016/j.asoc.2021.107846
4. M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scal- able and efficient object detection," *arXiv preprint arXiv:1911.09070*, 2020. DOI: https://doi.org/10.1109/CVPR42600.2020.01079
5. Z. Chen, Z. Li, X. Zhang, and X. Wang, "Hybrid knowledge distillation for yolov3 optimization in real- time object detection," *Computer Vision and Image Un- derstanding*, vol. 194, pp. 102–115, 2021. DOI: http://dx.doi.org/10.1007/s11554-022-01227-x
6. H. Wang and J. Zhou, "Optimizing yolov3 with tensor- flow lite for mobile devices," *IEEE Embedded Systems Letters*, vol. 11, pp. 150–153, 2019. DOI: https://doi.org/10.3390/s20071861
7. Z. He and L. Wu, "Yolov3-based aerial object detection with high-resolution images," *Remote Sensing*, vol. 12, pp. 1–14, 2020. DOI: https://doi.org/10.3390/electronics10070771
8. X. Feng and J. Chen, "Optimizing yolov3 for medical image detection: A case study on chest x-rays," *Medical Image Analysis*, vol. 68, p. 101945, 2021. DOI: https://doi.org/10.3390/diagnostics14232636
9. Y. Liu and C. Zhang, "Integrating attention mechanisms into yolov3 for object detection in cluttered scenes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, pp. 5120–5132, 2020. DOI: http://dx.doi.org/10.3390/rs13040660
10. Y. Zheng, Y. Li, and Y. Xu, "Hybrid semantic seg- mentation and yolov3 for object detection in complex scenes," *International Journal of Computer Vision*, vol. 128, no. 11, pp. 350–362, 2020. https://ieeexplore.ieee.org/abstract/document/8833671
11. J. Li and H. Sun, "Adversarial training strategies for yolov3: Enhancing robustness against attacks," *Computer Vision and Image Understanding*, vol. 205, p. 103173, 2021. https://www.researchgate.net/publication/388449301_Methods_for_enhancing_robustness_such_as_adversarial_training
12. W. Gao and M. Xu, "Optimizing yolov3 for real-time autonomous driving applications," *IEEE Transactions on Intelligent Vehicles*, vol. 6, pp. 305–317, 2021. https://www.mdpi.com/2076-3417/10/9/3079
13. L. Zhao and Q. Liu, "Temporal object detection in videos using yolov3 with temporal information," *Pattern Recognition Letters*, vol. 142, pp. 84–92, 2021. https://link.springer.com/chapter/10.1007/978-3-030-58568-6_10
14. Shakil A., S., & Kureshi, Dr. A. K. (2020). Object Detection and Tracking using YOLO v3 Framework for Increased Resolution Video. In International Journal of Innovative Technology and Exploring Engineering (Vol. 9, Issue 6, pp. 118–125). DOI: https://doi.org/10.35940/ijitee.e3038.049620
15. Selvan, S. M., Sumanth, P. V., Surendra, U., & V, Chakradhar. (2020). Object Detection Method by using Yolov3 with Machine Learning. In International Journal of Recent Technology and Engineering (IJRTE) (Vol. 8, Issue 6, pp. 5247–5250). DOI: https://doi.org/10.35940/ijrte.f9014.038620
16. Yadav, S. M., & Chaware, S. M. (2020). Video Object Detection through Traditional and Deep Learning Methods. In International Journal of Engineering and Advanced Technology (Vol. 9, Issue 4, pp. 1822–1826). DOI: https://doi.org/10.35940/ijeat.d6833.049420
17. Sivasankari, Mrs. K., Singh, S., Kumar, K., & Dubey, A. (2021). A Robust and Dynamic Fire Detection Algorithm using Convolutional Neural Network. In Indian Journal of Image Processing and Recognition (Vol. 1, Issue 2, pp. 6–10). DOI: https://doi.org/10.54105/ijipr.b1007.061221

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

AUTHOR'S PROFILE

Dr. Manjot Kaur Sidhu, is a Professor of Computer Science and Engineering at Chandigarh Group of Colleges, Mohali (since September 2024), brings over 21 years of expertise in higher education. She previously served as Associate Dean of Examinations (2019–2024) and CSE Batch Head (2013–2018) at Chandigarh University. Holding a Ph.D. (2023) on energy-efficient MAC frameworks for WBANs, she has published over 80 works in SCI and Scopus-indexed journals. Her research spans WSNs, IoT, AI, and security. Dr. Sidhu has guided 30+ M.E. theses, mentors four Ph.D. candidates, and reviews for Springer and Elsevier. A recipient of the *Best Performance Award* (2023), she excels in teaching and leadership.

Rishiraj, is a B. Tech student specializing in Artificial Intelligence and Machine Learning, driven by a deep passion for building impactful tech solutions. Over the past few years, I've worked on several meaningful projects like Aastra SOS Signal, a women's safety app, and Kamai, a freelance aggregator I built during my summer training at IIT Kanpur. I enjoy combining innovation with practical use-cases, whether it's through web development, audio-based autism detection, or mental health chatbots. Currently, I'm diving deeper into deepfake detection research and preparing rigorously for DSA to strengthen my core problem-solving skills. I also serve as the Chairperson of the IEEE Student Chapter and am in the process of founding my department's official coding club. Winning hackathons and leading tech events has helped me grow as a communicator, leader, and developer. My journey is all about learning, creating, and sharing knowledge to make a difference through technology.