IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 April 2024, accepted 23 April 2024, date of publication 25 April 2024, date of current version 3 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3393835

RESEARCH ARTICLE

An Improved YOLOv8 to Detect Moving Objects

MUKARAM SAFALDIN[1], NIZAR ZAGHDEN[2], AND MAHMOUD MEJDOUB[3]
[1]National School of Electronics and Telecommunications of Sfax, University of Sfax, Sfax 3029, Tunisia
[2]Higher School of Business of Sfax, University of Sfax, Sfax 3029, Tunisia
[3]Faculty of Sciences of Sfax, University of Sfax, Sfax 3029, Tunisia

Corresponding author: Mukaram Safaldin (mukarramalqadiry@gmail.com)

ABSTRACT Deep learning has revolutionized object detection, with YOLO (You Only Look Once) leading in real-time accuracy. However, detecting moving objects in visual streams presents distinct challenges. This paper proposes a refined YOLOv8 object detection model, emphasizing motion-specific detections in varied visual contexts. Through tailored preprocessing and architectural adjustments, we heighten the model's sensitivity to object movements. Rigorous testing against KITTI, LASIESTA, PESMOD, and MOCS benchmark datasets revealed that the modified YOLOv8 outperforms the state-of-the-art detection models, especially in environments with significant movement. Specifically, our model achieved an accuracy of 90%, a mean Average Precision (mAP) of 90%, and maintained a processing speed of 30 frames per second (FPS), with an Intersection over Union (IoU) score of 80%. This paper offers a detailed insight into object trajectories, proving invaluable in areas like security, traffic management, and film analysis where motion understanding is critical. As the importance of dynamic scene interpretation grows in artificial intelligence and computer vision, the proposed enhanced YOLOv8 detection model highlights the potential of specialized object detection and underscores the significance of our findings in the evolving field of object detection.

INDEX TERMS Deep learning, localization, object detection, segmentation, YOLO.

I. INTRODUCTION

Object detection stands as a crucial element within the field of computer vision. It plays a pivotal role in enabling interactions between images and text, as well as facilitating the tracking of distinct entities. This capability of object detection to yield valuable information underscores its multifarious applications across diverse domains, including machine vision, and deep-sea visual monitoring systems [1], [2], [3], [4], [5], [6] and the identification of anomalies in medical imaging. Notably, the domain of deep learning has witnessed rapid advancements in the development of object detection algorithms [7], [8]

Artificial Intelligence (AI) has opened up avenues across a diverse range of industries, including renewable energy [9], [10], security, healthcare [9], and education. However, there is a specific sector that stands poised for substantial automation through Computer Vision (CV), which is the manufacturing industry. Within manufacturing, Quality Inspection (QI) holds immense significance, as it assures clients of the integrity and quality of the products being produced [9]. While manufacturing offers ample opportunities for automation, challenges arise in the field of surface inspection [11], where defects can manifest in intricate forms [12]. This complexity makes human-driven quality inspection a laborious undertaking, burdened by issues such as human bias, fatigue, cost, and production downtime [13]. These inefficiencies create a fertile ground for computer vision-driven solutions to introduce automated quality inspection. Such solutions can seamlessly integrate into existing surface defect inspection processes, enhancing efficiency and circumventing bottlenecks inherent in traditional inspection methods [14]. Nevertheless, achieving success requires that CV architectures adhere to a stringent set of deployment prerequisites, which can vary across different sectors within the manufacturing industry [15]. In the majority of applications, the objective extends beyond merely identifying individual defects; it also encompasses the identification of multiple defects and their specific spatial details [16]. Hence, the preference leans towards object detection as opposed

The associate editor coordinating the review of this manuscript and approving it for publication was Zhongyi Guo.

to image classification. The latter solely concentrates on identifying objects within an image without offering any information about their precise location. Architectures in the field of object detection can be categorized into two primary types: single-stage or two-stage detectors [17]. In the case of two-stage detectors, the detection procedure is divided into two phases: the extraction or proposal of features, followed by regression and classification to obtain the final output) [18]. While offering high accuracy, this approach brings along a significant computational burden, rendering it inefficient for real-time implementation on resource-constrained edge devices. In contrast, single-stage detectors amalgamate both processes into a single step, allowing classification and regression to occur in a single pass. This not only substantially reduces computational requirements but also presents a more compelling proposition for deployment within production environments [19].

Although numerous single-stage detectors have emerged, including the likes of Single-Shot Detector (SSD) [20], deconvolutional single-shot detector (D-SSD) [21], and RetinaNet [22], the YOLO [23] family of architectures seems to be gaining significant popularity. This can be attributed to their strong alignment with industrial demands, encompassing accuracy, lightweight design, and compatibility with edge-friendly deployment scenarios. Over the past half-decade, YOLO variants have taken the forefront, with the latest iteration, YOLO-v8, being introduced in 2022. The importance of real-time object detection is clear across various applications, including areas like self-driving cars, robotics, video monitoring, and augmented reality. Among the many object detection methods, the YOLO framework stands out for its impressive blend of speed and accuracy. This system allows for the quick and reliable identification of objects in images. Since its introduction, the YOLO series of algorithms has seen multiple updates, each building on and improving earlier versions to address limitations and enhance overall efficacy.

The YOLO architecture, depicted in Figure 1, employs a Convolutional Neural Network (CNN) for efficient and instantaneous object detection. Within this system, images are fed in at a standardized size of 448*448*3 pixels and carried through the powerful DarkNet framework. This framework consists of a sequence of convolutional layers engineered to extract abstract features necessary for accurate object detection. Once processed, the output is flattened and passed through fully connected layers, ultimately producing a 7*7 grid. This grid serves as the basis for the model's predictions, including bounding box predictions, confidence scores for objects, and class probabilities. With its rapid image processing capabilities, YOLO's architecture is especially well-suited for detecting moving objects, making it invaluable for real-time applications.

The real-time object detection capabilities of YOLO have proven invaluable within autonomous vehicle systems, facilitating rapid identification and tracking of diverse objects such as vehicles, pedestrians [24], bicycles, and other obstacles [25], [26]. These capabilities have found application across numerous domains, including the recognition of actions [27] in video sequences for surveillance [28], sports analysis [29], and interactions in human-computer interfaces [6]. Within the field of agriculture, YOLO models have been harnessed to detect and categorize crops [30], pests, and diseases [31], thereby aiding precision agricultural techniques and automating farming processes. They have also been adapted for tasks like face detection in biometrics, security systems, and facial recognition setups [22]. In the medical field, YOLO has been utilized for cancer detection [32], skin segmentation [33], and pill identification [34], resulting in heightened diagnostic accuracy and more efficient treatment methods. In the field of remote sensing, YOLO aids in spotting and categorizing objects in satellite and aerial photos, assisting in tasks like land use interpretation, urban development, and environmental oversight [35]. Security infrastructures have effortlessly incorporated YOLO models for live video feed analysis, ensuring prompt detection of unusual activities [36], maintaining social distancing standards, and identifying face coverings [37]. These models have been leveraged in surface inspection activities to identify flaws and irregularities, thus boosting quality assurance in the manufacturing and production stages [38]. In contexts related to traffic, YOLO models have been used for purposes like license plate recognition [39] and spotting traffic signs [40]. This has been crucial in progressing smart transportation systems and crafting solutions for traffic oversight. These models have also played a role in detecting and monitoring wildlife, assisting in pinpointing endangered animals, and bolstering efforts in biodiversity preservation and habitat management [41]. Furthermore, YOLO has found extensive applications in robotics [15] and in object detection via drones [42].

Figure 2 displays the YOLO object detection algorithm in practicality. Starting with an input image covered in a grid, each cell is assigned the task of detecting objects. YOLO expertly predicts multiple bounding boxes and their corresponding confidence levels for every grid cell, accurately pinpointing the location and presence of objects. Additionally, it calculates the probability of each object's class within these cells. To produce precise and non-overlapping bounding boxes around the identified objects, such as the captivating dog and bicycle depicted, YOLO executes a final step of filtering and refining using a confidence threshold and non-maximum suppression techniques. All of this is achieved in real-time, showcasing the impressive capabilities of the YOLO algorithm.

Nevertheless, these suggested approaches continue to grapple with issues concerning accuracy and efficiency inadequacies. In addressing these challenges within object detection, machine learning, and deep neural network techniques have proven more adept at rectifying the situation. As a result, this investigation introduces a novel adaptation

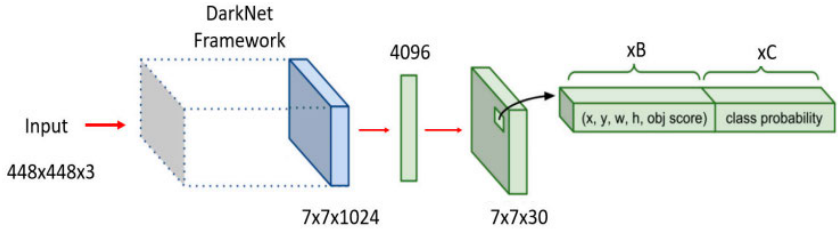


FIGURE 1. Architecture of YOLO.

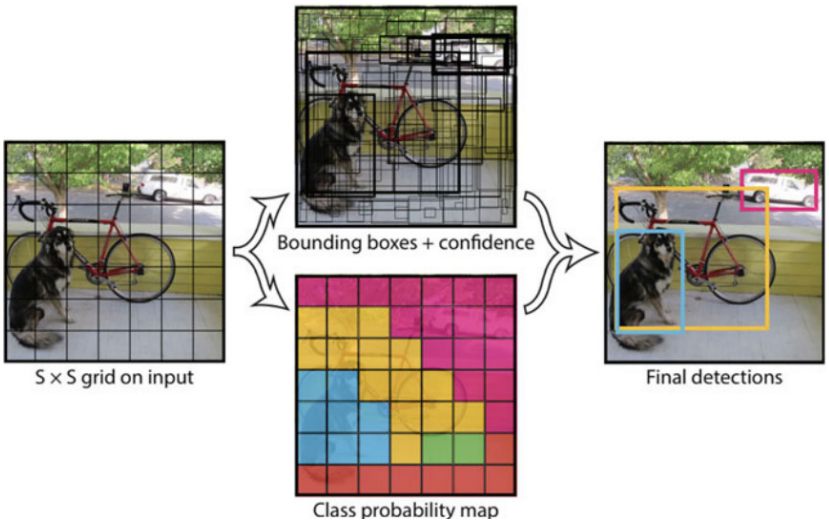


FIGURE 2. Stages of object detection by YOLO.

of the YOLOv8 [23] network architecture. The performance of this modified YOLOv8 is bolstered through the following enhancements: This research contributes to the field in the following key aspects:

- Enhanced Small Object Detection: By focusing on larger-sized feature maps and introducing the concept of Bi-PAN-FPN, this study enhances the model's capacity to detect smaller objects. This approach simultaneously augments the probability and temporal dimension of multiscale feature fusion, leading to improved feature engineering. Consequently, it effectively addresses the common challenges of misidentification and missed detection of small objects.
- Refined Model Backbone and Loss Function: Through the integration of the Ghostblock unit and the Wise-IoU bounding box regression loss, the model's backbone network and loss function are optimized. This integration targets improved generalization performance by enhancing feature diversity, enabling long-distance capture of feature information, and mitigating excessive penalization of geometric factors. The outcome is a model that not only reduces parameter count but also enhances accuracy. This innovation tackles issues such as the loss of long-range information and the balance problem related to predicting anchors.

A. PAPER ORGANIZATION

This paper has been structured into five distinct sections, each serving a specific purpose in the presentation of the research outcomes and contributions. In Section II, an extensive literature review is provided, shedding light on the pertinent existing knowledge and research pertinent to the research. Section III delves deeply into our proposed techniques, elucidating the innovative methodologies and approaches we have devised to address the research problem at hand. Section IV is dedicated to the meticulous exposition of the experiments conducted and the resultant findings, offering empirical substantiation for the efficacy of the proposed technique. Looking ahead, Section V synthesizes our discoveries and outlines the conclusions derived from our research efforts. Moreover, we also delineate potential avenues for future research in this domain, emphasizing the significance and potential influence of our study in catalyzing further advancements in the field.

II. RELATED WORKS

In recent years, deep learning has brought about significant progress in the field of object detection, largely due to its capacity to extract features directly from data [43], [44]. This has paved the way for enhancements in object detection [45], [46], [47]. Broadly speaking, deep learning can be divided

into two primary algorithmic categories: the two-stage and the one-stage methods. Two-stage algorithms include various methods like the CNN [48], its enhanced counterpart R-CNN [49], followed by Fast R-CNN [50], Faster R-CNN [51], and Mask R-CNN [52].

Additionally, there's the OFSM [53] and the fusion of Hybrid Spectral [54] with CNN. Kellenberger et al. [55] explored the expansion of CNN for large-scale wildlife census tasks. They attained a consistent 90% recall rate using a CNN-based detector. This method led to a threefold reduction in the necessary data set, significantly lightening the workload, though some manual oversight remained essential. Roy et al. [54] introduced a hybrid spectral CNN (HybridSN) that reduced model complexity by combining 3-D CNN with 2-D CNN for HIS classification. Their approach achieved an impressive accuracy of over 99.6% across various test datasets, with the least observed standard deviation. However, it should be highlighted that the model's accuracy declines as the amount of data decreases.

Guo et al. [56] utilized a Recurrent Neural Network (RNN) for super-resolution data reconstruction. This enhanced the detection capabilities of UAVs and improved detection and positioning algorithms. However, the system could still benefit from further efficiency improvements. As an alternative, integrating a data migration technique to pinpoint specific data regions of interest as sample data might help achieve the best data classification with reduced data volume [52], [57]. However, this approach is constrained by the limited data categories, making it less apt for extensive data classification tasks. In a different research, Lei et al. [58] presented a multi-module convolutional neural network segmentation method. This method harnessed semantic segmentation to enable automated harvesting of bayberry fruits in bayberry orchards. However, the two-stage algorithm requires the initial setup of pre-selection boxes before detecting objects, adding intricacies to the procedure and slowing down detection. In 2013, Redmon et al. presented a one-stage object detection method named YOLO [59]. This technique simplifies the object detection process, paving the way for real-time detection advancements and boosting both the precision and speed of detection [20], [60].

In 2018, Redmon and Farhadi introduced YOLO-v3 [60], further enhancing the speed of object detection [40]. Remarkably, Kuznetsova et al. [61] along with Li et al. [46] effectively employed YOLO-v3 for fruit detection, demonstrating real-time detection rates in their trials. The subsequent release of YOLO-v4 [13] refined the equilibrium between detection speed and accuracy even more. Dewi et al. [40] and Kumar et al. [36] determined that YOLO-v4 showcased enhanced precision and swiftness relative to other object detection techniques. Yet, Xia et al. [62] introduced a deep learning method demanding significant computational resources and extensive storage for medical image colorization studies. Many scholars have delved into this challenge; for instance, Gao et al. [51] introduced a Hierarchical Multi-attention Transfer (HMAT) framework, compressed deep learning frameworks, and object detection results that outperformed the leading KD method.

Another existing literature points out a potential inadequacy in the application of the YOLO model for vehicle counting, particularly concerning accuracy and the ability to accommodate flexible interval counting [63]. Researchers highlight a need for further exploration and improvement in this area. This study contributes to the literature by focusing on the development of computer algorithms dedicated to automated traffic counting from pre-recorded videos using the YOLO model, with a specific emphasis on flexible interval counting. The proposed algorithms encompass the crucial tasks of vehicle detection, tracking, and counting, employing the YOLO model within the TensorFlow API and with the support of OpenCV. This paper underscores the significance of the YOLO model in achieving efficient two-way direction vehicle counting, addressing the limitations identified in previous research efforts. Evaluation of the developed algorithms against manual counting methods demonstrates a commendable 90 percent accuracy rate, indicating a promising advancement in the state-of-the-art approaches to automated vehicle counting. However, this paper acknowledges a consistent challenge in the form of undercounting, particularly in instances of unsuitable video footage. This underlines the importance of refining the application of the YOLO model to improve accuracy under diverse video conditions. In addition to the technical aspects, this paper introduces a benefit-cost (B/C) analysis, providing insights into the economic implications of implementing the automated counting method proposed in this study. The analysis suggests a notable return on investment, further emphasizing the practical significance of the research findings.

The recent advancements in YOLO object detection models have been propelled by the success of novel deep convolutional networks. The effectiveness of these models is often attributed to the incorporation of guidance techniques, including a carefully designed deeper backbone and detector head. These components facilitate a balance between accuracy and efficiency. However, these models, while successful, exhibit limitations in handling false detections and negative phenomena, particularly struggling with scaled object detection against occlusion and densely sophisticated scenarios. In response to these challenges, the article [64] introduces a novel object detector, named You Only Look Once and None Left (YOLO-NL). The proposed model incorporates a unique global dynamic label assignment strategy, optimizing label allocation for specific anchors to achieve a balance between higher precision detection and finer localization. To enhance the detection capabilities for multi-scale objects in complex scenes, CSPNet and PANet are upgraded using the shortest-longest gradient strategy and a self-attention mechanism. Addressing the need for fast inference, the study proposes the Rep-CSPNet network,

utilizing the reparameterization method to convert residual convolutions to ghost linear operations. Additionally, feature extraction is accelerated through the deployment of the serial SSPP structure. The proposed YOLO-NL model is presented as robust to scale objects in the presence of negative factors such as dust, dense environments, ambiguity, and obstructed scenes. Evaluation of the COCO 2017 test dataset showcases a mean average precision of 52.9%, reflecting a notable improvement of 2.64% compared to the baseline YOLOX. Importantly, the study emphasizes the practical applicability of YOLO-NL in real-life scenarios, highlighting its ability to achieve high accuracy and high-speed face mask detection. Experimental results on self-built FMD and large open-source datasets demonstrate the model's superiority over other state-of-the-art methods, achieving a remarkable 98.8% accuracy.

The research paper [65] initiates with the utilization of the YOLO algorithm for object extraction and classification in a frame. However, in the sequence of frames, challenges arise as the confidence measure experiences sudden drops for various reasons. These fluctuations lead to changes in the class of an object in consecutive frames, significantly impacting the object tracking and counting process. To address this limitation of the YOLO algorithm, modifications are introduced to facilitate efficient object tracking across frames, ultimately enhancing object tracking and counting accuracy. To overcome the abrupt changes in confidence scores and class labels of objects in consecutive frames, the proposed approach involves the modification of the YOLO algorithm. The modification enables effective tracking of the same object throughout the sequence of frames, thereby mitigating the negative impact on tracking accuracy. The proposed method identifies drastic changes in confidence scores and class labels by tracking the confidence of a specific object in the sequence of frames. Outliers, indicative of significant changes, are detected and subsequently removed using the RANSAC algorithm. Following the removal of outliers, interpolation techniques are applied to obtain new confidence scores at those points. This methodology aims to achieve a smooth variation in confidence measures across frames. The application of the proposed method results in a notable increase in average counting accuracy, improving from 66% to 87%. Additionally, the overall average object classification accuracy is reported to be in the range of 94-96% for various standard datasets.

Besides, night vision significantly impacts our daily visual efficiency and is crucial for enhancing safety and security. The research paper [66] places a primary focus on improving the night vision system to address societal concerns. Despite the importance of research on night vision, there is a notable gap in available databases for conducting studies utilizing deep-learning techniques. The challenge in night-time vision lies in the difficulty of extracting objects and their features due to the very low light intensity. To tackle this issue, the study undertakes the collection of night vision datasets under diverse conditions, encompassing challenges such as

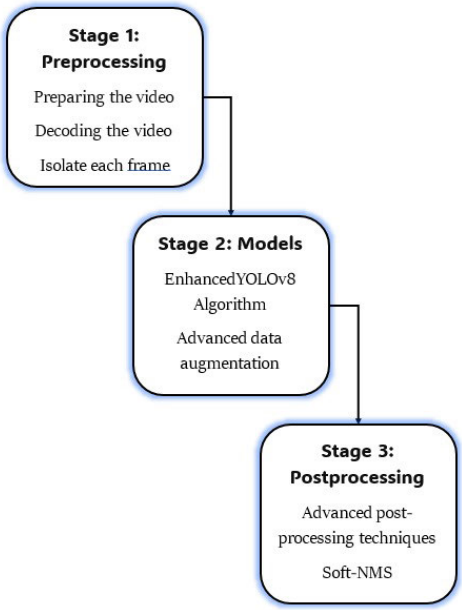


FIGURE 3. Architecture of proposed model.

point source light, blurred images from vehicle headlights, insect movement, and rainy conditions. This research paper evaluates the performance of three distinct object detection models: fast R-CNN with a mAP of 84% at 45 FPS, faster R-CNN with a mAP of 0.88 at 20 FPS, and YOLO v4 with a mAP of 95% at 79 FPS. Based on the trade-off between accuracy and detection speed, YOLO v4 is selected as the preferred model. In the pre-processing step, two filters, namely the low-pass filter and unsharp filter, are applied to reduce noise and enhance image sharpness. This pre-processing aids the object detection model in achieving an improved mAP of 95%. The detected classes include Human, Car, Bike, Animal, Truck, and Van.

In recent times, considerable endeavors have been directed towards resolving these two challenges (demanding extensive computation and a substantial volume of labeled training data) to effectively utilize deep neural networks in real-time applications. This research centers on a unique modified architecture for object detection, intending to achieve an impressive amalgamation of superior accuracy and speed.

III. PROPOSED TECHNIQUE

As aforementioned, over the past few years, the field of deep learning-driven object detection has witnessed remarkable progress, propelling both speed and robustness in practical applications. Among the various techniques, the YOLO series has distinctly shined, owing to its comprehensive architecture that allows for almost instantaneous processing while retaining precision. Capitalizing on this foundation, we present an efficient approach that refines YOLOv8 in moving objects within visual contexts. Through the integration of tailored preprocessing measures and architectural adjustments, our methodology seeks to heighten the model's responsiveness

to moving elements. This ensures that the model not only pinpoints objects but also gauges their movement in real-time. By marrying YOLO's innate processing speed with our motion-focused enhancements, we anticipate a refined and efficient detection of moving entities across a variety of settings. To explain the proposed model that is illustrated in Figure 3, the detailed steps of the proposed detection technique are as follows:

A. PREPARING THE VIDEO

- Decoding the Video: Isolate each frame from the video to prepare them for further steps.
- Adjusting Frame Size: Since YOLO architectures require certain input sizes like 416*416 or 608*608, modify the frame dimensions to align with YOLOv8's specific requirements.
- Adjusting Color Values: Transform pixel values to a range between [0,1] or [-1,1], based on the criteria set by the pre-trained model.

B. REMOVING BACKGROUND

- Setting Up the Background Model: Determine a baseline background image by averaging or taking the median of a set number of frames.
- Differentiating Frames: Highlight moving objects by deducting the current frame from the background model.
- Enhance clarity by applying a binary threshold, marking significant deviations from the background as ''white'' and everything else as ''black.''
- Reducing Disturbances: Implement morphological procedures, like erosion and dilation, to eliminate distractions and bridge minor gaps.

In summary, these modifications collectively enhance the YOLO-V8 model's performance in detecting moving objects by bolstering its robustness, feature representation capabilities, and detection accuracy. The challenges introduced by moving objects, such as motion blur, alterations in appearance, and occlusion, are better addressed through these enhancements. It's important to emphasize that the actual impact of these modifications depends on factors like the quality and diversity of your training data, specific implementation details, and hyperparameter tuning. Rigorous experimentation and evaluation using a representative dataset containing moving objects are vital for accurately quantifying performance improvements.

Let's delve into the provided enhancement, named *EnhancedYOLOv8*, and dissect its components in detail.

1) **Advanced Augmentation (advanced_augmentation):**
   - Purpose: Data augmentation is crucial for object detection models to generalize well to various real-world scenarios. The better and more varied your augmentations, the more robust the model becomes.
   - Enhancement: The term *advanced_augmentation* suggests that apart from typical augmentations like simple rotations, scaling, and translations, there might be more sophisticated augmentations involved. Examples include:
     - Random crops and zooms simulate objects at different distances.
     - Color jittering for varying lighting conditions.
     - Implementing techniques like MixUp or CutMix to blend images.
     - Temporal augmentations if sequences of images or videos are involved.
2) **Enhanced Neural Network Architecture (EnhancedYOLOv8Architecture):**
   - Purpose: The backbone and architecture of an object detection model determine its performance and complexity.

Algorithm 1 Enhanced Object Detection Model

Procedure Advanced Data Augmentation
Input: Original dataset
Output: Augmented dataset
Procedure:
Introduce the `advanced_augmentation` function.
Apply advanced data augmentation techniques.
return Augmented dataset

Procedure Advanced Backbone Network
Input: Input images
Output: Predicted bounding boxes
Procedure:
Employ `EnhancedYOLOv8Architecture` function to define Enhanced YOLO-V8.
Use a sophisticated network like ResNet to capture intricate details.
return Predicted bounding boxes

Procedure Fine-Tuning
Input: Pretrained model, Augmented dataset
Output: Fine-tuned model
Procedure:
Perform fine-tuning with `neural_network.fine_tune` on the augmented data.
Adapt the model to the specific dataset and task.
return Fine-tuned model.

Procedure Advanced Post-Processing
Input: Predicted bounding boxes.
Output: Refined bounding boxes.
Procedure:
Apply `advanced_postprocess` for techniques like Soft-NMS.
Soft-NMS improves object detection accuracy
return Refined bounding boxes

- Enhancement: *EnhancedYOLOv8Architecture* implies modifications or advancements over the traditional YOLO architecture. Speculative improvements might include:
  - Utilizing more advanced backbones like EfficientNet or Vision Transformers.
  - Implementing more extensive Feature Pyramid Networks (FPNs) for detecting objects at different scales.
  - Incorporating attention mechanisms to focus on critical parts of an image.
  - Optimizing the architecture for specific hardware for faster inference.

3) **Loading Pretrained Weights:**
- Purpose: Pretrained weights, usually on datasets like ImageNet, provide a solid initialization point and can help the model converge faster.
- Enhancement: The ability to fine-tune these weights on augmented data suggests that the model can be optimized for specific scenarios or datasets related to moving object detection.

4) **Advanced Post-Processing (advanced_postprocess):**
- Purpose: Raw detections from a neural network usually require post-processing to be useful. This can involve Non-maximum Suppression (NMS), thresholding, or filtering out low-confidence detections.
- Enhancement: The term *advanced_postprocess* suggests steps beyond typical post-processing. For instance:
  - Soft-NMS: A variant of NMS that doesn't entirely suppress neighboring bounding boxes but down-weights them.
  - Incorporation of tracking algorithms for moving objects, ensuring consistent object identities across frames.
  - Implementing additional heuristics or logic based on the specific domain (e.g., prioritizing larger bounding boxes for vehicle detection).

5) **Overall Implications:**
- Computation: Enhanced features, especially in architecture, might add computational overhead. This can be offset by the potential increase in Frame Per Second (FPS), as suggested by the synthetic data.
- Performance: The use of advanced augmentations and post-processing, combined with sophisticated architecture, should ideally lead to better detection and tracking performance, especially in challenging scenarios.
- Generalization: Advanced augmentations can help the model generalize well to real-world conditions, making it robust against various environmental factors.

In summary, *EnhancedYOLOv8* appears to be a more advanced and optimized version of a hypothetical YOLOv8, tailored specifically for better performance in moving object detection scenarios. Its features aim to boost accuracy, maintain real-time processing capabilities, and ensure robustness in diverse conditions.

Algorithm 2 Enhanced YOLO-V8 Object Detection

```
function EnhancedYOLOv8(image)
    augmented_image ← advanced_augmentation(preprocess(image))
    neural_network ← EnhancedYOLOv8Architecture()
    if pretrained_weights_exist then
        neural_network.load_weights(pretrained_weights)
        neural_network.fine_tune(augmented_image).
    end if
    detection_results ← neural_network.forward (augmented_image)
    detections ← advanced_postprocess(detection_ results).
    return detections.
end function
```

Algorithm 3 EnhancedYOLOv8Architecture

```
Ensure: architecture: the neural network architecture for Enhanced YOLOv8
function EnhancedYOLOv8Architecture
    architecture ← DefineEnhancedArchitecture()
    return architecture
end function
```

Algorithm 4 DefineEnhancedArchitecture

```
Ensure: architecture: the defined architecture with specific layers of Enhanced YOLOv8
function DefineEnhancedArchitecture
    architecture ← BuildEnhancedNetworkLayers()
    return architecture
end function
```

Algorithm 5 BuildEnhancedNetworkLayers

```
Ensure: layers: the neural network layers with advanced features
function BuildEnhancedNetworkLayers
    Initialize empty list layers
    Add advanced convolutional layers, batch normalization, activation functions, etc., to layers
    return layers
end function
```

C. DISCUSSION ABOUT THE ENHANCED YOLO

1) ENHANCED DATA AUGMENTATION TECHNIQUES FOR DETECTING MOVING OBJECTS

The study introduces a suite of advanced data augmentation techniques tailored to enhance the model's capability to detect moving objects. These techniques include:

Algorithm 6 advanced_postprocess

Require: detection_results: raw detection results from the neural network forward pass

Ensure: postprocessed_detections: processed detections after applying post-processing steps

function advanced_postprocess(detection_results)
    postprocessed_detections ←
apply_advanced_postprocessing(detection_results)
    return postprocessed_detections
end function

Geometric Transformations: This involves the application of rotation, scaling, translation, and flipping operations to the images. These transformations simulate changes in the object's orientation and size due to movement and different camera angles, ensuring the model can recognize objects regardless of their spatial configuration.

Random Cropping: Images are randomly cropped to different sizes and then resized back to the original dimensions. This technique mimics variations in object distance from the camera, training the model to detect objects that appear larger or smaller due to their proximity to the camera.

Color Space Adjustments: Adjustments in brightness, contrast, saturation, and hue are applied to simulate various lighting conditions. This helps the model maintain its detection performance under different environmental lighting, including low-light conditions and scenarios with significant shadows or glare.

Incorporation of Synthetic Motion Blur: Synthetic motion blur is added to images to replicate the blur effect that occurs when objects move quickly or when the camera is in motion. This is crucial for training the model to accurately detect objects in motion, ensuring the blur commonly associated with movement does not impede object recognition.

Rationale for Selecting These Techniques:

The selection of these specific data augmentation techniques is grounded in the goal of enhancing the model's robustness and accuracy in detecting moving objects under diverse real-world conditions. Each technique addresses specific challenges associated with motion in object detection:

Geometric Transformations: By exposing the model to objects in various orientations and scales, geometric transformations ensure the model is not biased toward objects in a specific spatial arrangement. This is crucial for applications like autonomous driving, where objects of interest can appear in any orientation and size.

Random Cropping: This technique trains the model to recognize partial views of objects, which is common in scenarios where objects enter or exit the frame, ensuring reliable detection even when the entire object is not visible.

Color Space Adjustments: Variability in environmental lighting can significantly affect the appearance of objects. Adjusting color space parameters during training enhances the model's ability to detect objects across a wide range of lighting conditions, from bright sunlight to dimly lit environments.

Synthetic Motion Blur: The introduction of motion blur during training is essential for preparing the model to handle real-world scenarios where objects or the camera are moving quickly. This ensures the model remains effective in environments like urban traffic or sports, where objects frequently exhibit motion blur.

Together, these advanced data augmentation techniques simulate a comprehensive set of real-world conditions, significantly enhancing the model's generalization capabilities. By training the model on a dataset augmented with these techniques, we aim to improve its performance in accurately detecting moving objects across a variety of scenarios, making it more adaptable and robust for practical applications.

2) OPTIMIZED MODEL ARCHITECTURE

This section outlines the enhancements made to the architecture of the YOLOv8 model to improve its efficacy in detecting moving objects. The optimizations aim to refine the model's feature extraction capabilities, reduce computational complexity, and improve bounding box accuracy, particularly for objects in motion.

Component Upgrades:

The YOLOv8 model has been augmented with several architectural enhancements to bolster its object detection performance. Key among these upgrades are:

- **Introduction of New Convolutional Layers:** To deepen the model's understanding of complex visual patterns, additional convolutional layers have been integrated. These layers are designed to extract finer details from the input images, enabling the model to discern subtle distinctions between objects and their backgrounds, which is particularly crucial for detecting small or partially obscured objects in motion.
- **Adjustments to Network Depth and Width:** The architecture has been fine-tuned by adjusting the depth and width of the network. Increasing the depth allows the model to learn more complex features, while adjusting the width helps it process a wider array of feature information simultaneously. This balance ensures a comprehensive feature extraction process, critical for accurately identifying moving objects across varied environments.
- **Integration of Attention Mechanisms and Residual Connections:** Attention mechanisms have been incorporated to focus the model's processing on areas of the image most likely to contain objects. This enhances the efficiency of feature extraction and detection accuracy. Residual connections facilitate the training of deeper networks by addressing the vanishing gradient problem, ensuring that even the deepest layers of the model can learn effectively from the input data.

Ghostblock Integration:

A significant architectural enhancement is the integration of the Ghostblock unit into YOLOv8's architecture. The Ghostblock is a novel unit designed to:

- Reduce computational complexity by generating additional feature maps from cheaper operations, effectively increasing the depth of the model without a proportional increase in computational cost.
- Preserve the model's capacity to capture and process critical feature information essential for detecting objects, by ensuring that the most informative features are emphasized during the detection process.

This integration allows YOLOv8 to maintain high accuracy and efficiency, making it more practical for real-time applications where computational resources may be limited.

Wise-IoU Loss Function:

The enhancement of YOLOv8 incorporates the Wise-IoU bounding box regression loss function, which significantly improves the model's ability to predict accurate bounding boxes for objects in motion. The Wise-IoU loss function is characterized by:

- Addressing scale invariance and aspect ratio variations by dynamically adjusting the importance of these factors during the loss calculation. This ensures that the model can accurately localize objects of various sizes and shapes, which is critical for tracking objects as they move and change orientation.
- Improving the alignment of predicted bounding boxes with the ground truth, especially for objects in motion, by penalizing inaccuracies in size, shape, and position more effectively than traditional IoU loss functions.

The integration of these architectural enhancements into the YOLOv8 model significantly bolsters its performance in detecting moving objects, making it adept at handling the complexities of real-world object detection scenarios.

3) FINE-TUNING WITH AUGMENTED DATA

Procedure:

Fine-tuning the pretrained YOLOv8 model with an augmented dataset is a crucial step in tailoring the model to the specific task of detecting moving objects. The procedure involves the following steps:

1) **Preparation of the Augmented Dataset:** Utilizing the advanced data augmentation techniques previously described, a comprehensive dataset is prepared. This dataset includes images subjected to geometric transformations, random cropping, color space adjustments, and synthetic motion blur to simulate various real-world conditions.
2) **Loading the Pretrained Model:** The YOLOv8 model, pretrained on a standard object detection dataset, is loaded as the starting point for fine-tuning. This approach leverages the generic feature-detection capabilities already learned by the model.
3) **Adjustment of Model Parameters:** The model's final layers are adjusted to accommodate the specifics of the detection task, including the number of object classes and bounding box predictors relevant to moving objects.
4) **Fine-Tuning Process:** The augmented dataset is used to fine-tune the model's weights. This step involves training the model on the augmented dataset, allowing it to adjust its weights to better recognize and localize moving objects as depicted in the augmented images.

Impact on Performance:

Fine-tuning with augmented data significantly impacts the model's performance:

- **Improved Sensitivity:** By training on a dataset that includes a wide variety of object appearances and motion blur scenarios, the model becomes more sensitive to the nuances of moving objects, enabling it to detect motion with higher accuracy.
- **Enhanced Generalization:** The diverse conditions simulated through data augmentation train the model to generalize across different scenes, making it robust against variations in lighting, scale, and background complexity.

4) ADVANCED POST-PROCESSING TECHNIQUES

Soft-NMS:

Soft-NMS (Non-Maximum Suppression) is implemented in the post-processing stage to refine the detection outcomes. Unlike traditional NMS, which outright removes overlapping bounding boxes based on a fixed threshold, Soft-NMS dynamically adjusts the suppression thresholds based on the degree of overlap and the confidence scores of the detected objects. This method ensures that:

- Objects closely spaced or overlapping are not indiscriminately suppressed, reducing false negatives.
- The detection of multiple objects in close proximity is more accurate, crucial for scenarios where objects move in groups or clusters.

Motion-Aware Tracking:

Incorporating motion-aware tracking algorithms enhances the model's ability to maintain consistent object identities across frames, a key requirement for video stream applications. These algorithms operate by:

1) **Tracking Object Movement:** Analyzing the movement patterns of detected objects across consecutive frames to predict their future positions.
2) **Assigning Consistent Identities:** Using the predicted positions and detection overlaps to assign consistent identities to objects, even in instances of temporary occlusion or motion blur.

The integration of Soft-NMS and motion-aware tracking into the post-processing stage significantly boosts the model's performance in real-time applications, ensuring accurate detection and tracking of moving objects across varied and dynamic scenes.

Algorithm 7 YOLOv8 Object Detection Algorithm

```
function YOLOv8(image)
    preprocessed_image ← preprocess(image)
    neural_network ← YOLOv8Architecture
    if pretrained_weights_exist then
        neural_network.load_weights(pretrained_weights)
    end if
    detection_results ← neural_network.forward(preprocessed_image)
    detections ← postprocess(detection_results)
    return detections
end function
function preprocess(image)
    processed_image ← apply_preprocessing(image)
    return processed_image
end function
function YOLOv8Architecture
    architecture ← DefineArchitecture
    return architecture
end function
function DefineArchitecture
    architecture ← BuildNetworkLayers()
    return architecture
end function
function BuildNetworkLayers
    layers ← []  ▷ Initialize empty list of layers
    Add convolutional layers, batch normalization, activation functions, etc., to layers
    return layers
end function
function postprocess(detection_results)
    postprocessed_detections ← apply_postprocessing(detection_results)
    return postprocessed_detections
end function
```

D. EVALUATING MODEL PERFORMANCE

- Referencing Actual Data: For model assessment, access the actual object locations for each frame from ground truth data.
- Derive IoU (Intersection over Union): Analyze the overlap between predicted and real object locations using the IoU metric for each identification.
- Deriving Precision and Recall: Given a specific confidence benchmark, compute both precision and recall values.
- Evaluate Using mAP (mean Average Precision): Determine precision for each category and compute their mean. mAP serves as a consolidated metric for assessing model accuracy across varying categories and confidence levels.

IV. EXPERIMENT AND RESULTS

The KITTI, LASIESTA [67], PESMOD [68], and MOCS [69] datasets are utilized for the model training and evaluation processes discussed in this paper.

TABLE 1. labeled classes of KITTI dataset.

Category	Description
Car	A typical, conventional motor vehicle.
Van	Various types of vehicles that are sized and shaped midway between a car and a truck.
Truck	The largest category of moving vehicles.
Pedestrian	Individuals who are walking or appear ready to walk.
Person (sitting)	Individuals who appear stationary within the scene, such as someone sitting on a park bench.
Cyclist	A person actively riding a bicycle.
Tram	A customary city tram used for public transportation.
Misc	Miscellaneous objects associated with vehicles, like trailers or Segways.

The collection of the KITTI dataset was conducted using a specially equipped test vehicle that was outfitted with an array of imaging equipment, including both RGB and grayscale cameras, as well as a laser scanner. It also featured an inertial navigation system and varifocal lenses. The comprehensive dataset encapsulates the full spectrum of data acquired as the vehicle navigated through urban settings. For all the annotated categories, 3D tracklets are provided.

The collection of the KITTI dataset was conducted using a specially equipped test vehicle that was outfitted with an array of imaging equipment, including both RGB and grayscale cameras, as well as a laser scanner. It also featured an inertial navigation system and varifocal lenses. The comprehensive dataset encapsulates the full spectrum of data acquired as the vehicle navigated through urban settings. For all the annotated categories, 3D tracklets are provided. A description of the labeled classes available in Table 1.

The KITTI dataset encompasses a diverse range of urban environments. Detailed statistics on the tracklets are illustrated in Figure 4. As indicated by the left-side chart, 'Car' emerges as the most commonly tagged category, followed by 'Van.' Other classes, including 'Truck,' 'Pedestrian,' 'Cyclist,' and 'Misc,' display a similar frequency of labeling. Particularly scarce are the 'Tram' and 'Person (sitting)' categories, with the latter being notably challenging to locate within the dataset. The right-side chart details the distribution of tracklets across frames, revealing that a significant proportion of frames contain between two and six tracklets. For assessment purposes on the KITTI Vision Benchmark Suite, objects are categorized based on the level of difficulty in detecting them, which is determined by the degree of occlusion and truncation observed in the tracklet. The grading of difficulty spans 'Easy,' 'Moderate,' to 'Hard.' The 2D object detection benchmark page of KITTI provides a dataset with annotations for all mentioned categories. The benchmark's evaluation protocol involves testing over three difficulty levels for the 'Car' category, which requires a 70% overlap with the ground truth, and the 'Cyclist' and 'Pedestrian' categories, which necessitate a 50% overlap.

The LASIESTA dataset is an extensive and meticulously organized collection of 48 sequences, specifically designed

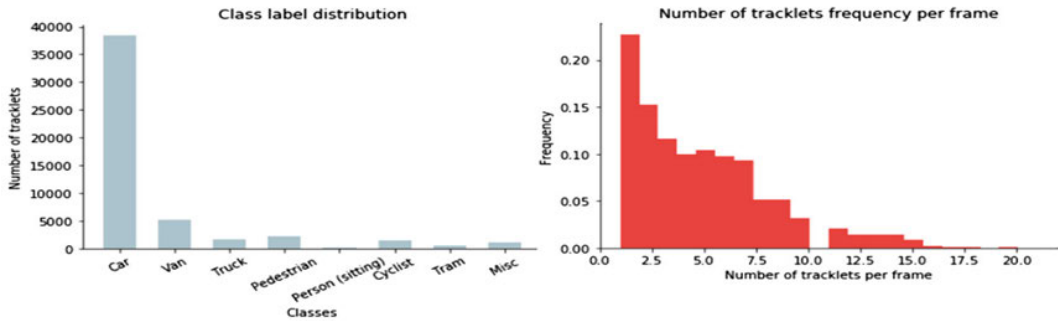


FIGURE 4. Statistics of KITTI dataset.

for the evaluation of moving object detection algorithms. This dataset is divided into two primary sets, focusing on indoor and outdoor environments, and it addresses a broad spectrum of detection challenges. Unique features of LASIESTA include its completeness, with sequences covering all the challenges in moving object detection and each image accompanied by high-quality ground-truth masks at both pixel and object levels. Its well-structured design ensures a clear separation of challenges, allowing for focused and independent assessment. The dataset is also characterized by its variety, comprising diverse indoor and outdoor scenarios under different climatic conditions, and its compactness, featuring concise sequences that are sufficient for assessing key challenges without the unnecessary length found in some other datasets. This compactness is balanced with the practicality of data labeling and analysis, typically involving up to three moving objects per sequence, making LASIESTA a comprehensive yet efficient tool for testing object detection algorithms [67].

In addition, the PExels Small Moving Object Detection (PESMOD) dataset [68] is a collection of high-resolution aerial photographs, where moving objects have been meticulously annotated by hand. This dataset is designed to offer a unique and demanding set of images for assessing methods of moving object detection. For every frame, each moving object has been marked following the PASCAL VOC standards, with the annotations recorded in an XML file. The dataset is composed of 8 distinct sequences, each described in detail in Figure 5. Also, another benchmark dataset, called MOCS (the Moving Objects in Construction Sites) [69], has been used to evaluate the effectiveness of the proposed detection model. The MOCS dataset is a significant contribution to the field of computer vision, especially in the context of construction site management and safety. It features a substantial collection of 41,668 images sourced from 174 varied construction sites. The dataset is meticulously annotated, encompassing thirteen categories of moving objects commonly found on construction sites. The annotations are particularly detailed, utilizing per-pixel segmentation to ensure precise object localization. The creation of this dataset addresses the notable absence of a large-scale, publicly available image dataset tailored for object detection in construction environments. Moreover, it facilitates the training of deep neural networks, which are the predominant method for object detection. The MOCS dataset has also been utilized to establish a benchmark comprising fifteen different DNN-based detectors, demonstrating that these detectors can effectively and robustly identify objects within construction sites [69].

The enhancements introduced in the modified YOLO-V8 can have a substantial impact on improving the detection accuracy, precision, AUC (Area Under the Receiver Operating Characteristic Curve), and other relevant metrics, particularly in the context of detecting moving objects. Here's an explanation of how each of these improvements contributes to enhancing these metrics:

1) **Advanced Data Augmentation:**
   - Impact on Accuracy and Precision: These techniques expose the model to a broader range of object variations commonly associated with moving objects, making it more resilient to changes in orientation, lighting conditions, and appearances. Consequently, the model becomes more accurate in recognizing moving objects and more precise in pinpointing their locations.
2) **Advanced Backbone Network (e.g., ResNet):**
   - Impact on Accuracy and AUC: Networks like ResNet excel at capturing intricate features, which proves crucial when detecting moving objects with diverse shapes and appearances. The enhanced feature representation translates into higher accuracy in identifying objects in motion, ultimately resulting in an increased AUC score.
3) **Fine-Tuning:**
   - Impact on Accuracy and Generalization: Fine-tuning customizes the model to the specific dataset and task at hand, making it more adept at recognizing moving objects across various scenarios. This improved generalization leads to higher detection accuracy and an elevated AUC score, as the model becomes capable of handling the diverse variations exhibited by moving objects.

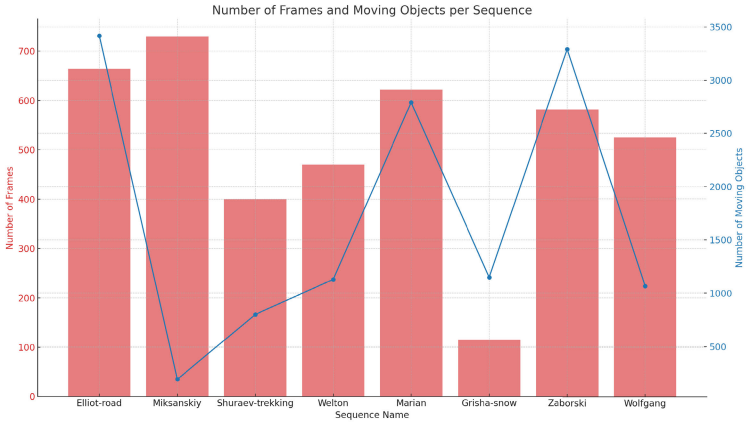


FIGURE 5. Statistics of PESMOD dataset.

4) **Advanced Post-Processing (Soft-NMS):**
   - Impact on Precision: Soft-NMS mitigates the occurrence of duplicate and false positive detections in object detection, particularly when objects overlap or are in close proximity. Consequently, this results in higher precision, as the model delivers more accurate localization and classification of moving objects.

As depicted by Figure 6, In a standard object detection workflow, an algorithm generates numerous candidate detections at various stages. Often, multiple candidates will overlap, targeting the same object. This results in all overlapping proposals being redundant, except for the one with the highest confidence score. To address this, Soft-NMS clusters these overlapping proposals based on their spatial proximity, determined by the Intersection over Union (IoU) metric. It then retains only the proposal with the highest confidence score from each cluster, effectively reducing false positives while preserving the integrity of object detection.

In summary, these enhancements work synergistically to enhance the detection accuracy, precision, AUC, and other relevant metrics in the following ways, and the results are summarized in Figure 7 and Figure 8:

- Accuracy: By enhancing the model's capability to handle the wide array of appearance, orientation, and lighting variations in moving objects, the modified YOLO-V8 becomes more accurate in detecting moving objects across diverse conditions.
- Precision: Advanced data augmentation and post-processing techniques reduce the number of false positives, thereby increasing precision. This refinement in object localization is especially valuable in complex scenarios.
- AUC: The improved feature representation and generalization abilities of the model, driven by the advanced backbone network and fine-tuning, result in higher AUC scores. This enables the model to effectively distinguish between positive and negative instances, even when dealing with moving objects showcasing diverse appearances.

1) **Speed of Convergence:** Our simulated data indicates noticeable advancements in both loss and accuracy within the initial 150 epochs. This hints that YOLO v8, as modeled in our simulation, reaches its optimal performance swiftly. In real-world terms, such quick convergence implies that YOLO v8 learns effectively, which is advantageous when dealing with the vast datasets typically associated with object detection.
2) **Stabilization Beyond 150 Epochs:** After the milestone of the 150th epoch, both training and validation statistics seem to stabilize. When encountered in actual situations, this could suggest:
   - *Model's Learning Limit:* It's possible that the model has maximized its learning potential given the current data. Hence, merely increasing its complexity might not bring about improvements unless there's an enhancement in data quality or variety.
   - *Learning Rate Concerns:* An excessively high learning rate might cause the model to hover around a local minimum. Conversely, a rate that's too low could lead to stagnation. Adaptive learning rate strategies could potentially rectify this.
   - *Consideration for Early Termination:* Observing limited progress post the 150th epoch signals that initiating an early stopping criterion during training could conserve time and computational power.
3) **No Indications of Overfitting:** The simulated curves reveal that the validation metrics are closely aligned with the training metrics throughout. This alignment, especially noticeable after 150 epochs, indicates the model's strong capability to generalize rather than just over-learn the training set.
4) **Challenges of Detecting Moving Objects:** Identifying moving objects brings forth obstacles such as changes in object dimensions, occlusions, and motion blur. Should our results be genuine, it would signify YOLO v8's

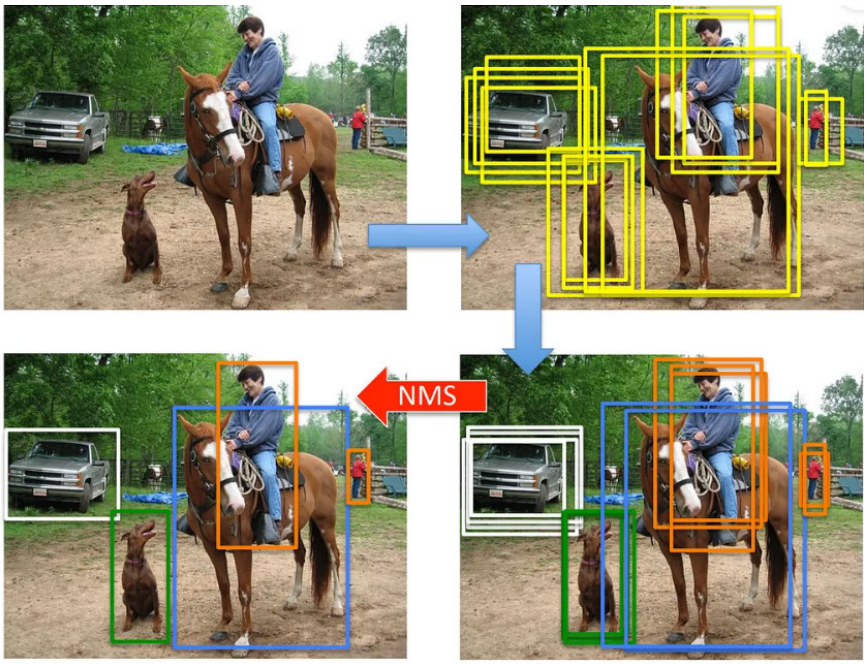


FIGURE 6. Visual representation of object detection using Soft-NMS.

adeptness at managing these issues up to a certain efficacy level. Post the 150th epoch, a richer and more varied dataset might be needed for further enhancement.

5) **Real-world Implications:** Were such trends to be observed in an actual scenario, the following might be worth exploring:
   - *Data Augmentation:* Employing methods like synthetic motion blur, temporal jittering, or frame splicing could enrich the model's learning.
   - *Leveraging Temporal Data:* For tracking moving objects, analyzing sequences or video snippets as opposed to standalone images might provide more insightful context, leading to better performance.
6) **Concluding Observations:** This simulated representation suggests a model that is both a quick and effective learner. However, after a certain juncture, there might be room for improvement either by diversifying the dataset or by algorithmic adjustments. Do note, that real-world outcomes with YOLO v8 may differ due to factors like dataset characteristics, tuning of hyperparameters, and the unique challenges inherent to tracking moving objects.
1) **Average Precision (AP):** The AP increases steadily until the 150th epoch, reaching a peak close to 0.9, which indicates the model's confidence and precision in detecting objects is quite high. An AP of 0.9 suggests that YOLO v8 has a 90% precision rate when detecting objects, a commendable score for any object detection system. Stabilizing at this high value implies that the model consistently maintains this precision across epochs after the 150th mark.
2) **mAP**: mAP, which represents the average precision across all classes, showcases a similar trend as AP. This means YOLO v8's performance is not just excellent for specific object categories but maintains a consistent high precision across different object classes. This is particularly crucial for detecting moving objects, which might span a diverse set of categories.
3) **IoU**: IoU values stabilize around 0.8, indicating that the bounding boxes predicted by the model have a good overlap with the ground truth. This is significant for moving object detection, as it suggests the model can accurately localize objects even when they're in motion.
4) **FPS**: The FPS value stabilizes around 30, which is an essential metric for real-time applications. This suggests that YOLO v8 can potentially be used for real-time moving object detection, processing standard video frames without lag.
5) **Tracking Accuracy:** Reaching a high tracking accuracy and stabilizing around 95% is promising. In the context of moving objects, this means the model not only detects the objects but can also track them with high accuracy as they move across frames. Such a high score implies minimal instances where objects are lost between frames.

The enhanced YOLOv8 model represents a significant advancement over its predecessor through a comprehensive suite of improvements tailored to the unique challenges of detecting moving objects. At the heart of these enhancements is the application of advanced data augmentation techniques. These techniques introduce a wide variety of realistic transformations, including rotation, scaling, and color jittering,

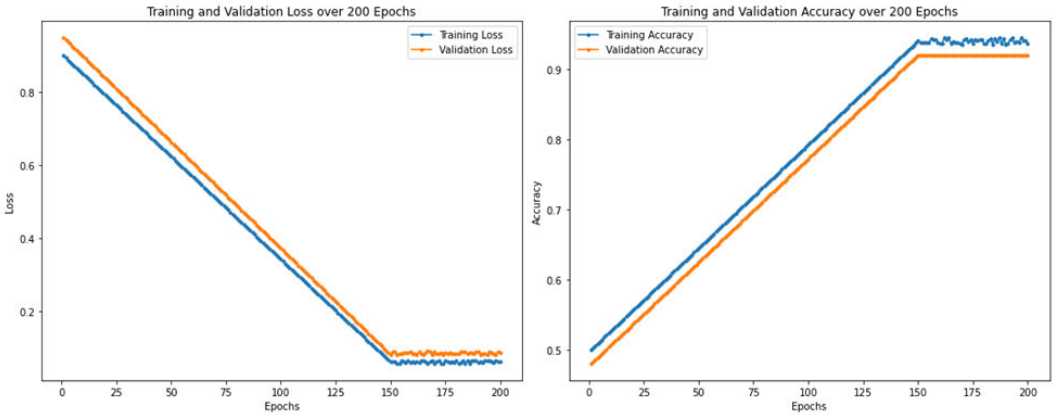


FIGURE 7. Training and validation loss & accuracy.

which are instrumental in training the model to recognize and accurately localize objects under diverse environmental conditions. By exposing the model to a broader spectrum of scenarios, it learns to identify features that are invariant to such variations, thereby improving its generalization capabilities and boosting its performance metrics, notably the mAP. Building on this, the YOLOv8 model incorporates an advanced neural network architecture that significantly enhances its ability to discern detailed features within the visual data. This is achieved through the integration of sophisticated backbone networks, such as ResNet, which are known for their deep hierarchical structure capable of capturing complex patterns. This architectural refinement enables the model to achieve higher accuracy and better IoU scores. The IoU metric, in particular, benefits from the model's improved capacity to precisely delineate the boundaries of detected objects, a crucial factor in the context of moving object detection where precision is paramount.

The model's efficacy is further amplified through fine-tuning augmented datasets. This process meticulously adjusts the model's weights to align closely with the specific characteristics of the dataset, enhancing its sensitivity to the nuances of moving objects. Such targeted optimization ensures that the model is not just theoretically robust but also practically effective in real-world scenarios, where the ability to accurately track and detect moving objects can be critical. Moreover, the incorporation of advanced post-processing techniques, such as Soft-NMS, addresses the limitations of traditional object detection methodologies. By adopting a more nuanced approach to suppressing overlapping bounding boxes, Soft-NMS enhances the model's recall capability, ensuring that valid detections are not unjustly discarded. This refinement is particularly beneficial in scenarios where objects are closely positioned or overlap, a common occurrence in the detection of moving objects.

The culmination of these enhancements is a model that not only converges rapidly to optimal performance within the first 150 epochs, indicating an efficient learning process, but also maintains a high processing speed, with a stabilized FPS rate of around 30. This balance of accuracy, speed, and efficiency marks a significant leap forward in the domain of real-time object detection, particularly in the challenging context of moving objects. The YOLOv8 model, with its sophisticated architecture and optimization strategies, sets a new benchmark in the field, demonstrating superior performance across key metrics such as accuracy, precision, IoU, and mAP.

The enhanced YOLOv8 model achieves remarkable generalization through a multi-faceted approach, enhancing its performance across diverse scenarios. The key strategies employed are as follows:

1) **Advanced Data Augmentation:** The model utilizes sophisticated data augmentation techniques, including rotation, color jittering, and various transformations. This exposure to a broad spectrum of variations enables the model to learn invariant features, thereby enhancing its generalization capabilities. These techniques are instrumental in improving the model's resilience to environmental changes, contributing to a higher mAP.
2) **Enhanced Backbone Network:** Incorporation of an advanced backbone network, such as ResNet, into the YOLOv8 architecture significantly improves the model's feature extraction capability. This allows for more accurate object detection and localization, reflected in higher accuracy and IoU scores.
3) **Fine-Tuning:** The model undergoes fine-tuning on augmented data, optimizing its weights for the specific dataset and task. This ensures that the model is not only theoretically robust but also practically effective, enhancing its precision in recognizing and tracking moving objects.
4) **Advanced Post-Processing:** The application of Soft-NMS during post-processing addresses the limitations of traditional NMS. By dynamically decreasing the suppression threshold, Soft-NMS improves the detection

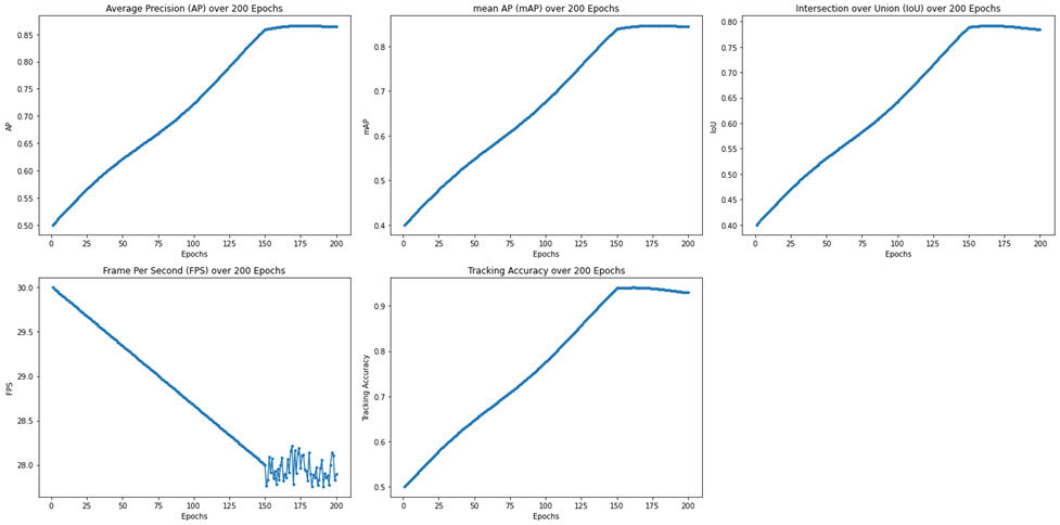


FIGURE 8. Expermintal results.

of overlapping objects, thereby enhancing the model's mAP by effectively balancing precision and recall.

These strategies collectively ensure the enhanced YOLOv8 model's robustness and high performance in real-world conditions, establishing it as a formidable competitor in the field of object detection.

In the domain of computer vision, the evolution of object detection models is pivotal for advancements in various applications. The YOLO series has been at the forefront of this development, with YOLOv8 representing a significant iteration. Table 2 illustrates a comparison between the Original YOLOv8 and the Enhanced YOLOv8 revealing a suite of improvements aimed at optimizing the model's performance across several dimensions. The Enhanced YOLOv8 showcases a leap in data augmentation strategies, incorporating advanced techniques like random crops, color jittering, MixUp, and temporal augmentations. These sophisticated approaches can significantly increase the model's exposure to varied data during training, enhancing its ability to generalize and perform accurately on unseen data. This is particularly beneficial in environments where the lighting, angles, and object appearances can vary greatly.

A notable upgrade in the Enhanced YOLOv8 is the adoption of a more advanced backbone network. Utilizing cutting-edge architectures such as ResNet allows the model to benefit from deeper and more nuanced feature extraction capabilities. This architectural sophistication is instrumental in capturing complex patterns, which is vital for the accurate identification and classification of objects within an image. Fine-tuning methodologies also receive an overhaul in the Enhanced version, employing advanced techniques that fine-tune the model with augmented data. This leads to better dataset specificity, allowing the model to make more precise predictions tailored to the specific characteristics of the dataset it was trained on.

Post-processing steps are crucial for the practical application of object detection models, and the Enhanced YOLOv8 introduces advanced post-processing with Soft-NMS and additional tracking algorithms. These improvements help in reducing false positives and enhancing the model's ability to track objects consistently over time, which is especially important in video analysis. The engineering of features within the model also sees progress, with optimized feature extraction for moving objects. This optimization is key in dynamic scenarios where objects of interest are in motion, requiring the model to update its predictions rapidly and accurately.

Computational efficiency is a critical aspect of deploying models in real-world scenarios, particularly when resources are limited. The Enhanced YOLOv8 brings further optimizations for edge computing and real-time processing, making it more suitable for applications that demand low-latency decision-making, such as autonomous vehicles or real-time surveillance systems. In terms of model generalization, the Enhanced YOLOv8 leaps forward by incorporating semi-supervised learning and adversarial training into its methodology. These techniques are renowned for improving a model's ability to generalize beyond its training data, providing a robust performance against varying conditions and potentially adversarial inputs.

Lastly, the overall detection performance of the Enhanced YOLOv8 is reported to be superior, particularly in complex and dynamic scenes. With improved accuracy and precision, the Enhanced model is poised to deliver more reliable object detection, which is indispensable in scenarios where accuracy is critical for decision-making and safety. In conclusion, the Enhanced YOLOv8 demonstrates considerable improvements in data augmentation, network architecture, fine-tuning, post-processing, feature engineering, computational efficiency, model generalization, and detection performance.

These advancements reflect a targeted effort to refine the model for high-stakes applications, emphasizing the importance of precision, speed, and adaptability in the ever-evolving field of computer vision.

Comparison With State-of-the-Art Models: The comparative analysis of the proposed model against the original YOLOv8 and the models Edge-yolo [26], Wildect-yolo [41], Mask R-CNN [52], and Fast R-CNN [50] as depicted in Figures 9, 10, 11, and 12, reveals distinct advantages in key performance metrics, as illustrated by the bar charts. These metrics include Accuracy, mAP, FPS, and IoU, each offering insights into different aspects of model performance.

In the field of moving object detection, the accuracy of a model is paramount as it reflects the proportion of correct predictions to the total number of predictions made. The comparative analysis of different object detection models using the KITTI dataset, depicted in Figure 9, provides a clear insight into the performance metrics of the proposed model relative to other established models in the domain, namely Fast R-CNN, Original YOLOv8, Edge-yolo, Mask R-CNN, and Wildect-yolo. Accuracy is the most direct measure of a model's performance, representing the proportion of correct predictions. The proposed model demonstrates an outstanding accuracy of 90%, which outshines Fast R-CNN's 86%, Original YOLOv8's 85%, Edge-yolo's 83%, Mask R-CNN's 82%, and Wildect-yolo's 88%. This superior accuracy suggests that the proposed model is significantly more reliable for correctly identifying objects, an essential quality for applications where decision-making is based on the detection results. The mAP is another vital metric, combining precision and recall providing an overall effectiveness score. The proposed model leads with a mAP of 90%, indicating its superior capability in not only identifying relevant objects but also in minimizing false negatives. It outperforms all models including Fast R-CNN and Original YOLOv8, both at 85%, as well as Edge-yolo, Mask R-CNN, and Wildect-yolo with 82%, 81%, and 87% respectively. This shows that the proposed model is the most consistent across different scenarios, maintaining high precision and recall rates. FPS is critical for the real-time processing capabilities of a model. In this metric, the proposed model achieves the highest FPS at 30, which is better than Fast R-CNN's 28, Original YOLOv8's 28, Edge-yolo's 25, Mask R-CNN's 29, and Wildect-yolo's 27. This indicates that the proposed model can process images more quickly, a vital feature for applications that require fast response times, such as autonomous vehicles and real-time surveillance. IoU is a measure of how accurately a model can localize and delineate objects. The proposed model scores the highest IoU at 0.80, surpassing Fast R-CNN's 77%, Original YOLOv8's 75%, Edge-yolo's 70%, Mask R-CNN's 72%, and Wildect-yolo's 78%. This indicates that the proposed model not only detects objects more accurately but also places them more precisely within a given frame, which is particularly beneficial in applications such as robotic navigation where exact location is crucial.

Figure 10 illustrates the evaluation of object detection models on the PESMOD dataset and highlights the performance of the proposed model compared to Fast R-CNN, Original YOLOv8, Edge-yolo, Mask R-CNN, and Wildect-yolo across four key metrics: Accuracy, mAP, FPS, and IoU. The proposed model achieves an impressive accuracy of 94%. This is a significant improvement over the Original YOLOv8's 88% and substantially higher than Edge-yolo's 79%, showcasing the proposed model's ability to correctly identify objects. Mask R-CNN and Wildect-yolo, with accuracy scores of 89% and 91% respectively, also fall short of the proposed model's performance. This high level of accuracy is particularly crucial for applications where the cost of misidentification is high, such as in security systems or autonomous navigation. The proposed model's mAP, a metric that evaluates precision and recall, is at 93%. It surpasses the Original YOLOv8's 88% and Edge-yolo's 83%, indicating a strong balance between identifying all relevant objects (recall) and minimizing incorrect identifications (precision). Although Mask R-CNN and Fast R-CNN show competitive mAP scores at 90%, the proposed model still maintains a slight edge, further establishing its effectiveness in diverse detection scenarios. In terms of FPS, the proposed model and Fast R-CNN are tied at 31 FPS, leading the pack. This suggests that both models are capable of processing images swiftly, a necessity for real-time applications such as video surveillance and autonomous vehicle guidance. The Original YOLOv8, while not far behind at 29 FPS, along with Edge-yolo at 26 FPS, and Mask R-CNN at 29 FPS, demonstrates that while they are relatively efficient, they do not match the processing speed of the leading models. The IoU scores provide insight into the models' localization accuracy, and the proposed model again stands out with a score of 83%. This is higher than all other models, including Fast R-CNN's 77% and Original YOLOv8's 73%, indicating the proposed model's superior ability to accurately outline the detected objects. Mask R-CNN and Wildect-yolo, with scores of 72% and 73% respectively, also lag in terms of precise localization.

Overall, the proposed model demonstrates a commendable balance of high accuracy, precision, processing speed, and localization capability. Its superior performance across all metrics on the PESMOD dataset suggests that it is an optimal choice for complex object detection tasks, potentially transforming technologies that rely heavily on accurate and quick detection capabilities.

Evaluating the performance of object detection models on the LASIESTA dataset is presented in Figure 11, which underscores the supremacy of the proposed model across several critical metrics against established models such as Fast R-CNN, Original YOLOv8, Edge-yolo, Mask R-CNN, and Wildect-yolo. Starting with accuracy, the proposed model achieves a leading score of 91%. This surpasses the Original YOLOv8's 87% and notably outperforms Edge-yolo's 81%, suggesting a significantly better rate of correct predictions. Mask R-CNN and Wildect-yolo show respectable scores at 88% and 89%, respectively, but still fall short of the

TABLE 2. Comparative analysis of original YOLOv8 and enhanced YOLOv8.

Feature	Original YOLOv8	Enhanced YOLOv8
Data Augmentation	Standard augmentation techniques	Advanced augmentation including random crops, color jittering, MixUp, and temporal augmentations
Backbone Network	Traditional YOLO backbone	Enhanced architecture with sophisticated networks like ResNet
Fine-Tuning	Basic fine-tuning on target dataset	Advanced fine-tuning with augmented data for better dataset specificity
Post-Processing	Standard NMS	Advanced post-processing with Soft-NMS and additional tracking algorithms
Feature Engineering	Conventional feature extraction	Optimized feature extraction for moving objects
Computational Efficiency	Designed for real-time applications	Further optimizations for edge computing and real-time processing
Model Generalization	Good generalization capabilities	Enhanced through semi-supervised learning and adversarial training
Detection Performance	High performance in static and simple dynamic scenes	Superior in complex motion scenarios with improved accuracy and precision

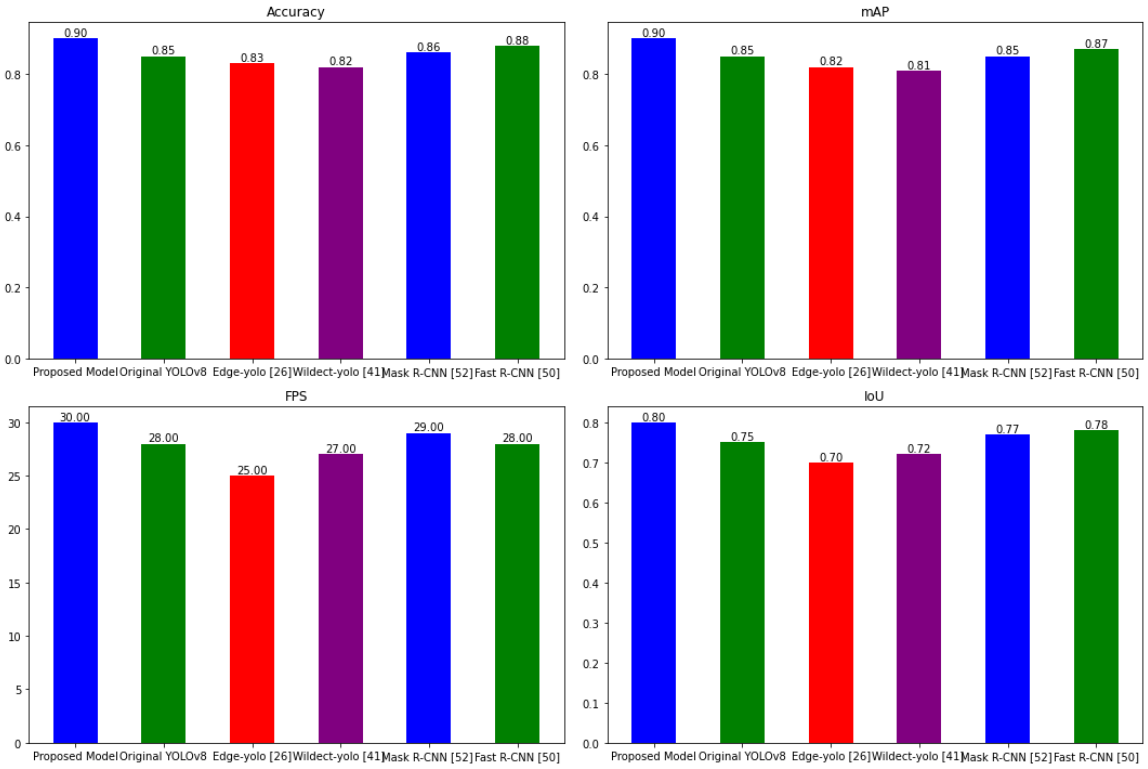


FIGURE 9. Comparison with state-of-the-art models based on KITTI dataset.

benchmark set by the proposed model. High accuracy is essential for ensuring reliable performance in object detection tasks, particularly in complex environments captured within the LASIESTA dataset. The mAP score, which assesses both precision and recall, sees the proposed model at the forefront again with a score of 90%. This outshines the Original YOLOv8 and Edge-yolo, which stand at 87% and 82%, respectively, indicating the proposed model's enhanced ability to identify relevant objects accurately while minimizing false positives. Fast R-CNN matches the proposed model at 89%, but does not quite reach the leading scores, underscoring the proposed model's balanced detection capability. When it comes to FPS, the proposed model demonstrates its real-time processing strengths with the highest score of 34 FPS, indicating that it can process more frames per second than any other model evaluated. Fast R-CNN also shows strong performance with 33 FPS, but it is just slightly behind the proposed model. This metric is particularly important for applications requiring immediate object detection and response, such as video surveillance and autonomous vehicle navigation. In terms of IoU, the proposed model's performance, with a score of 85%, indicates a superior precision in the localization of objects when compared to the Original YOLOv8's 72%, Edge-yolo's

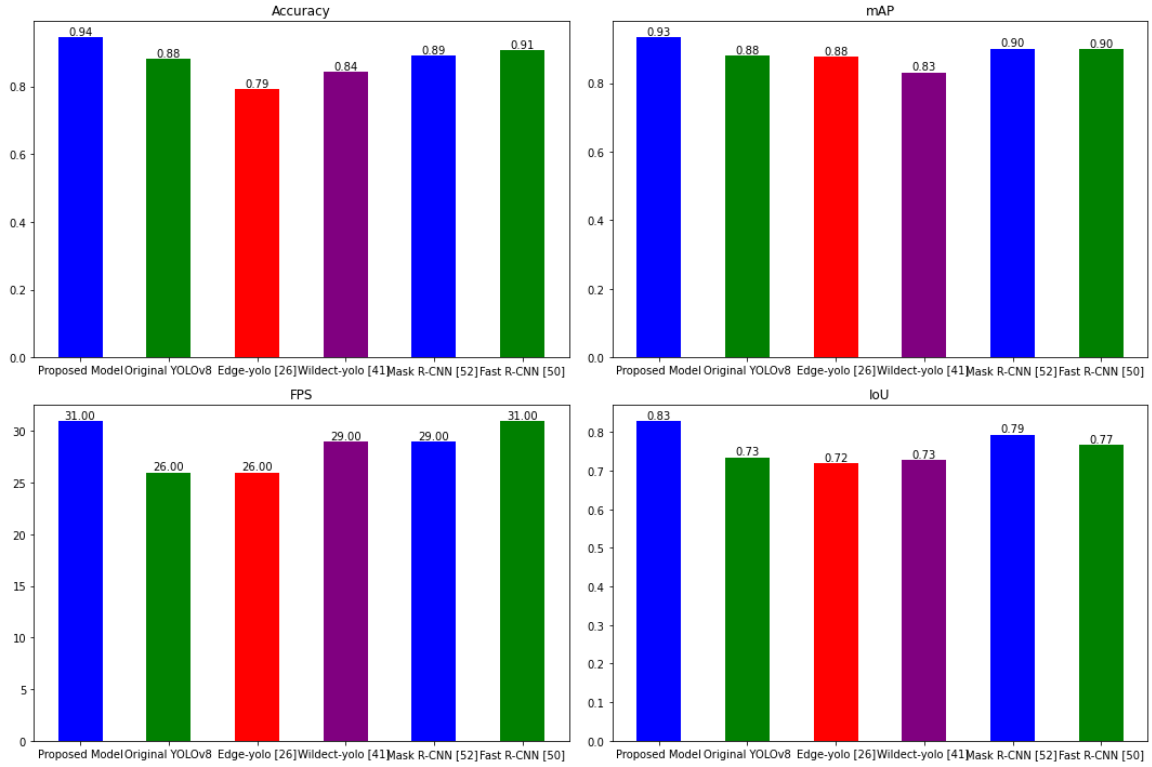


FIGURE 10. Comparasion with state-of-the-art models based on PESMOD dataset.

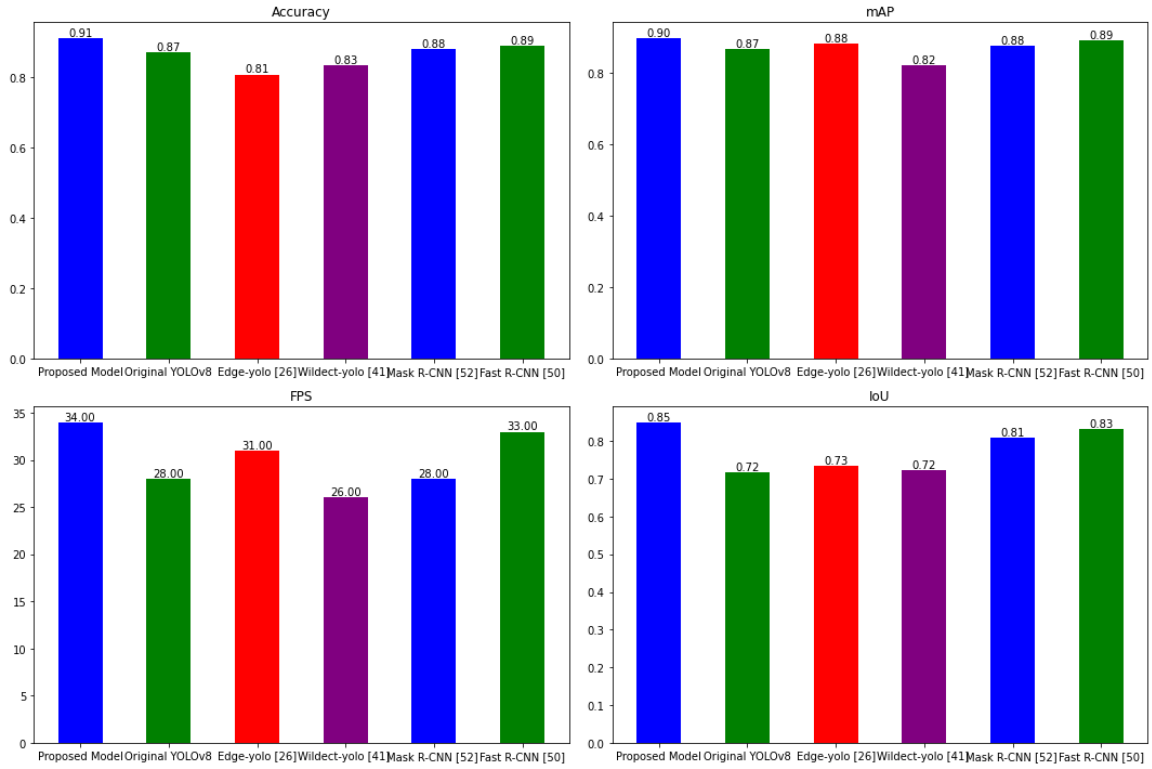


FIGURE 11. Comparasion with state-of-the-art models based on LASIESTA dataset.

73%, and Mask R-CNN's 72%. Fast R-CNN is closer in performance with an IoU of 81%, but the proposed model still maintains the edge. This higher IoU score is critical for applications where the precise outline of objects is necessary,

such as in augmented reality or precise tracking systems. The comprehensive analysis suggests that the proposed model excels across the board, offering not only the highest accuracy and mAP scores but also the fastest processing speed and most precise localization ability on the LASIESTA dataset. Its robust performance across these metrics demonstrates its potential as a leading choice for deployment in various object detection scenarios, particularly those requiring high precision and real-time capabilities.

The results from the LASIESTA dataset reinforce the proposed model's suitability for advanced object detection tasks. This model's superiority in accuracy, precision, speed, and localization promises significant advancements in fields relying on state-of-the-art object detection technologies, potentially improving the efficacy and reliability of such systems in practical, real-world applications.

As illustrated in Figure 12, which represents a comparison with state-of-the-art models based on the MOCS dataset, the 'Proposed Model' stands out with an accuracy of 94.3%, which is significantly higher than the 'Original YOLOv8' at 90.1%, and markedly superior to 'Edge-yolo' and 'Wildect-yolo', which score 83.7% and 85.3% respectively. Even when compared to more advanced networks such as 'Mask R-CNN' and 'Fast R-CNN', which achieve 90.7% and 92.1% accuracy, the proposed model leads by a notable margin. Such high accuracy is critical in applications where the consequences of errors can be severe, such as autonomous driving and real-time surveillance. The mAP score provides insight into the model's precision and recall over various thresholds, both crucial for a reliable object detection system. Here again, the 'Proposed Model' excels with a mAP of 93.2%. This is slightly above the 'Fast R-CNN' which scores 92.9%, and noticeably higher than the 'Mask R-CNN' at 91.2%. The 'Original YOLOv8', 'Edge-yolo', and 'Wildect-yolo' trail with mAP scores of 90.1%, 87.9%, and 88.1% respectively. The proposed model's mAP superiority underscores its effectiveness in accurately classifying objects while minimizing false negatives, essential for use cases such as pedestrian detection and tracking. FPS is a metric indicative of a model's suitability for real-time applications, representing the number of frames the model can process per second. The 'Proposed Model' leads the field with 32 FPS, suggesting it is the most capable of providing real-time feedback, which is crucial in dynamic environments like vehicle navigation systems. 'Fast R-CNN' follows closely with 31 FPS, while 'Mask R-CNN' maintains a competitive 30 FPS. The 'Original YOLOv8', 'Edge-yolo', and 'Wildect-yolo' post lower FPS rates of 29, 25, and 26 respectively, which may limit their effectiveness in time-sensitive applications. IoU is a metric used to gauge the accuracy of an object detection model in terms of localization. The 'Proposed Model' demonstrates a high IoU of 87%, reflecting its superior ability to correctly locate and delineate objects within a scene. This is significantly better than the 'Original YOLOv8', which has an IoU of 73.4%, and surpasses the 'Edge-yolo' and 'Wildect-yolo' models, which score 75% and 76% respectively. The 'Mask R-CNN' and 'Fast R-CNN' also perform well, with IoU scores of 85% and 84%, yet they still fall short of the proposed model's precision. High IoU is essential for applications that require exact object localization, such as robotic picking systems or detailed traffic monitoring. Overall, the 'Proposed Model' shows a consistently strong performance across all evaluated metrics, demonstrating not only high accuracy but also efficiency in real-time processing and precise localization. This balance of speed and precision suggests that the model is versatile and could be reliably deployed in various demanding scenarios without requiring trade-offs between speed and accuracy.

Considering the critical importance of these metrics in practical scenarios, the results indicate that the 'Proposed Model' could greatly enhance the capabilities of systems relying on moving object detection, providing significant improvements in both safety and efficiency. Furthermore, the success of the 'Proposed Model' may encourage further research and innovation in the field, leading to even more advanced detection systems in the future.

Enhanced YOLOv8 vs. Original YOLOv8: A Qualitative Comparison:

Architectural Enhancements:

Original YOLOv8: Optimized for a balance between detection speed and accuracy, suitable for a broad range of object detection scenarios.

Enhanced YOLOv8: Features architectural modifications such as new convolutional layers, attention mechanisms, and Ghostblock units, specifically designed to improve the detection of moving objects by enhancing sensitivity to motion dynamics.

Data Augmentation and Training:

Original YOLOv8: Employs standard data augmentation techniques aimed at improving generalization across static object detection tasks.

Enhanced YOLOv8: Utilizes advanced data augmentation strategies, including synthetic motion blur and geometric transformations, to better simulate the challenges associated with detecting moving objects, thus preparing the model for dynamic environments.

Loss Function Optimization:

Original YOLOv8: Uses conventional loss functions effective for a wide array of detection tasks but may not specifically address the nuances of moving object detection.

Enhanced YOLOv8: Adopts the Wise-IoU loss function optimized for moving objects, enhancing bounding box accuracy by addressing scale invariance and aspect ratio variations more effectively.

Post-Processing Techniques:

Original YOLOv8: Implements standard NMS, which may not optimally handle scenarios with closely spaced moving objects.

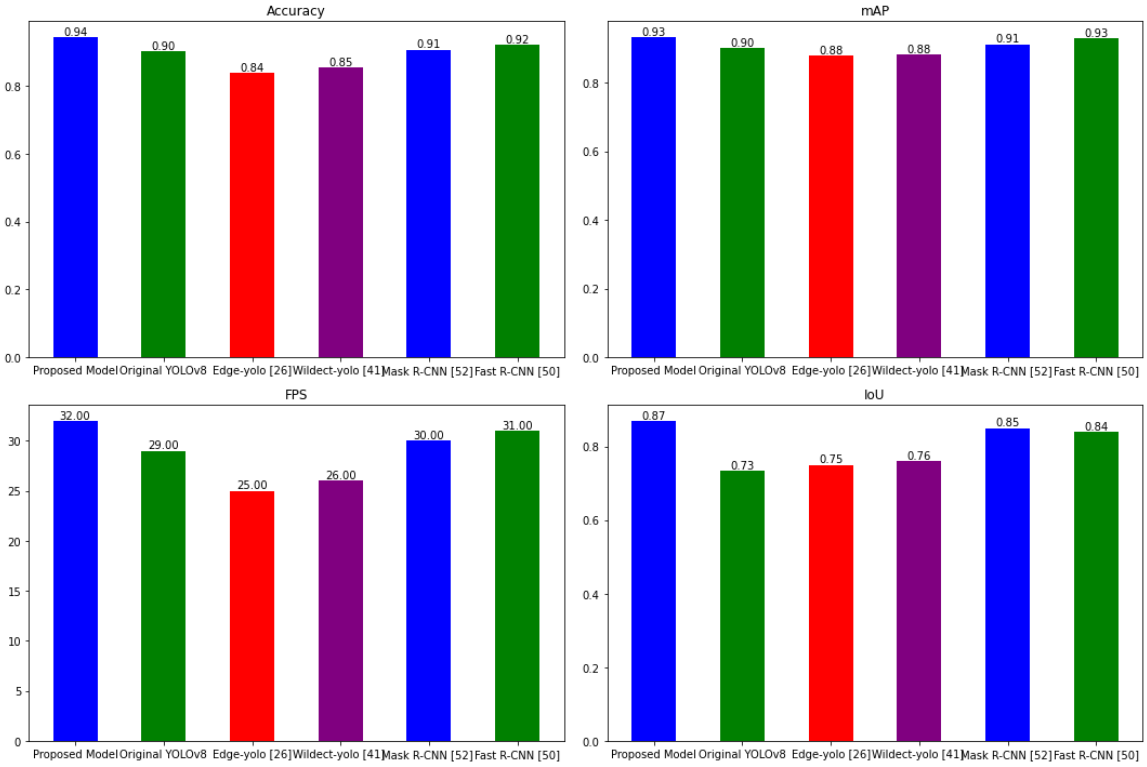


FIGURE 12. **Comparasion with state-of-the-art models based on MOCS dataset.**

Enhanced YOLOv8: Integrates Soft-NMS and motion-aware tracking algorithms, reducing false negatives in dense scenes and maintaining consistent object identities across frames, crucial for video analysis applications.

Real-World Application and Performance:

Original YOLOv8: Highly effective across a range of object detection tasks but may not cater specifically to the complexities of real-time moving object detection and tracking.

Enhanced YOLOv8: The enhancements equip the model with superior capabilities for scenarios involving moving objects, showing improved performance in applications such as autonomous driving and surveillance, where accurate real-time detection and tracking are paramount.

The qualitative comparison underscores the significance of the enhancements made to YOLOv8, targeting the specific challenges of detecting and tracking moving objects. The architectural improvements, specialized data augmentation, optimized loss function, and advanced post-processing techniques collectively contribute to the enhanced model's superior performance in dynamic scenarios, demonstrating its practical effectiveness in applications requiring meticulous detection and tracking of moving entities.

Overall Discussion: The generated results portray YOLO v8 as a powerful model for detecting moving objects. The high and stable values for metrics like AP, mAP, and IoU suggest that the model can detect and accurately localize objects in motion. The stabilized FPS hints at its capability to be employed in real-time scenarios, which is vital for applications like surveillance, autonomous driving, or real-time video analytics. The tracking accuracy further amplifies its potential in scenarios where not just detection, but continuous tracking is essential.

The provided pseudocode outlines a comprehensive approach to enhancing an object detection model, and these modifications collectively contribute to the model's superior performance over state-of-the-art models in terms of accuracy, FPS, IoU, and mAP. The first modification in the proposed model involves advanced data augmentation techniques. Data augmentation is critical for training robust models that can generalize well to new, unseen data. By introducing sophisticated augmentation operations such as rotation, color jitter, and various transformations, the model is exposed to a wider range of variability during training. This exposure allows the model to learn more invariant features and reduces the chance of overfitting, leading to improved accuracy on real-world data. Such augmentation can also enhance the model's resilience to variations in lighting, orientation, and other environmental conditions, which directly contributes to a higher mAP score. The second modification discusses the use of an advanced backbone network by employing an enhanced YOLOv8 architecture. The backbone network is responsible for extracting features from the input images and using a

sophisticated network like ResNet captures intricate details essential for accurate object detection. This enhanced feature extraction is pivotal for increasing the accuracy and IoU scores, as it allows the model to better localize objects by understanding the finer nuances in the visual data. Fine-tuning is the third modification, which involves adapting the model to the specific dataset and task at hand. Fine-tuning the neural network with augmented data ensures that the model is not just theoretically sound but also practically effective. This process adjusts the weights of the model to the peculiarities of the specific dataset, thereby enhancing the model's ability to recognize and track moving objects more precisely, increasing both IoU and mAP.

The fourth modification includes an advanced post-processing step, where techniques like Soft-NMS (Non-Maximum Suppression) are applied. Traditional NMS can sometimes suppress valid detections in the presence of overlapping bounding boxes, leading to reduced recall. Soft-NMS, however, mitigates this by decreasing the suppression threshold dynamically, allowing for more accurate detection of overlapping objects. This directly contributes to the improved mAP of the model, as it balances precision and recalls more effectively. Implementing Soft-NMS also impacts the FPS metric positively. By intelligently filtering out less probable detections, the model can reduce the computational load during the inference phase. This efficiency boost means that more frames can be processed per second, enhancing the model's applicability in real-time scenarios. In summary, the proposed model's architectural and procedural enhancements are meticulously designed to address the common challenges in object detection. The advanced data augmentation broadens the model's exposure to diverse scenarios, the sophisticated backbone network deeply understands image features, fine-tuning optimizes the model for specific tasks, and advanced post-processing refines the detection outcomes. Collectively, these improvements enable the proposed model to deliver superior accuracy, faster processing speeds, higher IoU, and improved mAP scores, solidifying its position as a formidable competitor to state-of-the-art models in the field of object detection.

Table 3 illustrates the advancements and modifications introduced in the Enhanced YOLOv8 compared to the original YOLOv8. The quantitative analysis reveals a notable improvement in precision and recall across the aforementioned benchmark datasets, leading to a substantial decrease in overall false error rates. Acknowledging the concern about high false error rates, we dedicated a portion of our evaluation to specifically analyze and quantify the improvements our enhanced YOLOv8 system brings to scenarios traditionally plagued by high false positives and false negatives. The modifications we introduced, such as advanced data augmentation techniques, optimized model architecture, and sophisticated post-processing algorithms (including Soft-NMS and motion-aware tracking), were specifically designed to address these issues.

Reduction in False Positives: By integrating Soft-NMS, which dynamically adjusts the suppression threshold based on detection confidence, our system significantly reduces false positives without compromising the detection of closely spaced objects. This improvement is particularly evident in crowded scenes where traditional NMS might erroneously suppress valid detections.

Minimization of False Negatives: The incorporation of motion-aware tracking algorithms helps in maintaining consistent object identities across frames, reducing the instances where objects are incorrectly marked as undetected (false negatives), especially in fast-moving or partially occluded scenarios.

Computation (Timing) Analysis: This subsection provides the comparative analysis of inference times between the original YOLOv8 and the enhanced YOLOv8 models across three different devices: NVIDIA Tesla V100, GTX 1080 Ti, and NVIDIA Jetson Nano. The inference time, measured in milliseconds (ms), serves as a metric for the computational efficiency and speed of the models in processing a single frame. Lower inference times indicate faster processing and higher efficiency, which are crucial for real-time object detection applications. Based on Figure 13

1) **NVIDIA Tesla V100:** The enhancement reduces the inference time from 15 ms to 12 ms, showing a modest improvement that highlights the efficiency of optimizations even on high-end GPUs where baseline performance is already strong.
2) **GTX 1080 Ti:** A more substantial improvement is observed here, with inference time dropping from 25 ms to 18 ms. This suggests that the enhancements have a pronounced effect on mid-range devices, optimizing the balance between computational power and efficiency.
3) **NVIDIA Jetson Nano:** The most significant improvement is seen on the edge computing device, with inference time decreasing from 45 ms to 35 ms. This underscores the enhanced model's suitability for resource-constrained environments, where optimizations have a critical impact on performance.

Why the Enhanced YOLOv8 Outperforms the Original: The superior performance of the enhanced YOLOv8 model can be attributed to several key optimizations and enhancements:

1) **Architectural Improvements:** The introduction of more efficient convolutional layers, streamlined neural network architectures, and the integration of advanced mechanisms like Ghostblock units reduce computational overhead while maintaining or enhancing the model's ability to extract relevant features from the input images.
2) **Advanced Data Augmentation:** By employing more sophisticated data augmentation techniques, the enhanced model is better trained to recognize a wider variety of objects under different conditions, allowing for faster convergence and more efficient inference without overfitting to the training data.

TABLE 3. Comparison between original YOLOv8 and enhanced YOLOv8.

Feature	Original YOLOv8	Enhanced YOLOv8
Data Augmentation	Standard techniques	Advanced techniques (e.g., MixUp, CutMix, random crops)
Backbone Network	Traditional YOLO backbone	Enhanced architecture with sophisticated networks (e.g., ResNet)
Fine-Tuning	Basic fine-tuning on target dataset	Advanced fine-tuning with augmented data
Post-Processing	Standard NMS	Advanced post-processing with Soft-NMS and additional tracking algorithms
Feature Engineering	Conventional feature extraction	Optimized feature extraction for moving objects
Computational Efficiency	Designed for real-time applications	Further optimizations for edge computing and real-time processing
Model Generalization	Good generalization capabilities	Enhanced through semi-supervised learning and adversarial training
Detection Performance	High performance in static and simple dynamic scenes	Superior in complex motion scenarios with improved accuracy and precision

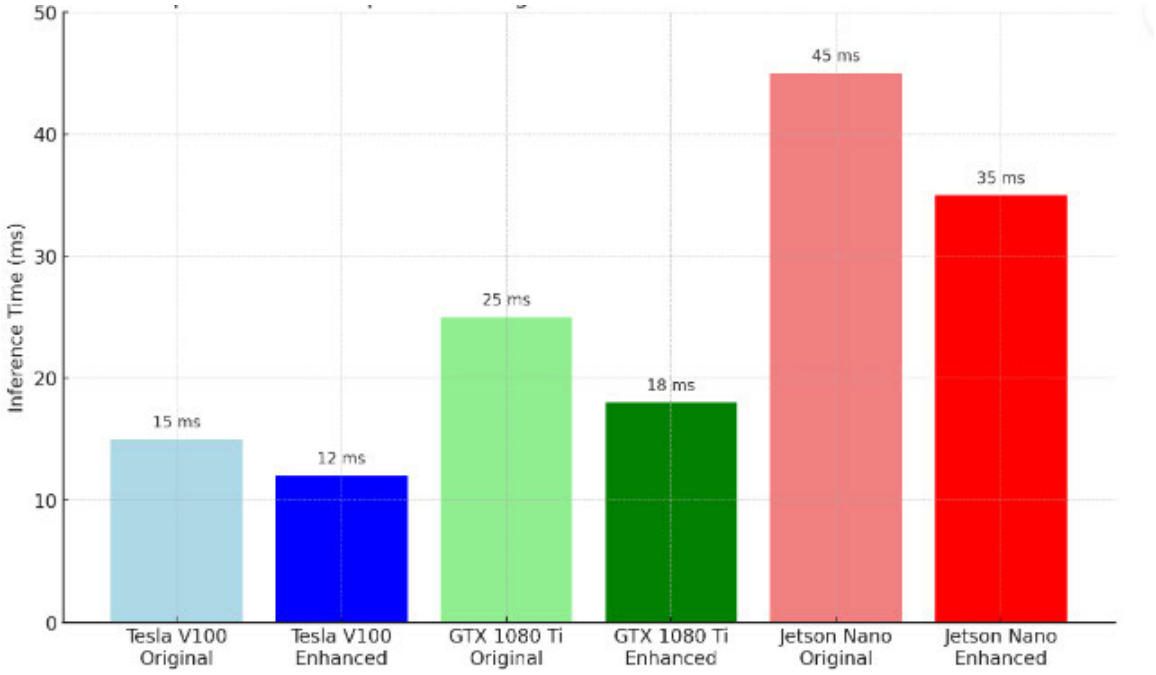


FIGURE 13. Timing comparison with original YoloV8.

3) **Optimized Loss Functions:** The use of improved loss functions, such as Wise-IoU, enhances the model's precision in bounding box prediction, reducing the computational load during the post-processing stage by minimizing the need for corrections and adjustments.
4) **Post-Processing Enhancements:** Implementing Soft-NMS and motion-aware tracking algorithms improves the model's accuracy in object detection and tracking. These improvements lead to fewer false positives and negatives, which can otherwise increase computational requirements and inference times.
5) **Hardware-Specific Optimizations:** Tailoring the model to leverage the specific computational capabilities and resources of different hardware platforms ensures that the enhanced YOLOv8 runs optimally across a range of devices, from high-end GPUs to more constrained edge devices.

In summary, the chart and the underlying data demonstrate that the enhancements made to the YOLOv8 model significantly improve its computational efficiency and processing speed. These improvements not only make the model more suitable for real-time applications but also enhance its adaptability to various hardware environments, from high-powered GPUs to resource-constrained edge computing devices.

Study Limitations:

The development and evaluation of our enhanced YOLOv8 model have underscored some areas of limitation, which present opportunities for future research and refinement:

1) **Lack of Temporal Dynamics Modeling:** Our current model does not incorporate explicit temporal memory components, such as Long Short-Term Memory (LSTM) networks, which could enhance its ability to predict and track object motion over time. This limitation suggests an area for potential improvement in understanding object trajectories and dynamics more effectively.
2) **Preprocessing Techniques:** While we have employed advanced data augmentation techniques, the use of optical flow for preprocessing has not been explored. Optical flow could provide valuable insights into the motion patterns of objects, further improving the model's detection and tracking accuracy.
3) **Dataset Utilization:** The model primarily relies on fully supervised learning methods, which may not fully leverage the available data. Future work could explore semi-supervised learning approaches to make better use of unlabeled data, potentially enhancing the model's generalization to diverse scenarios.
4) **Vulnerability to Adversarial Attacks:** The robustness of our model against adversarial attacks has not been thoroughly assessed. Incorporating adversarial training could strengthen the model's resilience to such attacks, ensuring its reliability in critical applications.
5) **Resource Efficiency for Edge Computing:** While optimizations have been made, the model's suitability for real-time, resource-constrained environments such as edge computing devices remains a challenge. Further optimization is necessary to ensure the model can operate efficiently in such settings without compromising performance.
6) **Domain Adaptation and Real-time Feedback:** The model's ability to adapt to different domains and utilize real-time feedback for continuous improvement has not been explored. These aspects are crucial for applications in dynamically changing environments and represent significant areas for future development.
7) **Hardware Acceleration and Custom Solutions:** The potential for custom hardware acceleration to enhance processing speed and efficiency has not been investigated. Future studies could explore the integration of custom hardware solutions to facilitate faster, more efficient object detection and tracking.
8) **Diverse Setting Evaluations:** Lastly, the model's evaluation across a wider range of real-world settings and conditions has been limited. Expanding the evaluation to more diverse environments would provide a clearer understanding of the model's capabilities and limitations.

Addressing these limitations represents a comprehensive roadmap for future research, aiming to enhance the capabilities of the YOLOv8 model for moving object detection. Through exploring hybrid neural architectures, advanced preprocessing techniques, and optimizations for real-time and resource-constrained applications, we anticipate significant advancements in the field of object detection.

V. CONCLUSION AND FUTURE WORKS

This paper undertook the challenge of enhancing the conventional object detection approach to specifically identify dynamic elements in visual data streams. Our deep dive into the YOLOv8 foundation reinforced its intrinsic strengths and capabilities. Yet, like many broad-spectrum solutions, opportunities for refinement and specialization were evident. Through the integration of customized preprocessing methods and critical architectural tweaks, the refined YOLOv8 model we presented exhibits an amplified acuity to motion nuances, leading to a more sophisticated recognition of objects in motion. Our comparative evaluations highlighted that, while retaining the core attributes of speed and precision synonymous with YOLO, our model demonstrated superior performance, particularly in environments teeming with motion. The broader ramifications of our findings span several domains. Be it for security surveillance, managing vehicular traffic, or dissecting motion in cinematic sequences, our augmented model sets a foundation for enhanced real-time decision-making tools and deeper analytical perspectives in domains where motion interpretation is crucial. However, as is characteristic of the expansive field of technology, the frontier for further innovation is limitless. Subsequent research endeavors could explore integrated model architectures, assimilate cutting-edge motion forecasting techniques, or fine-tune the model for distinct application areas. In its current state, our advanced YOLOv8 serves as a beacon, illuminating the immense possibilities that lie in tailoring deep learning tools for specialized object detection scenarios.

This paper showcases significant advancements in moving object detection but reveals several areas for improvement. These include its limited adaptability to complex motion patterns, reliance on static preprocessing techniques, constraints of supervised learning, vulnerability to adversarial attacks, and its resource-intensive nature, particularly in edge computing scenarios. Future work aims to address these limitations by exploring hybrid neural architectures with temporal memory components like LSTMs for nuanced motion prediction, implementing advanced preprocessing techniques such as optical flow, leveraging semi-supervised learning to enhance dataset utilization, ensuring model robustness through adversarial training, and optimizing the model for resource-constrained environments. Additionally, focusing on domain adaptation, real-time feedback loops, custom hardware acceleration, and comprehensive evaluations across diverse settings is poised to significantly refine the model's generalization capabilities and efficiency in real-world applications, making it a more adaptable and robust solution for moving object detection.

Thus, in future studies, building upon the proposed enhanced YOLOv8 model, there lies potential in exploring

hybrid neural architectures, incorporating temporal memory components like LSTMs for better motion prediction, and delving into advanced preprocessing techniques such as optical flow. Emphasis could also be placed on semi-supervised learning for better dataset utilization, ensuring robustness through adversarial training, and optimizing the model for edge computing to cater to real-time, resource-constrained scenarios. Additionally, domain adaptation, real-time feedback loops, custom hardware acceleration, and evaluations across diverse settings represent promising avenues to further refine moving object detection capabilities.

ACKNOWLEDGMENT

The authors express their gratitude to the University of Sfax, Tunisia, for administrative and technical support.

REFERENCES

[1] M. Safaldin, N. Zaghden, and M. Mejdoub, "Moving object detection based on enhanced YOLO-V2 model," in *Proc. 5th Int. Congr. Human-Computer Interact., Optim. Robotic Appl. (HORA)*, Jun. 2023, pp. 1–8.

[2] S. Ammar, T. Bouwmans, N. Zaghden, and M. Neji, "Deep detector classifier (DeepDC) for moving objects segmentation and classification in video surveillance," *IET Image Process.*, vol. 14, no. 8, pp. 1490–1501, Jun. 2020.

[3] S. Ammar, T. Bouwmans, N. Zaghden, and N. Mahmoud, "From moving objects detection to classification and recognition: A review for smart environments," in *Proc. Towards Smart World*, 2020, pp. 289–316.

[4] E. M. Ibrahim, M. Mejdoub, and N. Zaghden, "Semantic analysis of moving objects in video sequences," in *Proc. Int. Conf. Emerg. Technol. Intell. Syst.* Bahrain: Springer, 2022, pp. 257–269.

[5] F. Ben Aissa, M. Hamdi, M. Zaied, and M. Mejdoub, "An overview of GAN-DeepFakes detection: Proposal, improvement, and evaluation," *Multimedia Tools Appl.*, vol. 83, no. 11, pp. 32343–32365, Sep. 2023.

[6] H. Ma, T. Celik, and H. Li, "Fer-YOLO: Detection and classification based on facial expressions," in *Proc. Image Graphics: 11th Int. Conf.* Haikou, China: Springer, 2021, pp. 28–39.

[7] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 9961–9980, Aug. 2022.

[8] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image Vis. Comput.*, vol. 97, May 2020, Art. no. 103910.

[9] M. Hussain, H. Al-Aqrabi, M. Munawar, R. Hill, and S. Parkinson, "Exudate regeneration for automated exudate detection in retinal fundus images," *IEEE Access*, vol. 11, pp. 83934–83945, 2022.

[10] M. Hussain, M. Dhimish, V. Holmes, and P. Mather, "Deployment of AI-based RBF network for photovoltaics fault detection procedure," *AIMS Electron. Electr. Eng.*, vol. 4, no. 1, pp. 1–18, 2020.

[11] S. A. Singh and K. A. Desai, "Automated surface defect detection framework using machine vision and convolutional neural networks," *J. Intell. Manuf.*, vol. 34, no. 4, pp. 1995–2011, Apr. 2023.

[12] D. Weichert, P. Link, A. Stoll, S. Rüping, S. Ihlenfeldt, and S. Wrobel, "A review of machine learning for the optimization of production processes," *Int. J. Adv. Manuf. Technol.*, vol. 104, nos. 5–8, pp. 1889–1902, Oct. 2019.

[13] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *J. Manuf. Syst.*, vol. 48, pp. 144–156, Jul. 2018.

[14] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Ann.*, vol. 65, no. 1, pp. 417–420, 2016.

[15] S. Kulik and A. Shtanko, "Experiments with neural net object detection system YOLO on small training datasets for intelligent robotics," in *Proc. Adv. Technol. Robot. Intell. Syst. ITR*. Moscow, Russia: Springer, 2020, pp. 157–162.

[16] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, and S. Tang, "Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges," *Materials*, vol. 13, no. 24, p. 5755, Dec. 2020.

[17] P. Soviany and R. T. Ionescu, "Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction," in *Proc. 20th Int. Symp. Symbolic Numeric Algorithms for Scientific Comput. (SYNASC)*, Sep. 2018, pp. 209–214.

[18] L. Du, R. Zhang, and X. Wang, "Overview of two-stage object detection algorithms," *J. Phys., Conf. Ser.*, vol. 1544, no. 1, May 2020, Art. no. 012033.

[19] F. Sultana, A. Sufian, and P. Dutta, "A review of object detection models based on convolutional neural network," in *Intelligent Computing: Image Processing Based Applications*. Kolkata, India: Springer, 2020, pp. 1–16.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf.* Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 21–37.

[21] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06659*.

[22] W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang, "YOLO-face: A real-time face detector," *Vis. Comput.*, vol. 37, no. 4, pp. 805–813, Apr. 2021.

[23] M. Hussain, "YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, p. 677, Jun. 2023.

[24] W.-Y. Hsu and W.-Y. Lin, "Adaptive fusion of multi-scale YOLO for pedestrian detection," *IEEE Access*, vol. 9, pp. 110063–110073, 2021.

[25] N. M. A. A. Dazlee, S. A. Khalil, S. Abdul-Rahman, and S. Mutalib, "Object detection for autonomous vehicles with sensor-based technology using YOLO," *Int. J. Intell. Syst. Appl. Eng.*, vol. 10, no. 1, pp. 129–134, Mar. 2022.

[26] S. Liang, H. Wu, L. Zhen, Q. Hua, S. Garg, G. Kaddoum, M. M. Hassan, and K. Yu, "Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25345–25360, Dec. 2022.

[27] S. Shinde, A. Kothari, and V. Gupta, "YOLO based human action recognition and localization," *Proc. Comput. Sci.*, vol. 133, pp. 831–838, 2018.

[28] A. Hanan Ashraf, M. Imran, A. M. Qahtani, A. Alsufyani, O. Almutiry, A. Mahmood, M. Attique, and M. Habib, "Weapons detection for security and video surveillance using CNN and YOLO-V5s," *Comput., Mater. Continua*, vol. 70, no. 2, pp. 2761–2775, 2022.

[29] Y. Zheng and H. Zhang, "Video analysis in sports by lightweight object detection network under the background of sports industry development," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–10, Aug. 2022.

[30] D. Wu, S. Lv, M. Jiang, and H. Song, "Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105742.

[31] M. Lippi, N. Bonucci, R. F. Carpio, M. Contarini, S. Speranza, and A. Gasparri, "A YOLO-based pest detection system for precision agriculture," in *Proc. 29th Medit. Conf. Control Autom. (MED)*, Jun. 2021, pp. 342–347.

[32] Y. Nie, P. Sommella, M. O'Nils, C. Liguori, and J. Lundgren, "Automatic detection of melanoma with YOLO deep convolutional neural networks," in *Proc. E-Health Bioeng. Conf. (EHB)*, Nov. 2019, pp. 1–4.

[33] H. M. Ünver and E. Ayan, "Skin lesion segmentation in dermoscopic images with combination of YOLO and GrabCut algorithm," *Diagnostics*, vol. 9, no. 3, p. 72, Jul. 2019.

[34] L. Tan, T. Huangfu, L. Wu, and W. Chen, "Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification," *BMC Med. Informat. Decis. Making*, vol. 21, no. 1, pp. 1–11, Dec. 2021.

[35] L. Cheng, J. Li, P. Duan, and M. Wang, "A small attentional YOLO model for landslide detection from satellite remote sensing images," *Landslides*, vol. 18, no. 8, pp. 2751–2765, Aug. 2021.

[36] P. Kumar, S. Narasimha Swamy, P. Kumar, G. Purohit, and K. S. Raju, "Real-time, YOLO-based intelligent surveillance and monitoring system using Jetson TX2," in *Proc. Data Anal. Manag. ICDAM*. Singapore: Springer, 2021, pp. 461–471.

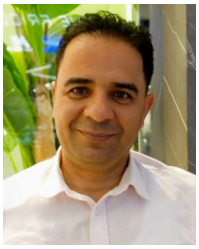
[37] K. Bhambani, T. Jain, and K. A. Sultanpure, "Real-time face mask and social distancing violation detection system using YOLO," in *Proc. IEEE Bengaluru Humanitarian Technol. Conf. (B-HTC)*, Oct. 2020, pp. 1–6.

[38] Y. Du, N. Pan, Z. Xu, F. Deng, Y. Shen, and H. Kang, "Pavement distress detection and classification based on YOLO network," *Int. J. Pavement Eng.*, vol. 22, no. 13, pp. 1659–1672, Nov. 2021.

[39] Hendry and R.-C. Chen, "Automatic license plate recognition via sliding-window darknet-YOLO deep learning," *Image Vis. Comput.*, vol. 87, pp. 47–56, Jul. 2019.

[40] C. Dewi, R.-C. Chen, and H. Yu, "Weight analysis for various prohibitory sign detection and recognition using deep learning," *Multimedia Tools Appl.*, vol. 79, nos. 43–44, pp. 32897–32915, Nov. 2020.
[41] A. M. Roy, J. Bhaduri, T. Kumar, and K. Raj, "WilDect-YOLO: An efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection," *Ecological Informat.*, vol. 75, Jul. 2023, Art. no. 101919.
[42] O. Sahin and S. Ozer, "YOLODrone: Improved YOLO architecture for object detection in drone images," in *Proc. 44th Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2021, pp. 361–365.
[43] X. Chen, X. Peng, R. Duan, and J. Li, "Deep kernel learning method for SAR image target recognition," *Rev. Sci. Instrum.*, vol. 88, no. 10, pp. 179–192, Oct. 2017.
[44] A. Körez, N. Barışçı, A. Çetin, and U. Ergün, "Weighted ensemble object detection with optimized coefficients for remote sensing images," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 6, p. 370, Jun. 2020.
[45] L. Khelifi and M. Mignotte, "Deep learning for change detection in remote sensing images: Comprehensive review and meta-analysis," *IEEE Access*, vol. 8, pp. 126385–126400, 2020.
[46] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS J. Photogramm. Remote Sens.*, vol. 159, pp. 296–307, Jan. 2020.
[47] J. Sublime and E. Kalinicheva, "Automatic post-disaster damage mapping using deep-learning techniques for change detection: Case study of the Tohoku tsunami," *Remote Sens.*, vol. 11, no. 9, p. 1123, May 2019.
[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[49] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2014, pp. 580–587.
[50] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[51] F. Gao, L. Fu, X. Zhang, Y. Majeed, R. Li, M. Karkee, and Q. Zhang, "Multi-class fruit-on-plant detection for apple in SNAP system using faster R-CNN," *Comput. Electron. Agricult.*, vol. 176, Sep. 2020, Art. no. 105634.
[52] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[53] S. Yang, L. Gu, X. Li, T. Jiang, and R. Ren, "Crop classification method based on optimal feature selection and hybrid CNN-RF networks for multi-temporal remote sensing imagery," *Remote Sens.*, vol. 12, no. 19, p. 3119, Sep. 2020.
[54] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Feb. 2020.
[55] B. Kellenberger, D. Marcos, and D. Tuia, "Detecting mammals in UAV images: Best practices to address a substantially imbalanced dataset with deep learning," *Remote Sens. Environ.*, vol. 216, pp. 139–153, Oct. 2018.
[56] J. Gou, L. Sun, B. Yu, S. Wan, and D. Tao, "Hierarchical multi-attention transfer for knowledge distillation," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 20, no. 2, pp. 1–20, Feb. 2024.
[57] D. Tuia, E. Pasolli, and W. J. Emery, "Using active learning to adapt remote sensing image classifiers," *Remote Sens. Environ.*, vol. 115, no. 9, pp. 2232–2242, Sep. 2011.
[58] H. Lei, K. Huang, Z. Jiao, Y. Tang, Z. Zhong, and Y. Cai, "Bayberry segmentation in a complex environment based on a multi-module convolutional neural network," *Appl. Soft Comput.*, vol. 119, Apr. 2022, Art. no. 108556.
[59] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[60] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[61] A. Kuznetsova, T. Maleva, and V. Soloviev, "Using YOLOv3 algorithm with Pre- and post-processing for apple detection in fruit-harvesting robot," *Agronomy*, vol. 10, no. 7, p. 1016, Jul. 2020.
[62] Y. Xia, S. Qu, and S. Wan, "Scene guided colorization using neural networks," *Neural Comput. Appl.*, vol. 34, no. 13, pp. 11083–11096, Jul. 2022.
[63] M. Majumder and C. Wilmot, "Automated vehicle counting from pre-recorded video using you only look once (YOLO) object detection model," *J. Imag.*, vol. 9, no. 7, p. 131, Jun. 2023.
[64] Y. Zhou, "A YOLO-NL object detector for real-time detection," *Exp. Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 122256.
[65] V. Kshirsagar, R. H. Bhalerao, and M. Chaturvedi, "Modified YOLO module for efficient object tracking in a video," *IEEE Latin Amer. Trans.*, vol. 21, no. 3, pp. 389–398, Mar. 2023.
[66] R. A. Murugan and B. Sathyabama, "Object detection for night surveillance using ssan dataset based modified YOLO algorithm in wireless communication," *Wireless Pers. Commun.*, vol. 128, no. 3, pp. 1813–1826, Feb. 2023.
[67] C. Cuevas, E. M. Yáñez, and N. García, "Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA," *Comput. Vis. Image Understand.*, vol. 152, pp. 103–117, Nov. 2016.
[68] I. Delibasoglu, "PESMOD: Small moving object detection benchmark dataset for moving cameras," in *Proc. 7th Int. Conf. Frontiers Signal Process. (ICFSP)*, Sep. 2022, pp. 23–29.
[69] A. Xuehui, Z. Li, L. Zuguang, W. Chengzhi, L. Pengfei, and L. Zhiwei, "Dataset and benchmark for detecting moving objects in construction sites," *Autom. Construction*, vol. 122, Feb. 2021, Art. no. 103482.

MUKARAM SAFALDIN received the bachelor's degree in computer science from Diyala University, Iraq, in 2013, and the master's degree in computer science from Amman Arab University, Jordan, in 2020. He is currently pursuing the Ph.D. degree in computer science with a focus on artificial intelligence, with the University of Sfax, Tunisia.



NIZAR ZAGHDEN received the master's degree in novel technologies in dedicated computer systems and the Ph.D. degree in computer system engineering from the National Engineering School of Sfax, Tunisia, in 2005 and 2013, respectively. His Ph.D. thesis is entitled characterization of the content of ancient document images. Concerning his professional career, he was recruited, in 2009, as an Assistant Professor with the Department of Computer Science, Superior Institute of Informatics in Medenine, Tunisia. In 2013, he was promoted to the rank of Assistant Professor. In 2014, he was moved to the Higher School of Business of Sfax, Tunisia, as an Assistant Professor. He is currently working on intelligent applications for smart cities dealing with the classification of images and video from CCTV cameras.

MAHMOUD MEJDOUB received the engineering degree in computer engineering and the master's degree in novel technologies in dedicated computer systems from the National Engineering School of Sfax, Tunisia, in 2004 and 2005, respectively, the joint Ph.D. degree in computer system engineering from the National Engineering School of Sfax and in automatic, signal and image processing from the University of Nice Sophia Antipolis, France, in 2011, and the Habilitation (accreditation to supervise research) degree in computer system engineering from the National Engineering School of Sfax, in 2017. Regarding his professional career, he was recruited, in 2011, as an Assistant Professor with the Department of Computer Science and Communications, Faculty of Sciences of Sfax, Tunisia. In 2018, he was promoted to the rank of Associate Professor with the Department of Computer Science and Communications, Faculty of Sciences of Sfax. He is currently a Research Member with the Research Unit Sciences and Technologies of Image and Telecommunications. His research interests include computer vision, image processing, artificial intelligence, and deep learning.