Journal of Theoretical and Applied Information Technology
31st March 2025. Vol.103. No.6

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A HYBRID FRAMEWORK FOR OBJECT DETECTION AND SEGMENTATION IN AUTONOMOUS VEHICLES USING YOLO NAS AND MASK R-CNN

DARTHY RABECKA V[1*],BRITTO PARI J [2]

[1,]Research Scholar,veltech Rangarajan Dr.sagunthala R and D Institute of science and Technology, ECE, chennai, India.

[2]Associate Professor veltech Rangarajan Dr.sagunthala R and D Institute of science and Technology, ECE, chennai, India.

E-mail: [1]vtd1112@veltech.edu.in, [2]brittopari@veltech.edu.in

ABSTRACT

A leading opinion prescription that can precisely establish and segregate objects in complex environments is necessary for the hurried development of self-driving autos. The new hybrid framework shown in this work improves object detection and segmentation performance by combining Mask R-CNN with You Only Look Once Neural Architecture Search (YOLO NAS). With the neck and head of YOLO-NAS retained, this study tries to boost the performance of YOLO-NAS by substituting a combination of Res Net and Feature Pyramid Network (FPN) for the default Rep Ne X t backbone. Additionally, to increase segmentation capabilities, the study integrates Mask R-CNN. Our methodology leverages the efficiency of YOLO NAS for fast object detection and the precision of Mask R-CNN for complex segmentation tasks using the KITTI dataset, a leading benchmark in autonomous driving research. This approach resolves issues such as disparate object sizes, obstructions, and complex backgrounds that are frequently encountered when driving in urban areas. Our cloud-based approach outperforms previous approaches in terms of precision, recall, and F1 scores. The results of our experiments show that this combination approach could greatly contribute to the development of more reliable and safer autonomous driving systems, paving the way for advancements in real-time perception technology.

Keywords: *Convolution Neural Network, Object Detection, Autonomous Vehicles, You Only Look Once*

1. INTRODUCTION

Accurately detecting and counting vehicles is crucial for improving traffic management since the increasing number of vehicles has made traffic control more difficult. In applications such as surveillance and autonomous driving, the recognition of objects and segmentation are crucial. Real-time detection can be accomplished with YOLO-based models, but accurate segmentation is problematic. Though Mask R-CNN promises precise segmentation, it is computationally intensive. The technique [1] analyzes road damage in real time by combining deep learning models with UAV data. Cracks, potholes, and other structural flaws can be correctly determined considering to the greater feature extraction provided by high-resolution aerial photographs. Nevertheless, the approach is constrained by weather and UAV flying constraints, which can possess an influence on data consistency.YOLOv4 pairs with a CNN-based architecture [2] to increase the accuracy of traffic sign detection. In complicated and shifting road situations, the model assures continuous detection by increasing feature extraction. But the model has difficulty understanding small and occluded signs, which result in wrong categorization during times with a lot of traffic. With a focus on object detection, lane recognition, and obstacle identification, this work [3] evaluates new methods for deep learning used in road analysis. Autonomous vehicle perception and decision-making are enhanced by complex CNN and transformer-based models. Still, real-time processing is difficult for systems with little power due to the high computing demands of transformer-based models. To enhance the interpretability of autonomous driving decisions, the study [4] utilizes the use of the Inception U-Net model with Grad-CAM visualization. Understanding scenes in unstructured traffic situations—especially on Indian roads—is its main focus. By raising model

openness, this strategy helps self-driving cars make safer choices. The fact that Grad-CAM visualization can sometimes not give clear explanations for intricate decision boundaries is a disadvantage that could cause confusion about model interpretations. The investigation [5] proposes a deep feature ensemble method for better vehicle detection that combines many feature extraction methods. The method improves accuracy in complex traffic jams with occlusions and changing lighting. The model is tested on a variety of datasets, proving that it survives in everyday situations. However, the efficacy of real-time detection have affected by the increased computational overhead incurred with combining various feature extraction algorithm design.

GS-YOLO Net [6], a lightweight object detection model intended for highway distance estimation and real-time tracking, is shown in this work. The approach promotes the speed of processing, which makes it appropriate for embedded autonomous driving systems. The model maintains a substantial amount of inference while delivering increased accuracy. Its performance, however, declines in extremely dynamic settings with lots of traffic and various product sizes. The suggested WGS-YOLO model [7] optimizes feature extraction and network efficiency to improve real-time detection in applications including autonomous driving. It optimizes robustness and small object the identification in a range of light exposure scenarios. The model performs more rapidly and accurately than previous YOLO versions. However, for extremely complicated situations, its real-time processing optimization leads to a decline in detection performance. By employing deep learning for adaptive real-time identification, the pavement damage detection system [8] improves road safety and maintenance scheduling. On the other hand, detection consistency has impacted by changes in pavement textures and illumination. Although the hybrid CNN-Transformer model [9] ensures accuracy and computational efficiency in complex traffic scenarios through enhancing the recognition of small and veiled objects, its increased computational burden restrict its use on edge devices. While YOLO-based skip connection model [10] improves large-scale defect monitoring through the use of UAV images to identify noise barrier flaws in high-speed rail environments, its usefulness in no-fly zones is constrained due to its reliance on UAV data.

The two-stage fusion network [11] combines transformer and CNN-based feature extraction to raise the accuracy of traffic sign detection; still its complexity increases the inference time on hardware with limited space. The dedicated neural network [12] optimizes ship tracking from radar blips by mimicking human vision and capturing domain-specific features, but the model is unable to handle unknown aquatic circumstances and noise swings. Lightweight face and landmark tracking are used in the joint localization a position [13] to detect driver drowsiness, ensuring real-time performance; however, detection reliability is decreased by face occlusions and insufficient illumination. For autonomous cars, the perception module design [14] combines stereo depth estimation and semantic segmentation to enhance obstacle recognition; nonetheless, real-time stereo processing needs substantial computational resources. Finally, the pavement distress system based on street view [15] uses deep learning-based feature extraction to automate large-scale road condition monitoring; nonetheless, misdetection exists due to vehicle occlusions and noise from the surroundings. In this field, research is currently ongoing.

1.1 RESEARCH MOTIVATION

Nowadays hybrid recognition of objects and segmentation models has problems economically striking equilibrium between accuracy and speed. YOLO-NAS, a cutting-edge real-time detector, lacks hierarchical feature extraction by default and encourages speed over precision when discovering small or hidden objects. Improving YOLO-NAS employing Res Net and Feature Pyramid Network (FPN) is essential for maintaining immediate performance while increasing detection and segmentation accuracy.

In spite of Rep Ne X t ability to gather multi-scale features, existing YOLO-based frameworks frequently sacrifice segmentation accuracy for speed. Research on putting Mask R-CNN into YOLO-NAS while improving its backbone using Res Net and FPN is scarce. The objective of the study is to close this gap through enhanced object recognition and segmentation, especially for applications utilizing autonomous driving.

This study suggests substituting the Rep Ne X t backbone in YOLO-NAS with Res Net and FPN while incorporating Mask R-CNN with the goal to enhance actual time object perception. With this modification, hierarchical feature extraction becomes better and more accurate segmentation is made possible, including for small or occluded objects in difficult driving conditions.

With the incorporation of Mask R-CNN and Res Net and FPN as the new YOLO-NAS backbone, the current research aims to enhance object detection and segmentation performance while sustaining real-time efficiency. For applications involving autonomous driving, the new model is expected to provide greater precision in acknowledging small and concealed objects.

An innovative combination technique is offered that improves feature extraction and segmentation by merging YOLO-NAS with Res Net, FPN, and Mask R-CNN. This method retains the quick inference skills required for autonomous perception in real time while ensuring improved object recognition.

Real-time recognition of objects and segmentation for autonomous cars is the main emphasis of the study, which validates performance using the KITTI dataset. For practical use in self-driving systems, the suggested approach seeks to improve detection precision and computing efficiency. Despite increasing segmentation, the combination of Mask R-CNN with hierarchical feature learning raises computational complexity. However, real-time processing capabilities remain intact through the use of suitable hardware and improved execution tactics, making the model viable for real-world applications. This work is predicated on the assumption that the KITTI dataset offers enough variation to confirm the effectiveness of the model and that the improvement in feature extraction and segmentation yields efficiency improves that are more than the computational cost.

The paper's main contributions include:

- Enhancing YOLO-NAS by replacing Res Net and FPN for its RepNeXt backbone for more effectively hierarchical feature extraction;
- Promoting Mask R-CNN into YOLO-NAS to improve segmentation accuracy without harming real-time performance;
- Sustaining the YOLO-NAS neck and head to preserve efficient information flow for high-performance object detection;
- Monitoring the proposed structure on the KITTI dataset, indicating improved detection of small and occluded objects; and
- Establishing balance between the segmentation accuracy and inference speed, making it suitable for intelligent surveillance and autonomous driving use.

The research begins by looking at the development of many YOLO-NAS and Mask R-CNN frameworks for vehicle recognition and segmentation. The Related Works (Section II) provide a detailed review of previous research on YOLO-NAS in combination with Mask R-CNN, focusing on their designs and outcomes. In Section III, the architectural design and workflow are discussed, along with the Mask R-CNN implementation of YOLO-NAS. Section IV displays the results of the experiments, including model training, visual outputs, and in-depth analysis. It also examines the used dataset and evaluates the model's performance. Section V, which concludes with the references, summarizes the main findings and offers potential directions for additional research. Given the increasing demand for reliable vehicle identification systems, a comparison between YOLO-NAS and Mask R-CNN is crucial. This study aims to investigate the benefits, drawbacks, and unique characteristics of these YOLO-NAS and Mask R-CNN variants in vehicle recognition and segmentation. By evaluating their architectures, performance, and real-world applications, this study seeks to provide researchers and practitioners in the fields of computer vision and object recognition with valuable insights.

2. RELATED WORK

Recent studies on YOLO's evolution have included a number of advances and unique methodologies in the field of object recognition, showing considerable improvements in performance and efficiency. On the MS COCO test-dev 2017, YOLOv8 [16] demonstrated significant performance gains, surpassing YOLOv5's 50.7% Average Precision (AP) with an incredible AP of 53.9%. Images with 640 pixels were utilized in both models. YOLO-NAS maximizes the accuracy-latency trade-off by applying quantization only to certain areas of the model. Although this approach improves inference speed and accuracy, it has disadvantages as well, such as increased deployment complexity and lengthier training times due to the usage of large datasets and self-distillation techniques. Application in Traffic Management [17]: YOLO is crucial for accurately detecting and following moving vehicles, monitoring traffic flow, and identifying traffic signs. Because of its accuracy and efficiency, it performs better than many other object identification methods, making it a strong candidate for tasks involving traffic. However,

despite its real-time accuracy, it has significant limitations, such as the difficulty to identify small or overlapping objects in cluttered or complex environments. With a focus on reducing detection mistakes through hyper parameter optimization and data augmentation, a novel method [18] that combines YOLO and Dark net for vehicle localization and identification was presented, showcasing YOLO's applicability in vehicle-related scenarios. This technique have negative aspects, such as the difficulty of modifying hyper parameters and the computing overhead required for effective data augmentation, even though it increases accuracy and robustness.

YOLOv8 for Feature Recognition with Custom Training [19]: Bus stop features were successfully detected by a tailored YOLOv8 model. The efficiency of the model for this task was demonstrated in an initial test utilizing the Bus Stop CV tool. This approach does, however, have several drawbacks. It is difficult to use outside of the particular features it was trained on and necessitates a large amount of training data. YOLOv8 fared better than previous YOLO versions in Emergency Vehicle Detection [20] in terms of recall, precision, and mean average precision (mAP). The study found that YOLOv8 is the most accurate and effective model for handling object detection and detecting emergency vehicles. Despite its advantages, which include increased detection accuracy and reliability, it has drawbacks, such as higher computational requirements and possible challenges with real-time processing in dynamic environments. In comparison an enhanced YOLOv8 network model [21] shows 14.6% in frames per second (FPS) and 1.4% in mean Average Precision (mAP). The experimental results showed a considerable boost in both processing speed and accuracy. Because of its complexity, implementing the improved model requires more processing power, even though these enhancements result in superior performance for real-time applications. Experiments revealed that YOLOv5, YOLOv7, and YOLOv8 [22] had a better balance across all performance parameters in both the validation and testing phases. Among these models, YOLO-NAS variants had the highest memory ratings, making them stand out in critical applications where high recall rates are necessary. Notwithstanding their advantages, the growing complexity of these designs results in higher resource consumption and longer training times.

It was suggested that YOLOv8 and Open CV [23] be combined to improve the effectiveness and dependability of traffic monitoring systems. This strategy seeks to increase general road safety and reduce traffic-related accidents. Although managing complicated traffic scenarios and varied illumination conditions presents hurdles, this combination offers notable gains in detection accuracy and response times. Promising improvements in vehicle detection accuracy were demonstrated by a new algorithm that uses YOLOv8 [24] to recognize different automobiles in CCTV footage in real-time. This technique works well in dynamic settings where prompt detection is essential. The computing load necessary for real-time processing and the requirement for large amounts of training data, however, pose difficulties. Depending on vehicle density, the YOLO algorithm [25] has demonstrated increased accuracy in identifying several vehicles within photos, with detection rates ranging from 80% to 85%. With this degree of precision, YOLO is positioned as a dependable option for a range of vehicle detection applications. However, problems occur when there is a lot of traffic since overlapping items makes detection less effective. Improved recognition accuracy was achieved by applying an optimization technique to the YOLO framework [26]. Compared to conventional YOLO implementations, experiments on images with varying resolutions showed superior accuracy. While performance is enhanced by this method, the model becomes more complex and more computational resources are needed for optimization. In comparison to other classification strategies, the YOLO-CFNN and YOLO-VCFNN vehicle [27] classification methods demonstrated an amazing accuracy rate of 99%. These techniques are very useful for a variety of vehicle classification jobs because of their high accuracy. Nevertheless, the intricacy of these models necessitates substantial time and computational resources for both deployment and training. In order to improve detection accuracy, modern algorithms for human-AI improved object identification [28] in autonomous cars make use of deep learning, reinforcement learning, and sophisticated sensors like LiDAR and high-resolution cameras. By integrating human input, adjusting to changing conditions, and facilitating sound decision-making, these systems increase efficiency and safety. However, they face difficulties with human input dependability, integration complexity, and high resource needs, which affect user trust and real-time performance.

YOLO-NAS and YOLOv8 act as detectives cracking a case by rapidly recognizing and evaluating license plates in a variety of settings. Vehicle license plate identification [29] makes use

of methods like deep learning and computer vision. These algorithms greatly enhance detection performance by providing benefits including high plate identification accuracy and conditional adaptability. But they face obstacles like occlusion, different plate designs, and weather conditions that can make detection difficult and need a lot of processing power. An improved version of YOLOv8 [30] is covered in this work, with an emphasis on methods such sophisticated data augmentation, optimal anchor box selection, and integrating temporal information from successive frames to increase moving object recognition. The benefits of this upgraded YOLOv8 include decreased false positives, more efficient operation in difficult situations, and more accuracy in detecting and tracking swiftly moving objects. But there are drawbacks as well, such as increased processing demands, possible delay in real-time applications, and the requirement for a large amount of training data to reach its full potential.

YOLOv8-Lite [31], a simplified object detection model created especially for real-time applications in autonomous driving systems, is presented in this appearance. Model pruning, quantization, and optimized design are some of the methods used to lower computing complexity without sacrificing detection accuracy. YOLOv8-Lite's benefits include quicker inference times, lower memory consumption, and increased efficiency, which makes it appropriate for environments with limited resources. Its drawbacks, however, include difficulties precisely detecting small or obscured objects and a possible trade-off in detection precision when compared to more sophisticated models. For object detection and segmentation tasks, YOLO-NAS and Mask R-CNN integration shows a significant improvement in performance and efficiency. YOLO-NAS offers quick and accurate object detection, and Mask R-CNN helps with accurate instance segmentation. Better detection rates and processing speeds are the outcomes of this combination, which makes it especially useful for real-time applications. Innovative methods like data augmentation and hybrid quantization also improve this approach's resilience in a variety of settings. Overall, the results show how YOLO-NAS and Mask R-CNN may be used to significantly improve object identification skills, particularly in autonomous systems and complicated visual settings.

3 METHODOLOGY

The detailed explanation of the architecture and dataset used in this proposed model is given below

3.1 Modified YOLONAS

With its remarkable performance and creative design, YOLO-NAS in conjunction with Mask R-CNN represents a substantial breakthrough in object identification and instance segmentation. This model incorporates new methods and improvements to increase accuracy and efficiency. Figure 1 shows the YOLO-NAS architecture.

A state-of-the-art object detection model called YOLO-NAS (You Only Look Once – Neural Architecture Search) uses Neural Architecture Search (NAS) to improve its structure for effectiveness and performance. Its three primary parts—the head, neck, and backbone—are all made to interpret visual data effectively and provide precise real-time object detection. The framework originates with the Backbone, which captures hierarchical information by transforming the original image through a number of steps. The Stem Block initiates the process at the beginning, while stages 1 through 4 concentrate on capturing low-level to high-level elements. These traits are improved upon in each succeeding step, guaranteeing reliable representation across various image scales and levels of complexity.

The fundamental basis for recognizing items in an image is the backbone. To improve its feature extraction capabilities, the backbone incorporates the Spatial Pyramid Pooling (SPP) module. The model can handle objects of different sizes since SPP enables it to pool features across many spatial scales. The Quantized Spatial Pyramid (QSP) block, which adds quantization to the SPP, is a significant improvement in YOLO-NAS. The model's capacity to handle multi-scale characteristics is maintained while computing efficiency is maximized by this innovation, which makes it ideal for real-time applications and deployment on edge devices. The neck of YOLO-NAS connects the head and backbone, guaranteeing efficient aggregation of features extracted at various scales. Attention methods are used to highlight

Journal of Theoretical and Applied Information Technology
31st March 2025. Vol.103. No.6

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

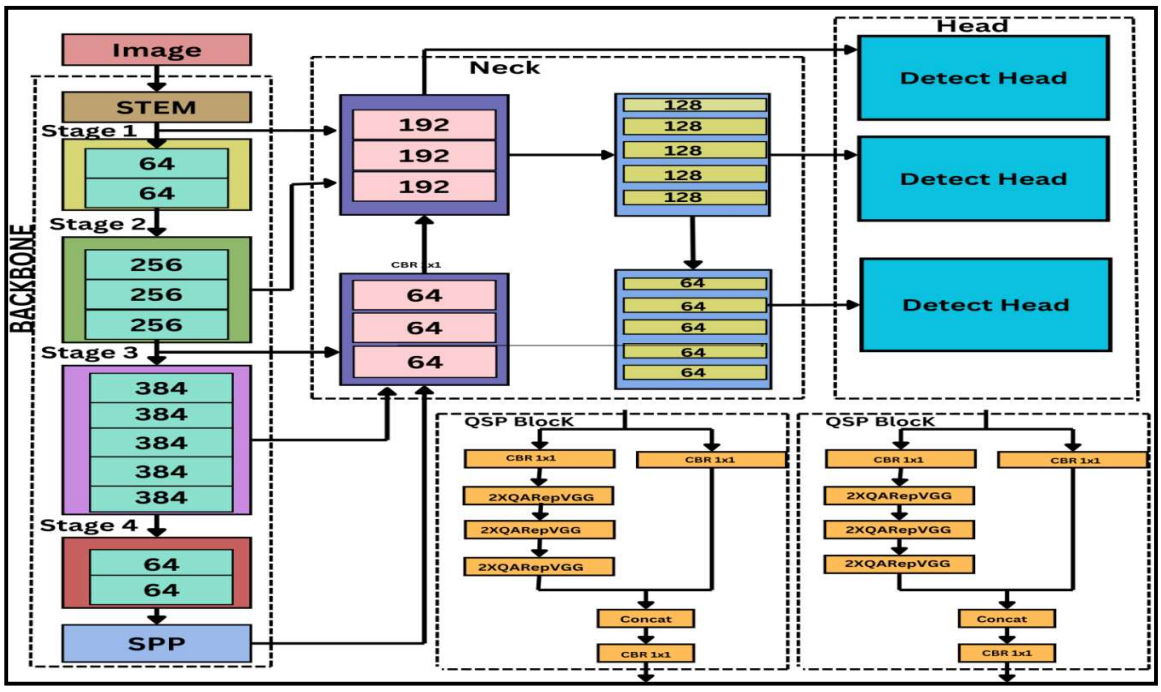

***Figure 1**: YOLO-NAS Architecture*

important area of the image in Stages 5 and 6, which incorporate multi-scale feature maps. Here, the QSP block is widely used to lower computing costs without sacrificing feature map richness. This guarantees that the neck produces a sophisticated feature set that strikes a balance between precision and effectiveness. The prediction and post-processing activities are handled in Stages 7 and 8 of the final Head. Convolutions are optimized during the prediction phase using the Quantized Convolution Integration (QCI) block, which allows for quicker inference and lower memory usage. This head locates and identifies items in the image by producing bounding boxes, class labels, and confidence ratings. YOLO-NAS is a great option for real-time detection in resource-constrained environments, such as autonomous driving and surveillance systems, because of the architecture's integration of QSP and QCI blocks, which guarantees excellent performance with low latency. With this approach, the neck and head of YOLO-NAS remain intact while the Rep Ne X t backbone is switched out for Res Net and Feature Pyramid Network (FPN). Through the application of top-down and lateral feature fusion, deep residual connections, and multi-scale learning, the Res Net-FPN combo improves feature extraction. To keep its feature refining capabilities, the neck of YOLO-NAS analyzes the multi-resolution feature maps that are produced by the modified backbone. YOLO-NAS's detection head, which manages object ness scoring, bounding box regression, and classification, stays the same. When FPN and Res Net merge, the multi-scale feature representation is improved, enabling the model to accurately and successfully understand objects of various sizes.

Algorithm 1: Rep Ne X t Backbone Replacement in YOLO-NAS Using Res Net and FPN

Step 1 Res Net-50 FPN Initialization: To enhance multi-scale feature maps, load ResNet-50 and integrate FPN.

$$P_i=\text{Conv}(F_i) + \text{Up sample}(P_{i+1})$$

Step 2 Multi-Scale Feature Extraction: Extract feature maps from multiple ResNet-50 layers and use FPN to enhance them.

Step 3 Rep Ne X t Replacement: Take out Rep Ne X t and feed the YOLO-NAS neck with FPN-enhanced features.

Step 4 Move Features to Detection Head: Generate predictions by analyzing YOLO-NAS's improved characteristics.

$$S_o=\sigma(W_oP),\hat{b}=W_bP \text{ and } C=\text{Soft max}(W_cP)$$

Step 5 Calculate Loss Function: Optimize bounding box regression, object ness, and grouping losses.

$$L=\lambda_1 L_{cls}+ \lambda_2 L_{obj}+ \lambda_3 L_{bbox}$$

Step 6: Stochastic Gradient Descent (SGD) Optimization: SGD is an approach for changing model weights.

$$W'=W-\eta\frac{\partial l}{\partial w}$$

Step 7: Infer and Employ NMS: Non-Maximum Suppression (NMS) is employed to enhance results, extract features, and make predictions.

In algorithm1, Rep Ne X t in YOLO-NAS is substituted by ResNet-50 with FPN for improved multi-scale feature extraction. Using lateral and top-down connections, the FPN enhances object detection at various scales by upgrading feature maps. After the updated features are transmitted to the YOLO-NAS detection head, class labels, bounding boxes, and object ness scores are generated. Weight updates are handled out via SGD, and training optimizes classification, objectness, and localization losses. To enhance accuracy and protect YOLO-NAS efficiency during inference, NMS serves to disable redundant detections.

3.2 YOLO-NAS with MASK RCNN

The design of YOLONAS with MASK R-CNN is shown in Fig 2. To provide effective object detection and segmentation, the integrated framework makes use of the advantages of modified YOLO NAS, ResNet-50, FPN (Feature Pyramid Network), and MASK R-CNN. The foundation for feature extraction is ResNet-50, which has 50 layers and a deep architecture. Its residual connections make it possible to efficiently

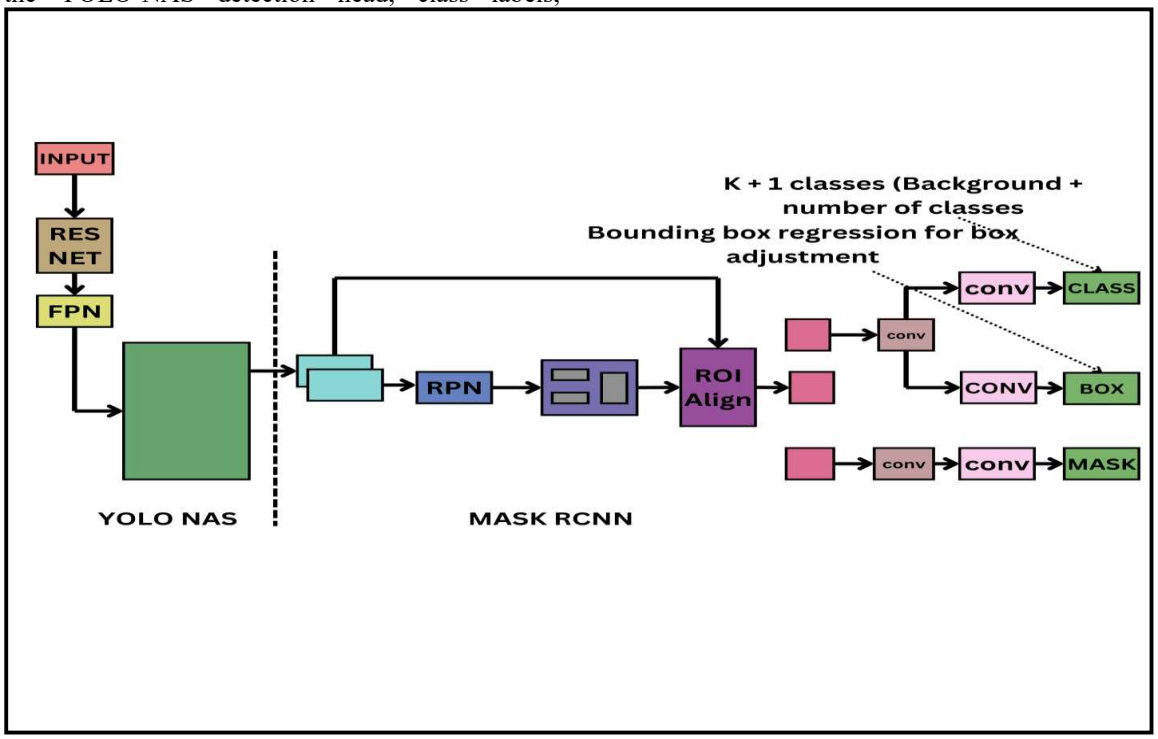

Figure 2: *Modified YOLO-NAS with MASK RCNN Architecture*

extract detailed information from pictures by preventing the vanishing gradient issue during training. ResNet-50 is very good at processing complicated visual data because it can extract both high-level and low-level features from the input image.

The FPN improves multi-scale feature representation after the features are retrieved. To provide accurate detection of objects of varied sizes, the FPN creates a pyramid of feature maps at various resolutions. Because object sizes can vary greatly in datasets like KITTI, it improves the model's ability to recognize both small and large

things by fine-tuning the spatial information. The YOLONAS and MASK R-CNN branches get the processed feature maps in this integrated architecture. The detection pipeline gains instance segmentation capabilities from MASK R-CNN. MASK R-CNN does more than just identify objects and create bounding boxes; it also adds a pixel-level mask to every object it finds. This enables accurate object segmentation, in which the shape of every object in the image is precisely recognized. MASK R-CNN generates object candidates using a region proposal network (RPN) and subsequently refines them for precise segmentation.

This framework generates extremely detailed outputs by fusing the deep feature extraction of ResNet-50, the multi-scale enhancements of FPN, the segmentation precision of YOLO NAS, and the segmentation power of MASK R-CNN. This architecture provides a reliable solution for both detection and segmentation tasks, producing class labels, bounding boxes, and masks for every identified object.

Algorithm 2: Mask R-CNN for Modified YOLO-NAS

Step 1 Backbone Replacement: For feature extraction, replace out Rep Ne X t with ResNet-50.

$$F=\text{RES NET 50(I)}$$

Step 2 Enhancement with the Feature Pyramid: FPN is utilized for enhancing multi-scale features.

$$P=\sum_{i=1}^{N} \alpha_i F_i$$

Step 3 identifying Head Processing: Estimate bounding boxes B and class scores C

$$B,C=W_d P$$

Step 4 Segmentation Mask Branch: Build segmentation masks M

$$M= \sigma(W_m P)\ (WmP)$$

Step5 Post-processing and Inference: Apply thresholding and Non-Maximum Suppression (NMS) for better outcomes.

In algorithm 2, the revamped YOLO-NAS integrates FPN for better feature extraction and swaps out Rep Ne X t for ResNet-50. To enhance spatial and contextual information before transferring it to the detecting head, the YOLO-NAS neck is maintained. The detection head successfully predicts bounding boxes and object classes, while instance segmentation is performed out by a parallel Mask R-CNN branch. Losses from segmentation, localization, and classification are all used to train the model. Non-maximum suppression and thresholding are used during inference to boost final detections and segmentation masks.

3.3 Dataset Used

The KITTI Panoptic Segmentation dataset is used in this project to enhance comprehension of urban scenes. With 1,055 carefully annotated photos for both the "stuff" and "thing" classes, the dataset is ideal for a wide range of segmentation applications. Of them, 200 are used for validation and 855 for training. With a resolution of 1280 x 384 pixels, each image is taken from an urban setting and features a wide range of surfaces and things that are commonly seen in cities. Eight 'thing' classes, including automobiles, pedestrians, and cyclists, and eleven 'stuff' classes, including roads, sidewalks, and vegetation, are annotated in the dataset. These annotations facilitate thorough panoptic segmentation, labeling both foreground ('things') and background ('stuff') items. For

Figure 3: Dataset Collection

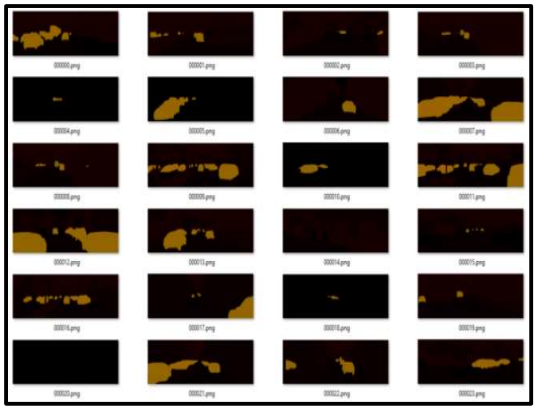

Figure 4: Data Annotation

research purposes, the dataset is available at http://panoptic.cs.uni-freiburg.de/ [46]. We used this dataset because it was accessible using current research methods.The Cityscapes collection, which includes photos taken in spring, summer, and fall in more than fifty European cities, is also used in addition to KITTI. There are 5,000 photos with good annotations and 20,000 photos with rough annotations in this dataset. 500 photos are used for validation, 1,525 for testing, and 2,975 for training. Pixel each pixel, each image is tagged for 19 object classes, consisting of 8 'thing' classes and 11'stuff' classes. Panoptic segmentation in Cityscapes is very difficult due to the variety of urban landscapes, dynamic objects, occlusions, and changing lighting conditions. With the use of current techniques [46], the suggested model is assessed using the validation and test sets, and performance results are uploaded to an online assessment server. For this research, a subset of 1,055 images has been manually labeled with panoptic ground truth, despite the fact that the original KITTI dataset does not provide panoptic segmentation annotations. Using the same distribution of' stuff' and 'thing' classes as in Cityscapes, 855 of these are used for training and 200 for validation. In order to guarantee exact borders between objects, instance masks were combined with semantic segmentation labels to produce panoptic annotations. Semantic segmentation maps are superimposed on the RGB pictures to produce precise item boundaries and masks, which serve as ground truth labels for panoptic segmentation. In Fig. 3, data collection is displayed.

Data annotation is the process of classifying and labeling different elements in the dataset, which is made up of 19 object classes classified into two primary groups: 8 'Thing' Classes and 11 'Stuff' Classes. Whereas the 'Thing' Classes relate to specific objects like automobiles, pedestrians, and cyclists, the 'Stuff' Classes depict amorphous regions like roads, skies, sidewalks, and plants. The dataset is made up of 1055 KITTI images, of which 200 are set aside for validation and 855 are set aside for training. Semantic information, such as instance masks, semantic segmentation pictures, and overlapping RGB data, is painstakingly added to each image. Applications in object detection and scene comprehension are made possible by these annotations, which offer a thorough grasp of the scenes. Fig. 4 provides an illustration of data annotation.

An essential step in getting the dataset ready for training is data preparation. To maintain consistency throughout the collection, this entails scaling photographs to standard dimensions, like 512x1024 pixels. Pixel values are adjusted for data normalization, resulting in mean values of 106.433, 116.617, and 119.559 for the Red, Green, and Blue channels, respectively. These mean values are then split by their standard deviations. By choosing random image portions during training, random cropping introduces variability and strengthens the model's resilience. Because of the high data quality and consistency guaranteed by these preprocessing methods, the dataset can be used to train deep learning models for a variety of computer vision tasks. Figure 5 shows data preprocessing.

The prepared dataset is sent into the YOLO-NAS and MASK R-CNN integrated framework after the data pretreatment processes. This partnership successfully improves segmentation and object detection skills. MASK R-CNN offers accurate instance segmentation, which enables the model to

Figure 5: Data Preprocessing

defines the borders of each identified object, while YOLO-NAS uses its sophisticated architecture to swiftly and precisely identify objects in images. The suggested framework exhibits notable gains in performance metrics by combining these two potent systems, attaining high accuracy in object detection and classification. The outcomes demonstrate the system's proficiency in managing intricate settings by precisely identifying and classifying items including cars, pedestrians, and infrastructure elements. Overall, this suggested approach is a useful tool for applications in autonomous driving, urban planning, and smart city development since it not only expedites the scene understanding process but also yields dependable and efficient results.

3. RESULT AND ANALYSIS

We combine the YOLO-NAS and Mask R-CNN models in this investigation. The KITTI dataset is used to refine both models in order to increase accuracy and performance. An Nvidia RTX3080 GPU was used for the training process, which improved computing efficiency and speed up model optimization. The power required to manage challenging training assignments and get the best results is supplied by this GPU. The YOLO-NAS model's capacity to acquire contextual information at different sizes is greatly improved by its ability to dynamically modify its receptive field in response to input data.

Effectively distinguishing between background elements and objects, especially in complicated settings, requires this functionality. Mask R-CNN, on the other hand, enhances this optimization by providing sophisticated detection and segmentation capabilities that allow for accurate object identification in images.

We want to enhance the YOLO-NAS model's performance in object detection and semantic segmentation tasks by improving it. Better feature extraction is made possible by these improvements, which enable the model to react to various contextual cues and specifics found in the input data. This flexibility guarantees that the model maintains a high degree of accuracy and flexibility while functioning well in a variety of settings. By combining the advantages of both models, YOLO-NAS and Mask R-CNN create a potent framework for picture analysis.

Effective object segmentation is made possible by Mask R-CNN's architecture, which also offers a thorough comprehension of object boundaries and their interactions within the scene. This combination improves the model's overall comprehension of complex scenarios and increases segmentation accuracy, providing useful insights for applications like augmented reality, robotics, and autonomous driving.

The enhanced YOLO-NAS model is also appropriate for situations that call for quick decisions based on visual input because it enables real-time processing capabilities. This improved framework demonstrates its ability to balance speed and accuracy, making it a strong instrument for furthering computer vision research and real-world applications. In Figure 6, the output's results are displayed.

When evaluating the effectiveness of object identification models, metrics like F1 Score, mAP (Mean Average Precision), FPS (Frames per Second), and Precision are essential. These measures assess processing speed, detection quality across all object classes, accuracy, and the trade-off between precision and recall.

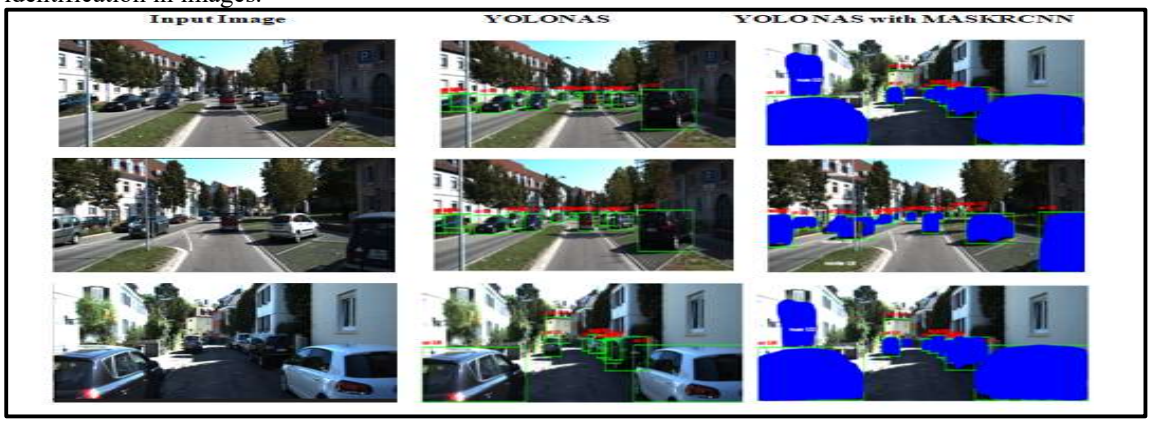

FIGURE 6: Yolo Nas With Mask Rcnn Output Results

We comprehend a model's overall efficacy and efficiency by examining these measures. The accuracy of the model's positive predictions is referred to as precision. It evaluates the proportion of predicted bounding boxes that match the real objects in object detection.

$$\text{Precision}=\frac{\text{True Positives}}{\text{True Positives}+\quad\text{Positives}} \quad (1)$$

Objects that were accurately anticipated are known as True Positives (TP). False Positives (FP): Predictions that are inaccurate or that don't include any objects. Fewer false positive predictions are indicated by a high precision number. Precision and Recall's harmonic mean is the F1 Score. It offers a balanced metric that considers both the model's recall of real items and the accuracy of predictions.

$$\text{F1-Score}=2.\frac{\text{Precision.Recall}}{\text{Precision+Rec}} \quad (2)$$

Where

$$\text{Recall}=\frac{\text{True Positive}}{\text{True Positive+False Negative}} \quad (3)$$

False Negatives (FN) are items that the model was unable to identify. The F1 Score is a useful statistic that strikes a compromise between recall (the number of objects recognized) and precision (the accuracy of the detections), which makes it especially applicable to datasets with an uneven object distribution. Mean Average Precision (mAP) is a crucial metric for assessing object detection models' performance. For every kind of object, it integrates precision and recall while taking into account different degrees of overlap (IoU) between forecasts and real objects. Average Precision (AP), which averages precision across various recall levels, focuses on assessing the model's performance for a particular object type. Lastly, the average AP score for all object types in the collection is called mAP.

$$\text{AveragePrecision}=\int_0^1 \text{precision}(r)dr \quad (4)$$

Where recall is denoted by r. discrete points from the precision-recall curve are used to estimate this integral. All classes' AP values are averaged to determine mAP:

$$\text{mAP}= 1/\text{N}\sum_{i=1}^{N}\text{APi} \quad (5)$$

where the number of classes is N. The quantity of photos (or frames) the model can handle in a second is measured by FPS. For real-time applications where speed is a crucial component, such as autonomous driving or surveillance, this metric is crucial.

$$\text{FPS}=\frac{1}{\text{Interference Time Per Image (s)}} \quad (6)$$

Table 1: Performance Of Yolo Nas With Mask Rcnn Using Various Techniques On The Kitti Dataset

Models	Precision	F1-Score	mAP	FPS
SSD[25]	58.37	54.09	56.29	18
Faster R-CNN[26]	64.16	52.07	49.09	20
FCOS[27]	71.58	52.62	59.82	21
YOLOv5n	66.13	64.56	60.89	42
Proposed	77.34	75.25	76.56 **FPS**	86

Table 2: Performance Of Yolo Nas With Mask Rcnn Using Various Techniques On The Kitti Dataset

Models	Precision	F1-Score	mAP	FPS
YOLOv7-tiny[29]	67.92	62.62	65.36	62
YOLOv8-lite[31]	76.72	74.62	75.32	85
Proposed	77.34	75.25	76.56	86

The model is quicker and more suited for real-time jobs when the FPS number is higher. The comparison of several methods, including SSD, Faster-R-CNN, FCOS, YOLOv5n, and our method (Ours), shows how well our model performs on key measures, including Precision, F1 Score, mAP, and FPS, as shown in Table 1 and Table 2. With a precision score of 77.34, our model outperforms SSD (58.37), Faster-R-CNN (64.16), FCOS (71.58), and YOLOv5n (66.13).

This suggests that our approach generates fewer false positives, demonstrating its higher prediction accuracy. Furthermore, our model outperforms the other methods and shows its efficacy in precisely detecting objects while minimizing missed detections, as evidenced by its F1 Score of 75.25, which maintains a balanced performance in both accuracy and recall.

Our model considerably outperforms SSD (56.29), Faster-R-CNN (49.09), FCOS (59.82), and YOLOv5n (60.89) in terms of mean Average Precision (mAP), with 76.56. Its capacity to reliably recognize items across many classes and Intersection over Union (IoU) thresholds is demonstrated here, which is crucial for object recognition. Furthermore, our model performs exceptionally well in terms of processing speed, attaining 86 FPS, which is more than twice as fast as YOLOv5n (42 FPS) and noticeably faster than FCOS (21 FPS), SSD (18 FPS), and Faster-R-CNN (20 FPS). Our model is especially well-suited for real-time applications because of its remarkable frame rate, which provides both speed and accuracy. This suggests that our approach generates fewer false positives, demonstrating its higher prediction accuracy. Furthermore, our model outperforms the other methods and demonstrates its efficacy in precisely detecting objects while minimizing missed detections, as evidenced by its well-balanced performance in accuracy and recall (F1 Score of 75.25).

Our model's mean Average Precision (mAP) of 76.56 is only marginally lower than YOLOv8-lite (75.32), but higher than YOLOv7-tiny (65.36), and YOLOv8n (66.35). In order to achieve dependable object recognition, our model must be able to recognize objects accurately across a variety of classes and Intersection over Union (IoU) thresholds.

Furthermore, our model performs exceptionally well in terms of processing speed, attaining 86 FPS, which is much higher than YOLOv7-tiny (62 FPS) and YOLOv8n (57 FPS) and surpasses the performance of YOLOv8-lite (85 FPS). This remarkable frame rate highlights how providing both accuracy and speed.

Table 3: Comparative Analysis Of Proposed Method With Other Existing Methods

Model	Strength	Weakness
SSD	It is lightweight and simple to use.	F1-score (54.09) and precision (58.37) are the lowest. The inability to notice little objects.
Faster R-CNN	Strong ability to notice objects.	Processing speed is slow (20 FPS).Compared to other models, the mAP (49.09) is lower.
FCOS	Implementation is made easier with anchor-free design.	Lower F1-score (52.62) in contrast to variations of YOLO.
YOLOv5n	Balanced detecting performance and speed (42 FPS).	Little less accurate than the suggested model and YOLOv8-lite.
YOLOv7-tiny	62 frames per second is a high speed for real-time applications.	Lower mAP and F1-score in contrast to more recent models.
YOLOv8-lite	Strong F1-score (74.62) and high accuracy (76.72). Outstanding performance in real time (85 frames per second).	The computational cost is more than in previous iterations of YOLO.
Proposed Model	Best overall results in mAP (76.56), F1-score (75.25), and precision(77.34). It is perfect for real-time applications because it has the fastest processing speed (86 FPS).	Generalization necessitates validation on larger datasets .

From Table 3 we observe that the one recommended shows the best accuracy-to-speed relation of any model, making it the most efficient option for real-time object detection. Its increased detection performance is guaranteed by its increased precision and FPS, clearing the door for more rapid and more trustworthy applications in real-time vision systems and autonomous driving.

Four important metrics—precision, F1 Score, mean Average Precision (mAP), and frames per second (FPS)—are used in Figure 7 to compare different object detection techniques. These metrics are essential for evaluating how well detection algorithms work, especially in real-time situations. While the F1 Score strikes a compromise between precision and recall, precision quantifies how accurately things are detected. The processing

speed, or the number of frames per second that the model can process, is shown by FPS, while the overall detection accuracy is reflected by mAP.An F1 Score of 54.09, a mAP of 56.29, an FPS of 18, and a Precision of 58.37% are all attained by the SSD model. With an FPS of 20, Faster-R-CNN exhibits superior precision (64.16%), but its F1 Score and mAP are lower at 52.07 and 49.09, respectively. Although FCOS has a lower F1 Score of 52.62 and an FPS of 21, it scores better in terms of Precision (71.58%) and mAP (59.82). With 66.13% Precision, a 64.56 F1 Score, a mAP of 60.89, and an enhanced FPS of 42, YOLOv5n offers a better balanced performance.

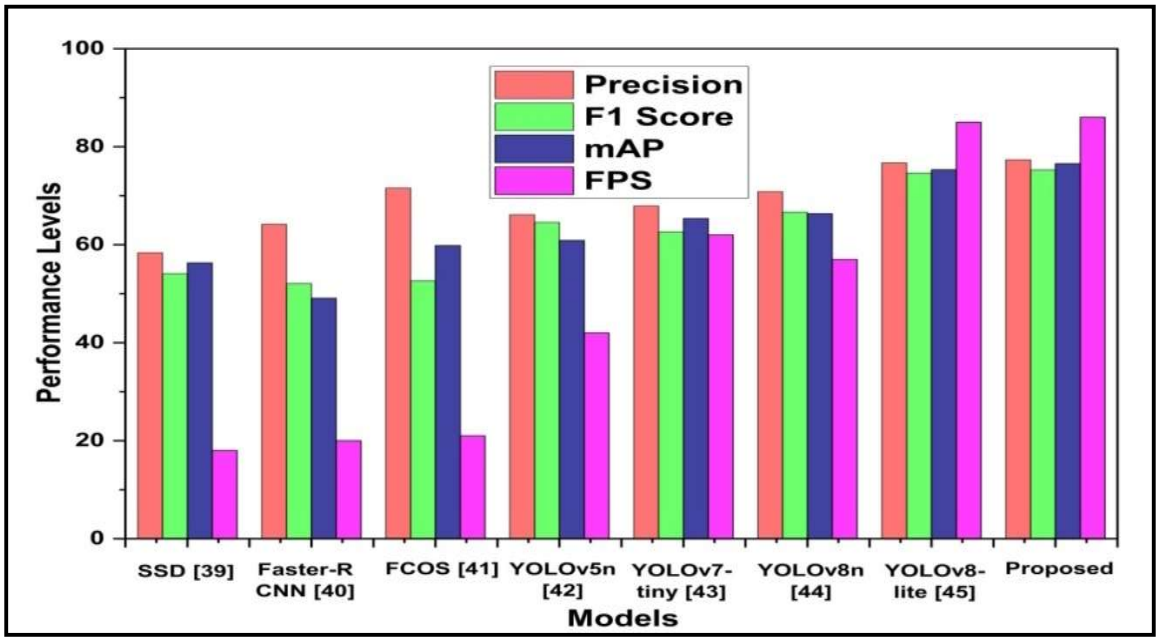

Figure 7: Visual Representation Of Proposed Model With Other Existing Model

Accuracy and speed are improved by the more modern YOLO models. YOLOv7-tiny records a mAP of 65.36, a 62.62 F1 Score, a 67.92% Precision, and 62 FPS. With 70.82% Precision, a 66.63 F1 Score, a mAP of 66.35, and 57 FPS, YOLOv8n demonstrates even more progress. With 76.72% Precision, a 74.62 F1 Score, a mAP of 75.32, and 85 FPS, YOLOv8-lite performs better than earlier editions. Our approach, which excels in accuracy and real-time capacity, finally shows the best performance, attaining 77.34% Precision, a 75.25 F1 Score, a mAP of 76.56, and an impressive 86 FPS. All things considered, our method proves to be an extremely accurate and effective object detection model. These exceptional results are clearly the result of the incorporation of sophisticated features like deformable convolution layers, which have increased the efficacy of your model for applications that need high precision and real-time processing, like autonomous driving or video surveillance.

5. CONCLUSION AND FUTURE WORK

The YOLO-NAS neck and head remain intact while a novel integration of YOLO-NAS and Mask R-CNN is shown in this work, which swaps ResNet + FPN for Rep Ne X t. The accuracy of feature extraction, small-object recognition, and segmentation is significantly enhanced by this upgrade, making it ideal for real-time applications such as video surveillance and autonomous driving. The model surpasses current techniques in terms of accuracy and processing speed, producing better results across key metrics such as precision, F1 score, mAP, and FPS. A slight rise in computational complexity comes forward by the integration, but this trade-off guarantees improved feature representation and segmentation quality—two essential elements of high-precision positions. The KITTI dataset is used for the evaluation, however in order to improve generalizability, future studies can extend testing to other datasets and difficult situations. Additional enhancements can involve domain adaptation for increased resilience,

attention mechanisms for better feature selection, and hybrid models for managing difficult scenarios. Optimizing the model for low-resource contexts, such edge computing and embedded systems, will also make it more useful for real-time applications. By addressing these issues, our work provides a new standard for recognition of objects and segmentation, offering a solid foundation for the next advancements in autonomous systems and real-time visual perception.

CONFLICTS OF INTEREST

None of the writers have any conflicts of interest.

ACKNOWLEDGEMENT

There was no financial assistance for this effort.

REFERENCES:

[1] Silva, L. A., Leithardt, V. R. Q., Batista, V. F. L., Villarrubia González, G. & De Paz Santana, J. F. Automated road damage detection using uav images and deep learning techniques. IEEE Access11,62918–62931 (2023).

[2] Dewi, C., Chen, R.-C., Jiang, X. & Yu, H. Deep convolutional neural network for enhancing traffic sign recognition developed on yolo V4. Multimedia Tools and Applications 81, 37821–37845 (2022).

[3] Botezatu, A.-P., Burlacu, A. & Orhei, C. A review of deep learning advancements in road analysis for autonomous driving. Applied Sciences14, (2024).

[4] S. Kolekar, S. Gite, B. Pradhan, A. Alamri, Explainable AI in scene understanding for autonomous vehicles in unstructured traffic environments on indian roads using the inception U-net model with Grad-CAM visualization, Sensors 22 (24) (2022) 9677 Vol 22, Page 9677.

[5] Bhattacharya, M. Agrebi, A. Roy, P.K. Singh, DFE-AVD: deep feature ensemble for automatic vehicle detection.; 2022

[6] Wei, F. & Wang, W. Gs-yolonet: A lightweight network for detection, tracking, and distance estimation on highways. In 2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring), 1–6, (2024).

[7] Yue, S., Zhang, Z., Shi, Y. & Cai, Y. Wgs-yolo: A real-time object detector based on yolo framework for autonomous driving. Computer Vision and Image Understanding 249, 104200 (2024).

[8] Zhang, Y. & Liu, C. Real-time pavement damage detection with damage shape adaptation. IEEE Transactions on Intelligent Transportation Systems (2024).

[9] Chen, Z., Yang, K., Wu, Y., Yang, H. & Tang, X. Hclt-yolo: A hybrid cnn and lightweight transformer architecture for object detection in complex traffic scenes. *IEEE Transactions on Vehicular Technology* (2024).

[10] Dai, Cui, J., Qin, Y., Wu, Y., Shao, C. & Yang, H. Skip connection yolo architecture for noise barrier defect detection using uav-based images in high-speed railway. IEEE Transactions on Intelligent Transportation Systems (2023).

[11] Li, Z. et al. Toward effective traffic sign detection via two-stage fusion neural networks. IEEE Transactions on Intelligent Transportation Systems (2024).

[12] Ma, F. et al. Identifying ships from radar blips like humans using a customized neural network. *IEEE Transactions on Intelligent Transportation Systems* (2024).

[13] Wu, Q., Li, N., Zhang, L. & Yu, F. R. Driver drowsiness detection based on joint human face and facial landmark localization with cheap operations. IEEE Transactions on Intelligent Transportation Systems (2024).

[14] Wei, F. & Wang, W. A method for designing the perception module of autonomous vehicles using stereo depth and semantic segmentation. In *2024 24th International Conference on Control, Automation and Systems (ICCAS)*, 1423–1428,(2024).

[15] Zhang, Z., Liu, F., Huang, Y. & Hou, Y. Detection and statistics system of pavement distresses based on street view videos. IEEE Transactions on Intelligent Transportation Systems (2024).

[16] Juan R. Terven, Diana M. Cordova-Esparaza. A COMPREHENSIVE REVIEW OF YOLO:FROMYOLOV1ANDBEYOND.DOI:https://ui.adsabs.harvard.edu/link_gateway/2023arXiv230400501T/doi:10.48550/arXiv.2304.00501

[17] Karishama Bisen, Rohini Shahare, Karishma Wasnik, Prof. P. Jaipurkar, International Research Journal of Modernization in Engineering Technology and Science. Volume:05/Issue:04/April-2023. DOI : https://www.doi.org/10.56726/IRJMETS35340

[18] Hao Yi, Bo Liu, Bin Zhao, Enhai Liu. Small Object Detection Algorithm Based on Improved YOLOv8 for Remote Sensing. January 2023 IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

PP(99):1-15 DOI: https://doi.org/10.1109/JSTARS.2023.3339235

[19] Minchu Kulkarni, Chu Li, Jaye Ahn, Katrina Ma, Zhihan Zhang, Michael Saugstad, Kevin Wu, Yochai Eisenberg, Valerie Novack, Brent Chamberlain, and Jon E. Froehlich. 2023. BusStopCV: A Real-time AI Assistant for Labeling Bus Stop Accessibility Features in Streetscape Imagery. 25th International ACM SIGACCESS Conference on Computers and Accessibility DOI: https://doi.org/10.1145/3597638.3614481

[20] Ali, Omari Alaoui, Omaima, El bahi, Mariame, Oumoulylte, Ahmad, El Al-laoui, Ahmed Elyoussefi, and Yousef, Farhaoui. 2023. Optimizing Emergency Vehicle Detection for Safer and Smoother Passages ACM ISBN 979-8-4007- 0019-4/23/05. DOI: https://doi.org/10.1145/3607720.3607728

[21] Peng Yang, Chuanying Yang, Bao Shi, Legen Ao, and Shaoying Ma. 2023. Research on Mask Detection Method Based on Yolov8. (ICCVIT 2023), August 25–28, 2023, Chenzhou, China. ACM, New York, NY, USA, 10 pages. DOI: https://doi.org/10.1145/3627341.3630411

[22] Mane, D., Kumbharkar, P., Sangve, S., Earan, N., Patil, K. and Bonde, S., 2024. A Metaphor Analysis on Vehicle License Plate Detection using Yolo-NAS and Yolov8. Journal of Electrical Systems, 20(1s), pp.152-164.

[23] Safaldin, M., Zaghden, N. and Mejdoub, M., 2024. An Improved YOLOv8 to Detect Moving Objects. IEEE Access.

[24] Yang, M. and Fan, X., 2024. YOLOv8-Lite: A Lightweight Object Detection Model for Real-time Autonomous Driving Systems. IECE Transactions on Emerging Topics in Artificial Intelligence, 1(1), pp.1-16.

[25] Zhichao Chen, Haoqi Guo, Jie Yang, Haining Jiao, Zhicheng Feng, Lifang Chen, and Tao Gao. Fast vehicle detection algorithm in traffic scene based on improved ssd. Measurement, 201:111655, 2022.

[26] Mohamed Othmani. A vehicle detection and tracking method for traffic video based on faster r-cnn. Multimedia Tools and Applications, 81(20):28347–28365, 2022.

[27] SP Krishnendhu and Prabu Mohandas. Sad: Sensor-based anomaly detection system for smart junctions. IEEE Sensors Journal, 2023.

[28] Xumeiqi Chen. Traffic lights detection method based on the improved yolov5 network. In 2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), pages 1111–1114. IEEE, 2022.

[29] Songjiang Li, Shilong Wang, and Peng Wang. A small object detection algorithm for traffic signs based on improved yolov7. Sensors, 23(16):7145, 2023.

[30] Emel Soylu and Tuncay Soylu. A performance comparison of yolov8 models for traffic sign detection in the robotaxi-full scale autonomous vehicle competition. Multimedia Tools and Applications, pages 1–31, 2023.

[31] Yang, M. and Fan, X., 2024. YOLOv8-Lite: A Lightweight Object Detection Model for Real-time Autonomous Driving Systems. IECE Transactions on Emerging Topics in Artificial Intelligence, 1(1), pp.1-16.

[32] Mohan, R. and Valada, A., 2021. Efficientps: Efficient panoptic segmentation. International Journal of Computer Vision, 129(5), pp.1551-1579.