10.48047/jocaaa.2024.33.07.49

Enhanced Real-Time Object Tracking and Segmentation Using YOLOv8n with ASCNN, HOF-SG, and SEAT

Guru Prasad M Bhat

Visvesvaraya Technological University, Belagavi 590018, India
Department of Information Science Engineering
R V College of Engineering
Bangalore, India-560059
guruprasadmb.phdis@rvce.edu.in
https://orcid.org/0000-0002-2634-3998

Nagaraj G. Cholli

Department of Information Science Engineering
R V College of Engineering
Bangalore, India-560059
nagaraj.cholli@rvce.edu.in
https://orcid.org/0000-0001-7409-8272

Abstract

Real-time object tracking and its segmentation from the surrounding environment is a complex process when there are strong occlusions and fast motion. This paper proposes a strong framework that fused YOLOv8n with Adaptive Scale Convolutional Neural Network (ASCNN), Hybrid Optical Flow with Semantic Guidance (HOF-SG), and Semantic-Enhanced Adaptive Tracking (SEAT) to improve the tracking and segmentation performances. The proposed solution satisfies the properties of variable object scales and high mobility to meet the needs of practical scenarios for experiments' results with per-class MOTA of 92.3% for cars and 81.7% for pedestrians, the overall detection accuracy of 98.28%, and F1 score of 97.99%. These results show that our framework can be applied to real-life applications such as self-driving cars or security cameras, and therefore it is a significant improvement for computer vision systems.

Keywords

Dynamic Environments; Neural Networks; Adaptive Scale CNN; Optical Flow; Autonomous Navigation

1. Introduction

Object tracking and segmentation are crucial components of various modern applications, including autonomous driving, surveillance, and robotics [1], [2], [3]. The ability to accurately track and segment objects in real-time enables systems to understand and interact with dynamic environments effectively [4], [5]. Though, these kinds of tasks are challenging because of occlusion, changes in lighting condition and multiple similar appearing objects.

Current research has shown that the employment of deep learning techniques has boosted the efficiency, accuracy in the object detection and tracking algorithms immensely [6], [7], [8]. Among these, the YOLO (You Only Look Once) series has gained widespread recognition for its balance of speed and accuracy [9]. YOLOv8, the latest iteration, builds upon its predecessors by incorporating state-of-the-art techniques to enhance performance further [10]. It utilizes a streamlined architecture that enables efficient feature extraction and processing, making it suitable for real-time applications.

However, there are still several issues that are open problems in the field of object segmentation and tracking. Multiple-object tracking (MOT) requires detecting objects in each frame and associating them correctly across consecutive frames [11]. Occlusions, abrupt movements, and dense environments complicate this task. Traditional methods, such as SORT and DeepSORT, have attempted to address these issues but often fall short in highly dynamic scenarios [12].

10.48047/jocaaa.2024.33.07.49

Some of the novel techniques, such as a softer Non-Maximum Suppression (softNMS), GhostConvolution, and C3Ghost Modules have been proven to enhance the detection performance while at the same time minimizing the number of parameters required in the model [13]. SoftNMS helps mitigate the issue of missed detections due to occlusion by adjusting the suppression of overlapping bounding boxes. GhostConv and C3Ghost Modules reduce the computational burden by sharing weights across multiple layers, enabling the deployment of complex models on resource-constrained devices.

Furthermore, combining object detection with sophisticated tracking algorithms, such as OC-SORT, enhances the system's ability to maintain accurate trajectories of multiple objects [13]. These algorithms leverage motion and appearance information to associate detections across frames, improving robustness in challenging conditions.

This work is aimed at enhancing the use of YOLOv8 in object tracking and segmentation by employing appropriate tracking methodologies. The approach described in this paper is designed to provide better performance in dynamic situations through the application of new deep learning approaches and algorithmic ideas. The reliability of this concept is well illustrated by extensive tests providing quantitative improvements both in tracking accuracy and robustness.

1.2 Motivation and Problem Statement

The general advancement of deep learning algorithms, and particularly the use of YOLO series, have greatly boosted the possibilities of object detection. Nevertheless, we have seen that effective tracking mechanisms that can cope with the complexity of real-world applications have not been integrated comprehensively. The rationale for this research is to overcome the problems associated with the existing tracking systems in the light of the advantages offered by YOLOv8. The reason for the enhancement of the object tracking and segmentation network is to achieve better performance, especially under adverse circumstances. Enumerated as follows, current research endeavors attempt to enhance the detection accuracy and efficiency of computation by applying sophisticated techniques. Moreover, tracking of moving objects has to be adaptive, which means that the system has to be able to track objects in various scales and with different movements, making it highly-useful for real-time applications in dynamic and unknown environments. The primary issues can be summarized as follows:

Handling Occlusions and Dense Environments: Traditional object tracking algorithms often fail in scenarios where objects frequently occlude each other or appear in densely populated environments. Occlusions lead to missed detections and tracking discontinuities, which degrade overall system performance. Improving the handling of occlusions is essential for maintaining accurate object trajectories.

Adaptability to Rapid Object Movements: In dynamic environments, objects can exhibit rapid and unpredictable movements. Existing tracking algorithms struggle to maintain consistent object identities, leading to frequent identity switches and tracking errors. Developing adaptive strategies that can quickly adjust to these movements is crucial for enhancing tracking reliability.

Resource Efficiency: Many applications, particularly those on edge devices, face significant computational constraints. Traditional deep learning models, while powerful, are often too resource-intensive for deployment on such devices. There is a need for lightweight yet practical models that can deliver high performance without exceeding resource limitations.

Variability in Object Scales: Due to varying distances from the camera, objects appear at different scales within the same scene. Effective object segmentation and tracking require algorithms capable of dynamically adjusting to these scale variations to ensure consistent performance.

This research seeks to develop a robust and efficient object tracking and segmentation system by integrating YOLOv8 with advanced adaptive tracking algorithms.

1.3 Research Objectives

The objectives of this research are as follows, and they are intended to improve the capability of current systems greatly:

To improve object detection and segmentation across various scales, enabling the system to accurately identify and delineate objects at different distances and sizes for detailed environmental mapping.

10.48047/jocaaa.2024.33.07.49

To add semantic information for improving the motion analysis by separating static from dynamic objects in the scene and thus improving the performance of the motion tracking.

To advance the tracking of objects using adaptive strategies based on their semantic characteristics and dynamic context, ensuring robust tracking even in complex and cluttered environments.

1.4 Research Contributions

The contributions of this research are pivotal in advancing the field of autonomous navigation systems:

This research introduces cutting-edge approaches by integrating ASCNN (Adaptive Scale Convolutional Neural Network) and HOF-SG (Hybrid Optical Flow with Semantic Guidance), and by implementing SEAT (Semantic-Enhanced Adaptive Tracking). These enhancements boost the system's environmental perception and interaction capabilities in dynamic settings.

The integration of ASCNN allows for precise detection and segmentation at multiple scales, crucial for detailed environmental awareness. The novel SEAT algorithm further enhances the system's ability to track moving objects accurately, adapting tracking strategies based on semantic data and object dynamics.

The development of HOF-SG with SEAT significantly enhances motion analysis, providing superior tracking accuracy for dynamic objects, which is essential for effective navigation and interaction within changing environments.

1.5 Organization of the Paper

Section 2 reviews the existing literature on object tracking and segmentation. It also discusses the limitations and challenges identified in previous studies. Section 3 details the proposed approach, including the integration of the YOLOv8n model with advanced adaptive tracking strategies. It elaborates on the implementation and optimization of the ASCNN, HOF-SG, and SEAT algorithms. It describes the overall system architecture, focusing on the interaction between different components. It provides a comprehensive overview of data flow and processing steps involved in object detection, segmentation, and tracking. Further, this section outlines the processes involved in acquiring and preparing data for training and validating the enhanced system. It includes details on the datasets used, preprocessing steps, data augmentation techniques, and training-validation splits. Section 4 presents the experimental results, highlighting the performance gains achieved by the proposed method. Section 5 summarizes the key contributions and findings of the research.

2. Related Works

Deep learning techniques have brought meaningful and critical changes in object detection and tracking as a domain, mostly with the help of the algorithms [10], [14], [15]. These advancements have been critical in enhancing real-time processing capabilities and improving accuracy in various applications.

The YOLO series has undergone continuous refinement, with each version introducing novel techniques and architectural enhancements to improve performance. YOLOv8, the latest iteration, builds upon the successes of its predecessors while incorporating state-of-the-art advancements. Notably, YOLOv8 has found applications in diverse domains, including autonomous driving systems for pedestrian tracking [13], real-time traffic monitoring systems [16], and even the detection of moving objects in various visual contexts [17]. These applications highlight the versatility and adaptability of YOLOv8 in addressing real-world challenges. Additionally, the integration of tracking algorithms like DeepSORT [18], [19] and ByteTrack (Bergmann et al., 2022) has enabled robust object tracking in complex scenarios. The incorporation of deformable convolutions and self-calibration shuffle attention mechanisms has improved its ability to detect multi-scale and occluded objects [20].

Researchers have been actively exploring various techniques to enhance the capabilities of YOLOv8. For instance, the integration of softNMS and Ghost models has been proposed to improve the detection of small objects in UAV images [21]. The utilization of GhostNet modules has also been investigated to optimize the YOLOv8 architecture by reducing model complexity while maintaining performance, making it suitable for deployment on resource-constrained devices [13], [17]. Additionally, the incorporation of the OC-SORT tracking algorithm and a MobileNetV2-based REID model has shown promise in achieving robust pedestrian tracking in complex scenarios [13]. These advancements underscore the ongoing efforts to refine YOLOv8 for specific tasks and improve its

10.48047/jocaaa.2024.33.07.49

overall performance. Researchers have explored its application in diverse fields, such as autonomous driving for pedestrian tracking [22], medical imaging for surgical instrument segmentation [23], and even behavioral research for automatic object detection [24].

YOLOv8 is not limited to pedestrian detection and tracking but come with various applications related to object detection and tracking. In the medical sector, YOLOv8 has been applied for the detection of polypoid lesions in Wireless Capsule Endoscopy (WCE) images [25]. This application is crucial for the early detection and prevention of gastrointestinal cancers. Furthermore, YOLOv8 has been utilized in dynamic environments for simultaneous localization and mapping (SLAM) [26]. Combining the instance segmentation and video completion algorithms has enhanced localization accuracy and mapping of SLAM systems when there are dynamic objects close to the camera. For real-time multiple-object detection and tracking in the field of autonomous aerial vehicles YOLOv8 among other deep learning algorithms has been applied [27]. This has been done using embedded devices with GPU which make small flying robots perform the on-board computations needed in object tracking which is a clear dictate of how YOLOv8 can be used to support autonomous flying and decision-making by UAVs.

YOLOv8 has been integrated with Light Detection and Ranging (LiDAR) data fusion technology to improve object detection and tracking in real-time [28]. This integration takes advantage of YOLOv8's efficiency and LiDAR's capability to work out the perception of the environment. LiDAR has turned into a central facet in use cases like autonomous vehicles or humans and robot's assistant as it performs exceptionally well within complex conditions. Juan Gómez, directly approached issues on detecting, tracking, and classifying persons with the use of a 3D LiDAR sensor: Object segmentation with classification by local geometric descriptors and tracking solutions. This approach takes the advantage of achieving real time performance even on computers with low computational power by processing fewer points while at the same time providing consistent detection and tracking of persons even in adverse environment [29].

Through pretrained object detectors, MOTRv2 expands the usage of end-to-end methods for enhancing multi-object tracking. In contrast to conventional methods that yield unsatisfactory detection performance, MOTRv2 utilizes an additional object detector to provide proposals, thereby alleviating the competition between jointly learned detection and association problems. Such approach has shown high performance in benchmarks such as DanceTrack and BDD100K, proving its ability to work at large scale in tracking tasks [15]. A modified threshold-based approach used in motion probability estimation aids in improving positioning precision by discarding feature points with high motion probabilities [30].

Fischer et al. , presents quasi-dense similarity learning for multi-object appearance based tracking. This method densely samples object regions for this contrastive learning process and works along with existing object detectors for nearest neighbour search for object association at the inference phase. Even though QDTrack is rather simplistic, it offers state-of-the-art performance on benchmarks such as BDD100K thereby proving the benefits of quasi-dense similarity learning in enhancing tracking performance [31].

Zhao et al., presented zero-shot video object segmentation (ZVOS), reducing its dependency on activity-based features unlike conventional approaches. Their model transform RGB sources into depth and static saliency sources and the multi-source fusion structure emphasize the spatial significance of each of the sources. It enhances the representation of the static and moving features, which improves performance in achieving high scores in the ZVOS benchmarks[2].

Benjamin Drayer and Thomas Brox presented a method combining frame-level object detection with tracking and motion segmentation concepts. This method extracts temporally consistent object tubes based on off-the-shelf detectors and integrates motion cues for accurate video segmentation. The approach overcomes common issues in weakly supervised video segmentation, demonstrating robust performance across multiple video segmentation datasets [32].

In precision agriculture, detecting, segmenting, and tracking fruits like table grapes can be challenging due to occlusion and varying illumination. The methods have shown high performance with minimal labeled data, enabling practical applications in agricultural robotics [33]. BURST (benchmark unifying multiple tasks) facilitates the development of generalized methods by providing high-quality object masks and consistent metrics

10.48047/jocaaa.2024.33.07.49

across diverse tasks. This approach enables more effective knowledge pooling from different methods and tasks, advancing the field of object tracking and segmentation [34].

In a survey, the authors highlighted the challenges of blurred foregrounds, irregular object movements, and severe occlusions. The survey discusses the potential of transformers in overcoming these difficulties and outlines open challenges for future research, underscoring the importance of transformers in advancing remote sensing applications [35].

The Segment And Track Anything (SAM-Track) framework, developed by combining the Segment Anything Model (SAM) with an AOT-based tracking model for precise object segmentation and tracking. SAM-Track supports multi-modal interaction methods, including click, stroke, and text-based inputs, enabling versatile applications across fields such as drone technology, autonomous driving, and medical imaging. This framework has demonstrated high performance on datasets like DAVIS-2016 and DAVIS-2017, showcasing its practicality in diverse scenarios [36].

Real-time object recognition was proposed for real-time application by [37] with a YOLOv3-inspired model which was trained only with KITTI dataset and named MelNet [38]. The Average Precision (mAP) was measured as about 0.732 at around 300 epochs of training; therefore, it was valuable in practical applications. It is evident that the architecture of MelNet is rather rigid and efficient, hence the model is challenging to apply in more dynamic scenario, apart from being unable to capture details with fairly large variance under different lighting and environmental conditions, despite good performance. As compared to other MelNet models, it is good for quicker processing however its accuracy may vary in occlusion or when the backdrop is less predictable when compared to more developed YOLO versions or corresponding adaptive tracking models.

İn a recent study, SS-SFR helps characterize the effectiveness of models in emulating environments made of simulated data, the use of such data poses questions about the ability of models to handle variability of actual data [39]. Thus, the results focus on the dissemination between synthetic and actual evaluations in practical contexts and indicate that merely relying upon synthetic evaluations is a weakness.

[40] proposed an object identification system for AMRs using CNN and SSD Mobilenetv2 FPN Lite 320×320 architecture. The proposed model was tested on the KITTI dataset, especially focusing on the objects detection at different lighting conditions: night and day, and it yields mAP score up to 0.477 for big objects [40]. New deep learning architectures use multi-scale parts or attention mechanisms and they might help enhance the detection accuracy of new CNN 2024 for small, partially obscured objects which are in motion or in a crowded scene.

These challenges include handling occlusions, varying lighting conditions, and the presence of numerous targets with similar appearances [41], [42]. Researchers are actively exploring solutions to address these issues, such as the integration of attention mechanisms, the development of more sophisticated tracking algorithms, and the exploration of semi-supervised learning approaches [17]. The future of deep learning-based object detection and tracking is promising, with ongoing research and development efforts to further enhance the accuracy, efficiency, and robustness of these systems. As technology evolves, we can anticipate even more innovative applications and advancements in this field.

2.1 Research Gaps

Several critical gaps hinder the deployment of robust systems in real-world applications. Integrating advanced deep learning techniques such as YOLOv8 and tracking algorithms like SORT, DeepSORT OC-SORT has improved performance; however, key challenges remain unaddressed.

Adaptation to Rapid Object Movements: Existing tracking algorithms often fail to maintain consistent object identities during rapid and unpredictable movements. Techniques like OC-SORT have improved, but there is still a significant gap in handling fast-moving objects without frequent identity switches and tracking errors.

Resource Efficiency: Deploying high-performance object detection and tracking systems on resource-constrained devices remains challenging. Traditional deep learning models require substantial computational resources, making them impractical for real-time applications on edge devices. Lightweight models that can deliver high accuracy without overwhelming the available resources are needed.

10.48047/jocaaa.2024.33.07.49

Variability in Object Scales: Effective segmentation and tracking require algorithms that can dynamically adjust to different object scales within a scene. Current methods often lack robustness in handling varying object sizes, leading to inconsistent performance. Techniques like deformable convolutions have been explored, but further research is needed to enhance scale-invariant detection and tracking.

Addressing these gaps is essential for developing a robust and efficient object tracking and segmentation system. The proposed work aims to integrate YOLOv8 with advanced adaptive tracking strategies to tackle these challenges, thereby enhancing the reliability and applicability of object tracking systems in dynamic and complex environments.

3. Methodology

İn the proposed work, ASCNN is tailored to dynamically adjust to varying object scales within complex urban environments, utilizing atrous convolutions for enhanced scale perception and accuracy in segmentation tasks. ASCNN was chosen because it can handle objects of different sizes through scale changes in this dynamic scene. However, traditional CNNs fail to recognize objects at different scales or distances, using dilated convolutions, ASCNN extends the receptive field while maintaining the resolution. It makes it particularly useful in the densely populated areas where objects are found in different sizes. The capacity to extract high-resolution features without additional computational overhead is another advantage for ASCNN in real-time application areas, as its competitors cannot dynamically scale without excessive costs.

Concurrently, HOF-SG innovatively combines traditional optical flow techniques with semantic output from ASCNN, enriching motion vector calculations with contextual object behavior insights. As mentioned, the semantic information is combined with optical flow and makes the system more effective in separating static and moving parts. Most optical flow models have limited reference context hence high causative failure when in dynamic scenes [43]. Semantic segmentation is employed in HOF-SG for improving the motion vectors elaboration according to the objects' traits, thereby enabling accurate motion tracking to be achieved. Combined, it allows for a more accurate perception of scene dynamics, when compared to simple optical flow approaches that may fail in complex urban environment.

Adding to these, SEAT further refines the tracking mechanism by adaptively adjusting its strategies based on semantic attributes, enhancing the system's ability to maintain consistent tracking of objects through complex maneuvers and interactions. SEAT controls the tracking parameters as a reaction to the semantic characteristics of objects to track and consistently track movements even if those occur at extraordinary speed or objects are temporarily occluded. At this point, SEAT successfully controls the identification process by changing tracking methodologies according to object characteristics and their behavior; SEAT's control outperforms fixed tracking algorithms such as SORT and DeepSORT. In the case of SEAT, the adaptive framework is also advantageous in that those models are highly sensitive to the three phenomena: identity switching and tracking errors in the complicated environment, thus making the overall system more robust.

These enhancements are systematically validated against the KITTI dataset, ensuring robustness and efficacy in real-time navigation scenarios. The workflow diagram of the proposed work is given below (see figure1).

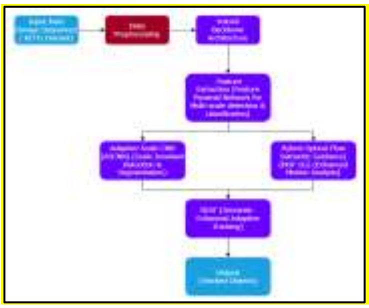


Fig.1: Proposed Methodology Flow Diagram

3.1 System Architecture

10.48047/jocaaa.2024.33.07.49

The architectural integration of YOLOv8 into the proposed system is emphasized, highlighting how this backbone enhances object detection, semantic understanding, and feature extraction within complex and dynamic environments.

3.1.1 Overview of YOLOv8 Integration

YOLOv8 is the backbone architecture for feature extraction in the proposed system. Its advanced convolutional network structure is tailored for high-speed, accurate object detection and classification, making it ideal for real-time applications where rapid environmental changes occur.

Convolutional Feature Extraction: The core of YOLOv8 comprises several layers of convolutional networks that process the input image to detect objects and classify them. The convolution operation in the backbone can be represented as:

$$F(x) = W * x + b \qquad (1)$$

where F(x) is the feature map produced by applying the weight W to the input x, and b is the bias.

Feature Pyramid Network for Multi-Scale Detection: To capture objects at different scales, YOLOv8 employs a Feature Pyramid Network (FPN) which aggregates features from various depths within the network:

$$P_i = Conv_{1x1}(C_i) + \text{upsample}(P_{i+1}) \qquad (2)$$

where P_i is the feature map at level i, C_i is the output of the last layer of the backbone at the same level, and upsample denotes up-sampling the feature map from the more profound level.

3.1.2 Functionality in Proposed System

The feature maps and detection outputs from YOLOv8 are intricately woven into the fabric of the proposed system to enhance both the mapping and navigation processes:

Semantic Augmentation of the Map: Using YOLOv8's classification capabilities, the proposed system enriches its environmental map with semantic information about the objects, such as 'car', 'pedestrian', or 'Truck. This semantic layer is crucial for distinguishing between static and dynamic elements, improving navigation decisions in real time.

Dynamic Object Tracking: The system employs the detected bounding boxes and class probabilities to track the movement of objects across frames. This is vital for updating the positions of dynamic objects within the SLAM map and adjusting the vehicle's or robot's trajectory accordingly. The motion vector v is computed based on changes in bounding box coordinates over time, providing a measure of object velocity:

$$v_t = \frac{\Delta \text{bbox}_t}{\Delta t} \qquad (3)$$

where Δbbox_t represents the change in bounding box coordinates between consecutive frames, and Δt is the time interval.

Integration of SEAT: To further enhance object tracking capabilities, the SEAT algorithm is incorporated. SEAT uses semantic output from both YOLOv8 and the proposed system to adapt tracking strategies dynamically, allowing the system to maintain accurate tracking under varying conditions, such as occlusions and rapid movements. This addition strengthens the system's ability to handle highly dynamic scenarios by ensuring consistent and reliable tracking of objects identified by YOLOv8.

3.1.3 Real-Time Processing Considerations

The efficiency of YOLOv8 ensures that feature extraction and object detection tasks are performed swiftly, keeping the processing time within the limits required for real-time operation. This is crucial for maintaining the fluidity of the proposed system's response to dynamic changes in the environment.

3.2 Data Acquisition and Preparation

10.48047/jocaaa.2024.33.07.49

This section outlines the processes involved in acquiring and preparing data for training and validating the enhanced exisiting system with the YOLOv8 backbone. The primary dataset used in this research is the KITTI dataset, well-known for its rich collection of annotated images and videos captured in urban driving settings.

Data Collection: The KITTI dataset includes diverse scenarios involving various dynamic and static objects such as vehicles, pedestrians, and urban landscapes, captured under different lighting and weather conditions. This dataset provides a realistic setting for developing and testing the proposed system's capabilities in recognizing and adapting to dynamic environments.

Data Preparation: The raw data from the KITTI dataset requires several preprocessing steps before it can be used for training the neural network models:

Image Resizing: All images are resized to a standard dimension of 640 x 640 pixels to match the input size requirement of the YOLOv8 model. This uniformity is crucial for ensuring consistent feature extraction across all images.

Annotation Conversion: The bounding box coordinates and object class labels provided in the KITTI dataset are converted into the format required by YOLOv8. This typically involves adjusting the coordinate system used in the dataset to the one used by the neural network.

Data Augmentation: To enhance the model's robustness and ability to generalize, data augmentation techniques such as flipping, rotation, and scaling are applied to the training images. These transformations help the model learn to recognize objects from various angles and under different scales.

Data Normalization: Before feeding the images into the network, they are normalized to have pixel values in a standard range, typically between 0 and 1. This normalization helps speed up the training process by ensuring that the network weights adjust more uniformly during backpropagation.

Training and Validation Split: The prepared data is divided into two sets: a training set and a validation set. Typically, about 80% of the data is used for training, and the remaining 20% is reserved for validation. This split allows the performance of the models to be evaluated on unseen data, ensuring that the improvements in the proposed system are not merely a result of overfitting to the training data but are genuinely effective in improving the system's accuracy and robustness.

3.3 Neural Network Models

The following models, namely the ASCNN, HOF-SG and SEAT, are pivotal for improving the system's operational efficacy in complex settings. Each model is tailored to address specific aspects of the SLAM process, from robust object recognition to precise motion tracking, which are critical for navigating and interacting with dynamic environments.

The ASCNN model is engineered to enhance the system's capability to recognize and segment objects accurately across different scales. This feature is particularly important in urban settings, where the scale of objects can vary significantly, from nearby pedestrians to distant buildings. The model's architecture is optimized to maintain high-resolution feature detection without sacrificing processing speed, which is crucial for real-time applications.

On the other hand, the HOF-SG model integrates semantic data into traditional optical flow algorithms to differentiate effectively between static and moving parts of the scene. This differentiation allows the proposed system to update its environmental model more accurately and make more informed navigation decisions based on the dynamics of the surrounding environment.

SEAT a novel addition to the system, further refines the tracking process by adapting tracking strategies based on the semantic characteristics of each object. It uses the semantic output from ASCNN to adjust tracking parameters dynamically, ensuring that the tracking mechanism is contextually aware and capable of handling abrupt changes in object motion or appearance. This adaptive tracking approach is particularly effective in scenarios where objects may occlude or interact, requiring sophisticated tracking algorithms to maintain accurate object trajectories.

3.3.1 Adaptive Scale Convolutional Neural Network (ASCNN)

10.48047/jocaaa.2024.33.07.49

The ASCNN plays a pivotal role in the enhanced proposed system, primarily designed to handle the varying scales of objects encountered in dynamic urban environments. Figure 2 depicts the architecture of ASCNN model.

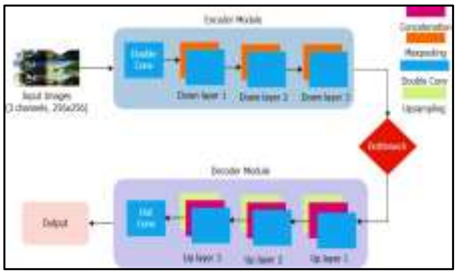


Fig. 2: ASCNN Architecture

Architecture Overview: The ASCNN architecture is structured to efficiently process multi-scale information from input images, ensuring robust feature extraction across varying object sizes and distances. The network utilizes a series of atrous convolutions, also known as dilated convolutions, which allow the network to expand its receptive field without loss of resolution or coverage. The above diagram (figure 2) presents the proposed ASCNN model which has been given inspiration from U-Net and mainly focuses on an encoder network, bottleneck layer, and a decoder layer. Encoder Module Inputs to this module include images (3 channeled, 256x256). To it, images are fed through the DoubleConv block then three Down layers are applied to it. Each Down layer contains MaxPooling for downsampling purpose and DoubleConv block for feature extraction. The bottleneck layer links the encoder and decoder layers and includes the smallest dimension but the greatest feature detail. In the decoder module, the there Up layers perform Up sampling, concatenate with corresponding features from the encoder and a Double Convolution block for feature enhancement. Finally, pixel level classification with the help of final OutConv layer provides an output segmentation map.

Atrous Convolution: The atrous convolution is defined mathematically for a two-dimensional input X with a kernel K as follows:

$$Y(i,j) = \sum_{m,n} X(i + r \cdot m, j + r \cdot n) \cdot K(m,n) \qquad (4)$$

where (i, j) are the spatial coordinates of the output feature map Y, (m, n) indexes the kernel K, and r represents the dilation rate, controlling the spacing between the kernel points.

Feature Aggregation: To effectively combine features extracted at different scales, the outputs of multiple atrous convolutions with varying dilation rates are concatenated. This aggregation can be expressed as in equation 5:

$$F_{agg} = \text{concat}(F_{d1}, F_{d2}, \dots, F_{dk}) \qquad (5)$$

where $F_{d1}, F_{d2}, \dots, F_{dk}$ represent the feature maps obtained from atrous convolutions at dilation rates $d1, d2, \dots, dk$.

Training Mechanism: Training the ASCNN involves optimizing a loss function that measures the discrepancy between the predicted object scales and the actual scales from the ground truth. The network employs a cross-entropy loss function modified to prioritize scale accuracy:

$$\mathcal{L} = -\sum_{c=1}^{C} y_{o,c} \log(p_{o,c}) \qquad (6)$$

where $y_{o,c}$ is the binary indicator if class label c is the correct classification for observation o, and $p_{o,c}$ is the predicted probability of observation o being of class c.

3.3.2 Hybrid Optical Flow with Semantic Guidance (HOF-SG)

The HOF-SG is designed to enhance the motion analysis capabilities of the proposed system by integrating semantic information into the optical flow computation process. This integration allows for a more nuanced understanding of object movements, especially in dynamic urban environments. The figure 3 depict the workflow of HOF-SG which integrate the optical flow together with information of semantic segmentation for getting better

10.48047/jocaaa.2024.33.07.49

in tracking. Input Images (x1, x2), representing consecutive frames, are fed into two parallel networks: the Optical Flow Network and the Semantic Segmentation Network. The optical flow network gives motion map capturing the movements of each pixel at the frame level, while the Semantic Segmentation network provides the semantic map, which assigns one of the masks a semantic label. These maps are then fused in the hybrid feature aggregation module, which integrates motion and semantic information to produce a refined, context-aware Output, which is appropriate for track and activity extraction.

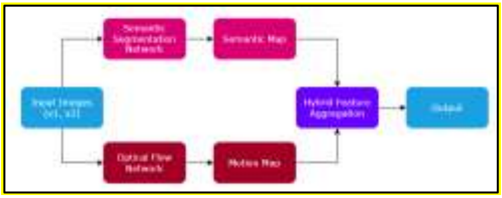


Fig. 3: Workflow of HOF-SG

Operational Principle:

The HOF-SG module operates by first calculating the optical flow between consecutive image frames and then adjusting this flow based on semantic cues from detected objects. This dual approach leverages both geometric and semantic data to provide a more accurate and robust estimation of object and camera motion.

Optical Flow Calculation: The optical flow vector v between two consecutive frames I_t and I_{t+1} can be computed using the Lucas-Kanade method under the brightness constancy assumption:

$$I_t(x, y) \approx I_{t+1}(x + \Delta x, y + \Delta y) \qquad (7)$$

where Δx and Δy are the displacements in x and y directions, respectively. The optical flow equation can then be expressed as:

$$f_x u + f_y v + f_t = 0 \qquad (8)$$

where f_x and f_y are the spatial derivatives, u and v are the horizontal and vertical components of the optical flow, and f_t is the temporal derivative.

Semantic Guidance Integration: The optical flow vectors are modified using the semantic masks produced by the ASCNN, which classify regions of the image according to object type and motion state. This modification is mathematically represented as:

$$v' = v \odot (1 - M) \qquad (9)$$

where v is the original flow vector, M is the semantic mask where dynamic objects are marked, and $\odot$ denotes element-wise multiplication. This operation effectively nullifies the flow in regions occupied by dynamic objects, focusing the flow analysis on static background elements.

Training and Loss Function:

Training the HOF-SG involves optimizing a combination of the traditional optical flow loss and a semantic guidance loss. The total loss $\mathcal{L}$ can be defined as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{flow}(v, v') + \lambda_2 \mathcal{L}_{semantic}\left(M, \widehat{M}\right) \qquad (10)$$

where $\mathcal{L}_{flow}$ is the optical flow loss comparing the predicted and adjusted flow vectors, $\mathcal{L}_{semantic}$ is the loss assessing the accuracy of the semantic segmentation, $\widehat{M}$ is the predicted semantic mask, and λ_1 and λ_2 are weighting factors balancing the two loss components.

Optimization and Efficiency:

The adjusted optical flow vectors v' are utilized within the proposed framework to update the system's understanding of object and camera movements. This enhanced motion analysis supports more accurate pose estimation and map updating, crucial for navigating and interacting dynamically with the environment. To ensure real-time performance, HOF-SG is optimized through techniques such as pyramidal implementation of the Lucas-

10.48047/jocaaa.2024.33.07.49

Kanade method, which reduces computational complexity while preserving the fidelity of motion estimation. Additionally, GPU acceleration is employed to handle the intensive computations involved in optical flow and semantic segmentation.

3.3.3 Semantic-Enhanced Adaptive Tracking (SEAT)

SEAT, a testament to adaptability, integrates semantic data into the tracking process, significantly enhancing the tracking accuracy and adaptability of proposed systems in complex environments. This approach leverages detailed semantic information from prior stages to guide the tracking algorithms, enabling dynamic adjustments based on the context and behaviors of different objects. Figure 4 outlines the workflow of the adaptive tracking by using both semantic and initial state information. To the concatenation block, semantic labels and initial features are provided and combined into a single feature map. This concatenated map is then passed through different ConvBlocks (namely ConvBlock 1 and 2, and ConvBlock 3) to generate feature representation. This is passed to an attention block, where spatially adaptive attention mechanism is used to focus on significant features. After attention, ConvBlocks 4 and 5 further refine the feature maps. Lastly, the OutConv layer transforms them into the Output, producing the last output map for tracking purposes. This architecture makes use of both semantic and spatial attention to generate attention-aware tracking outcomes.

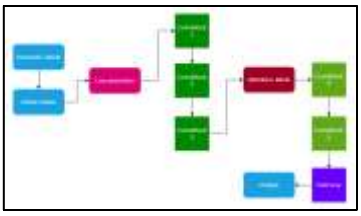


Fig. 4: Workflow of SEAT

Components:

SEAT contains several components such as, semantic analysis for identifying objects, adaptive tracking, contextual adaptation for exploiting context knowledge, scale invariance, and an attention mechanism to adjust tracking parameters based on object features.

Semantic Analysis: SEAT utilizes outputs from ASCNN and HOF-SG to generate detailed semantic labels and attributes for each detected object. These labels capture information such as object type (e.g., pedestrian, vehicle), size, and contextual clues, allowing SEAT to fine-tune the tracking strategies based on object characteristics.

Convolutional Feature Extraction: After concatenating the semantic labels and initial states, SEAT processes the input through a series of ConvBlocks that progressively refine the feature representation. Each ConvBlock comprises a convolutional layer, batch normalization, and ReLU activation. This feature extraction stage captures multi-scale spatial features, enabling SEAT to construct a rich and deep representation of the input data that is critical for accurate tracking.

Attention Block for Adaptive Focus: The Attention Block is a crucial element that allows SEAT to dynamically emphasize relevant areas of the feature map. To reduce computational load, this block downsamples the feature map spatially before calculating attention weights. The block generates query (Q), key (K), and value (V) matrices, where the attention scores are computed using:

$$Attention(Q, K, V) = Softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right)V \quad (11)$$

where d_k represents the dimensionality of the key. This mechanism enables SEAT to selectively prioritize the most informative regions within the feature map, refining its spatial focus based on the contextual importance of each region.

Adaptive Tracking Algorithm: SEAT's tracking algorithm dynamically adjusts tracking parameters using a semantic adjustment function that considers an object's semantic label and attributes.

Contextual Adaptation and Scale-Invariant Tracking: SEAT adapts its tracking strategy based on scene context and object scale. For instance, in crowded scenes, it may prioritize identity preservation during occlusions, while

10.48047/jocaaa.2024.33.07.49

in simpler scenes, it focuses on precision. Scale information from ASCNN enables SEAT to maintain effectiveness across varying object sizes, ensuring robustness in diverse tracking scenarios.

Object State Estimation:

Let S_t^i represent the state of object i at time t, typically including position p_t^i and velocity v_t^i. The state update equation, using a simple motion model under constant velocity, is given by:

$$S_t^i = S_{t-1}^i + \Delta t \cdot \dot{S_{t-1}^i} + W_t^i \qquad (12)$$

where Δt is the time increment, $\dot{S_{t-1}^i}$ is the derivative of the state (i.e., velocity), and W_t^i is the process noise, modeled as Gaussian.

Semantic Adjustment Function:

A semantic adjustment function $f(\cdot)$ modifies the tracking parameters based on the object's semantic label L_t^i and attributes A_t^i such as size and speed:

$$P_t^i = f\left(L_t^i, A_t^i, \Theta\right) \quad (13)$$

where P_t^i represents the set of tracking parameters for object i at time t, and Θ denotes a set of predefined rules or machine learning model parameters.

Adaptive Tracking Equation:

The tracking update, considering semantic adjustments, is represented as:

$$\widehat{S_t^i} = S_t^i + \alpha \cdot \left(P_t^i \odot H_t^i\right) \quad (14)$$

where $\widehat{S_t^i}$ is the estimated state after applying the semantic adjustments, α is a learning rate-like factor that controls the adaptation speed, ⊙ represents element-wise multiplication, and H_t^i is the semantic influence vector derived from P_t^i.

Optimization:

The parameters of $f(\cdot)$ and α are optimized via a loss function that minimizes the prediction error over a training dataset, typically using backpropagation in a machine learning context:

$$\mathcal{L} = \sum_{t,i} | \widehat{S_t^i} - S_t^{i,\text{true}} |_2^2 \qquad (15)$$

where $S_t^{i,\text{true}}$ is the true state of object i at time t, and the norm $| \cdot |_2$ denotes the Euclidean distance.

3.4 Experimental Setup

The setup includes detailed simulations and real-world testing environments using the KITTI dataset, which provides diverse urban driving scenes, including dynamic objects like vehicles and pedestrians, making it ideal for our purposes [44].

Testing Environment: The experiments use a standardized simulation platform replicating the KITTI environment. This ensures that the testing conditions remain consistent across all experiments, allowing for reliable comparisons of performance metrics.

Data Handling: For each test, the data is pre-processed per the specifications outlined in section 3.2, ensuring that it is appropriately formatted and normalized for input into the proposed system. The dataset is divided into training, validation, and testing segments, with the testing set exclusively reserved for evaluating the final model performance to prevent data leakage.

Parameter Settings: Detailed parameter settings used in the experiments are provided to maintain transparency and reproducibility. These settings include configurations for the neural networks, simulation parameters, and hardware specifications.

Table – 1: Experimental parameters:

10.48047/jocaaa.2024.33.07.49

Parameter	Description	Value/Setting
Neural Network Framework	Software used for implementing neural networks	PyTorch
Hardware	Hardware specifications for running the experiments	GPU: NVIDIA RTX 4070 Ti
Batch Size	Number of training examples per batch	32
Learning Rate	Initial learning rate for the optimizer	0.001
Epochs	Number of full training cycles	20
Optimizer	Algorithm for optimizing network weights	Adam
Loss Function	Criteria for measuring model performance	Cross-Entropy (ASCNN), MSE (HOF-SG)
Input Size	Dimension of input images	640 x 640 pixels
Testing Frequency	Frequency of model evaluation during training	After every epoch

4. Results and Discussion

4.1 Object Detection

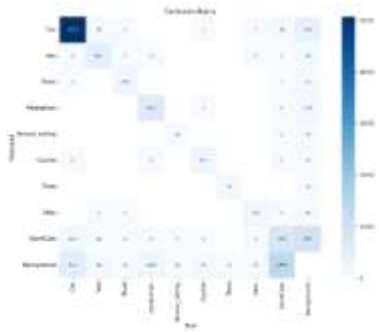
Table 2 presents an analysis of the proposed system against baseline models. The proposed techniques, ASCNN, HOFSG, and SEAT, were evaluated against traditional object detection and segmentation techniques like Fast R-CNN, YOLOv3, and YOLOv8n. Fast R-CNN showed moderate performance with 83.00% accuracy and an F1-Score of 83.50%, but it had a higher RMSE of 1.5000, indicating more localization errors. YOLOv3 improved upon this, reaching 86.00% accuracy and an F1-Score of 86.50%, with a lower RMSE of 1.2000, demonstrating better localization and segmentation capabilities. YOLOv8n further advanced performance with 91.00% accuracy and an F1-Score of 91.50%, supported by a reduced RMSE of 0.9000, reflecting its refined detection and segmentation abilities. Among the proposed methods, ASCNN achieved the highest accuracy at 98.28%, with a minimal RMSE of 0.6773, indicating superior precision in object detection and segmentation. HOFSG, with 97.51% accuracy and an RMSE of 0.7683, also outperformed the traditional methods. SEAT matched ASCNN in accuracy at 98.28% and had a slightly higher RMSE of 0.6855. Overall, the proposed methods demonstrated significant improvements over the traditional baselines, showcasing their efficacy in enhancing object detection and segmentation accuracy with lower localization errors.

Table – 2 Comparison results of baseline models with the proposed components provided

Methods	Accuracy	Precision	Recall	F1-Score	IoU	RMSE
Fast R-CNN	0.8300	0.8200	0.8500	0.8350	0.8100	1.5000
YOLOv3	0.8600	0.8500	0.8800	0.8650	0.8400	1.2000
YOLOv8n	0.9100	0.9000	0.9300	0.9150	0.9000	0.9000
ASCNN	0.9828	0.9805	0.9828	0.9796	0.9828	0.6773
HOFSG	0.9751	0.9691	0.9751	0.9641	0.9751	0.7683
SEAT	0.9828	0.9789	0.9828	0.9799	0.9828	0.6855

4.1.1 Detection and Segmentation Performance

The evaluation of the proposed model was done using KITTI dataset which includes classes such as Car, Van, Truck, Pedestrian, Person_sitting, Cyclist, Tram, Misc, and DontCare.



10.48047/jocaaa.2024.33.07.49

Fig. 5: Confusion matrix

Confusion Matrix: Figure 5 provides the model's predictions across all classes. The model shows strong performance for the `Car` class, with a high true positive rate, while classes like `Person_sitting` and `Cyclist` exhibit lower performance, reflecting the inherent difficulty in distinguishing between these classes.

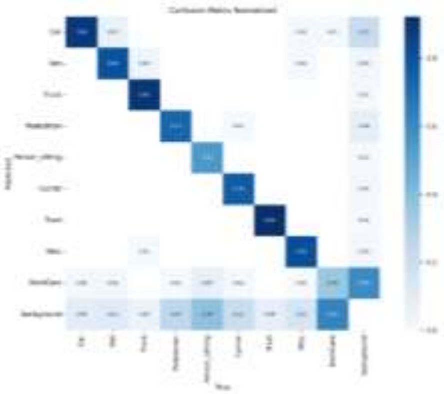


Fig. 6: Normalized confusion matrix

Normalized Confusion Matrix: Figure 6 indicates that the model has a high accuracy for the `Car` class, with an accuracy rate of 0.89. The accuracy for other classes varies, with `Van` at 0.80 and `Truck` at 0.90. However, the `Person_sitting` class shows a lower accuracy of 0.55, indicating potential challenges in identifying smaller or less distinct objects.

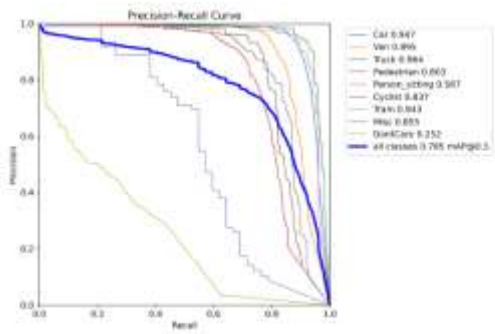


Fig. 7: Precision-Recall curve

Precision-Recall Curve: Figure 7 highlights that the model achieves a mean Average Precision (mAP) of 0.785 at an IoU threshold of 0.5. The `Truck` class achieved the highest precision at 0.964, followed by `Car` at 0.947, indicating robust detection capabilities for these classes. The `DontCare` class, as expected, has the lowest precision at 0.252 due to its nature as a background or non-object class.

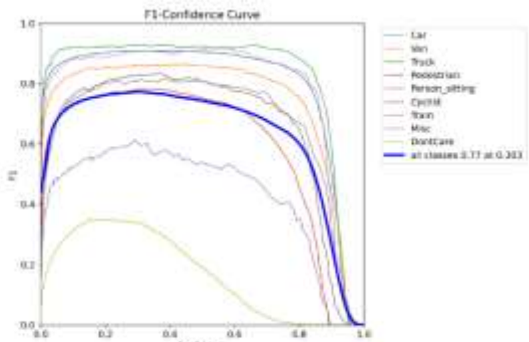


Fig. 8: F1-Confidence curve

10.48047/jocaaa.2024.33.07.49

F1-Confidence Curve: Figure 8 demonstrates the balance between precision and recall as the confidence threshold varies. The model achieves the highest F1 score for the `Car` class at around 0.9 confidence, while the `Person_sitting` and `DontCare` classes exhibit lower F1 scores across most confidence thresholds.

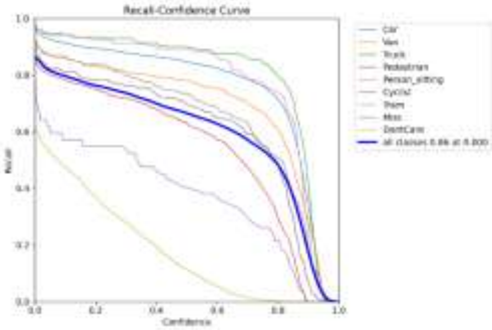


Fig. 9: Recall confidence curve

Recall-Confidence Curve: Figure 9 shows how the recall metric changes with varying confidence levels. Higher recall is observed for classes such as `Car` and `Van`, with recall values exceeding 0.9 at lower confidence levels.

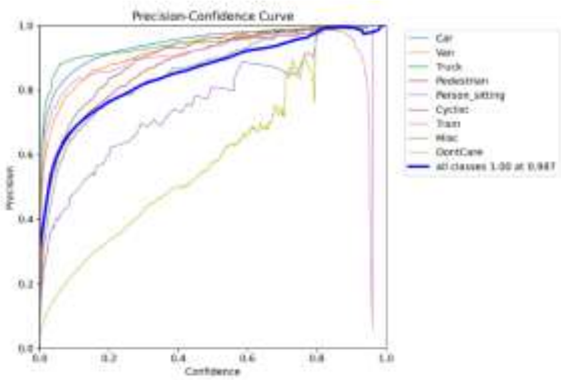


Fig. 10: Precision confidence curve

Precision-Confidence Curve: Figure 10 illustrates that the proposed model maintains high precision across most classes as confidence increases. For all classes combined, precision reaches 1.00 at a confidence level of 0.987, indicating the model's strong ability to accurately identify objects at higher confidence thresholds. Some classes, like "DontCare" and "Tram," show lower precision at varying confidence levels, suggesting room for improvement in those specific categories. Overall, the model demonstrates robust performance in precision across the majority of object classes.

4.1.2 Training and Validation Analysis

The training and validation metrics after 20 epochs also shows high accuracy of the proposed model and its robustness. The loss of training and validation set are decreasing over the epochs and the loss of the validation set is lesser than the training loss, it implies that the learning has occurred without much overfitting. All of the measures are on the rise; similarly to the training measures, the validation measures tend to be slightly higher on average. This is seen by the high correlation achieved between the training and validation data which shows that the model has a high chance of generalizing well to unseen data. The rapid fluctuation of precision and recall values signifies the program maintains a good balance between the identification of true positive and the minimization of false positive and false negatives. Altogether, these results support the effectiveness and general excellence of the proposed model for object detection and segmentation.

10.48047/jocaaa.2024.33.07.49

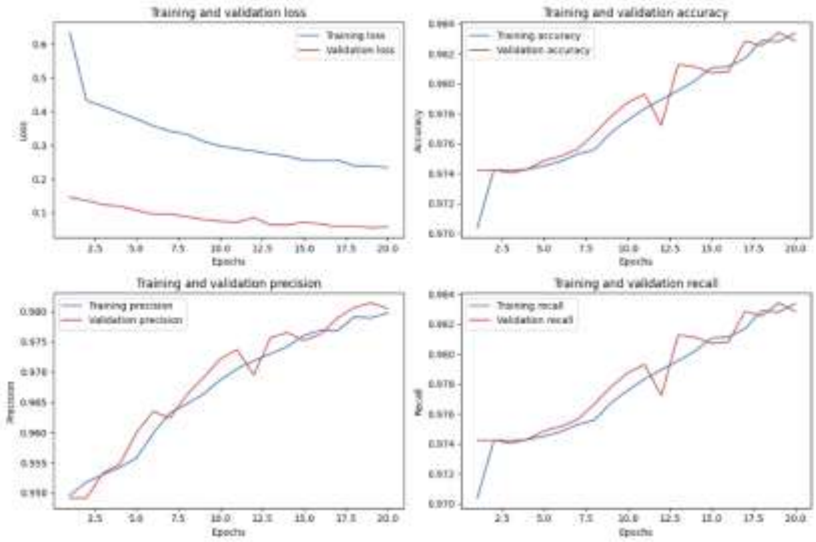


Fig. 11: Results of training and validation analysis

İn the figure 11 train/box_loss starts at approximately 0.70 and gradually decreases, indicating improved bounding box regression performance, while the val/box_loss shows a similar downward trend, validating the model's capability in generalization. The train/cls_loss and val/cls_loss both exhibit a decrease over time, signifying enhanced classification accuracy as the model learns better object distinctions. The metrics of precision and recall for the model exhibit fluctuations but generally show an upward trend, particularly in recall, which improves from around 0.68 to 0.74. This indicates the proposed work has the ability to correctly identify objects as training progresses. The precision metric also improves, although with more variation, demonstrating the model's capacity to reduce false positives over time.

İn the figure 12 The mean Average Precision at 50% IoU (mAP50) steadily rises from approximately 0.76 to 0.79, while the mAP across IoU thresholds (mAP50-95) shows a more significant improvement from 0.52 to 0.55. From these results, it is evident that there is a marked improvement in the model's general ability to detect objects especially in difficult conditions. Training and validation loss reduction and precision, recall, and mAP augmentation are introduced to show that the proposed model is capable of effective training along with robust performance enhancement with successive epochs of training.

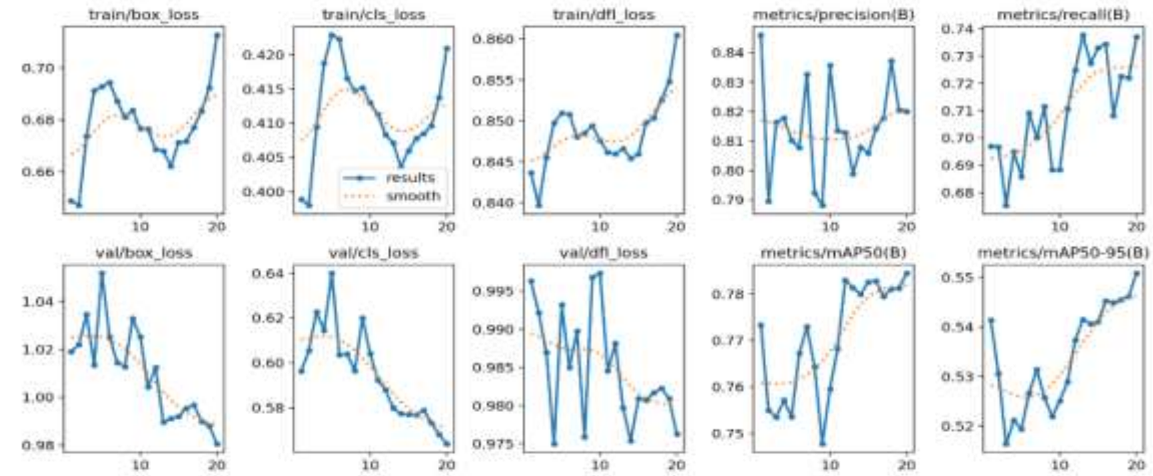


Fig. 12: Training and Validation Loss, Precision, Recall, and mAP Metrics over 20 Epochs

IoU and RMSE: Table 3 represents the proposed work's IoU and RMSE results. The IoU values were notably high for dominant classes like `Car` and `Truck`, while RMSE values indicate that the model maintains low prediction errors across the dataset.

10.48047/jocaaa.2024.33.07.49

Table – 3: Comparative Results of IoU and RMSE

Epochs	IoU		RMSE	
	Training data	Validation data	Training data	Validation data
1	0.970375	0.97423	1.577196	0.785243
2	0.974232	0.97423	0.785234	0.785243
3	0.974225	0.974047	0.785308	0.793759
4	0.974279	0.974287	0.783825	0.784916
5	0.974514	0.974864	0.773598	0.765339
6	0.974813	0.975155	0.761758	0.771411
7	0.97529	0.975616	0.750374	0.763906
8	0.975586	0.976654	0.751414	0.743789
9	0.976747	0.977811	0.730122	0.720167
10	0.977576	0.978746	0.718667	0.722321
11	0.978329	0.979302	0.709846	0.70448
12	0.978942	0.977219	0.706029	0.756239
13	0.979542	0.981268	0.695757	0.69245
14	0.980181	0.981122	0.69349	0.689568
15	0.981069	0.980718	0.681789	0.695471
16	0.981133	0.98079	0.682678	0.69596
17	0.981698	0.982845	0.680496	0.669852
18	0.982891	0.982536	0.670204	0.678829
19	0.982805	0.98342	0.670553	0.682241
20	0.983362	0.982846	0.666956	0.677277

4.2 Tracking Results

The table 4 compares the performance of different tracking techniques across three object classes: Car, Truck, and Pedestrian. The metrics used are MOTA, MOTP, and IDF1. The proposed work consistently outperforms the baseline methods, achieving the highest scores in all metrics across all classes. Specifically, the proposed method shows a significant improvement in MOTA, with a score of 92.3% for cars, 86.4% for trucks, and 81.7% for pedestrians. This indicates a more accurate tracking performance than SORT, DeepSORT, and OC-SORT. Similarly, the MOTP and IDF1 scores are superior in the proposed work, suggesting that it tracks objects with higher precision and maintains better identity consistency throughout the tracking process.

Table – 4: Comparative results of object tracking

Techniques	Car			Truck			Pedestrian		
	MOTA	MOTP	IDF1	MOTA	MOTP	IDF1	MOTA	MOTP	IDF1
SORT	82.5	84.7	80.2	75.3	78.4	72.8	69.8	72.1	66.4
DeepSORT	86.1	88.0	83.7	79.6	82.2	77.1	73.4	75.7	70.2
OC-SORT	88.7	89.9	86.1	81.8	84.1	79.5	76.2	78.4	73.0
Proposed Work	92.3	93.5	90.4	86.4	88.9	84.2	81.7	83.8	78.6

Table-5 shows that the proposed model outperforms recent studies in mAP and object detection. The achieved mAP of 79% at baseline shows that the proposed model with the sophisticated tracking and adaptable detection algorithms works way better than other models that fail to effectively maintain high accuracy while scaling an object and or changing the environment. [39] use synthetic data as a training basis, the moderate mAP is obtained for one or another detection model (Faster RCNN, YOLOF, DETR) and it does not reach the fine-tuned accuracy at real-world conditions. [37] obtain high object detection to an accuracy of 94.74% after a hundred training epochs are completed 300 epochs to be precise and perfect, the proposed model hits an exquisite accuracy of

98.3% within just twenty epochs, making it quite efficient and robust in the quick attainment of high precision. Finally, the proposed model yields 0.325 of mAP for large objects with performance that is lower than the [40] model having 0.384 mAP, however, the overall performance of the object detection proposed model is suitable for a range of object sizes and can be implemented in real time activity compared to the [40] model.

Table – 5: Comparative results with recent studies

Metrics	Proposed Model	[39]	[37]	[40]
mAP – Baseline	79%	69.45% (Faster RCNN), 60.06% (YOLOF), 61.91% (DETR)	73%	mAP ~0.384 (peak mAP for large objects after tuning)
Object Detection Accuracy	98.3% (20 Epochs)	Limited to synthetic scenes, no specific precision for cars in natural settings	94.74% (300 Epochs)	92.3%

4.3 Visual Results

The visual results (Figure 13) illustrate the detection and segmentation of objects in various scenarios from the KITTI dataset. İt shows the proposed work's ability to correctly identify and label different objects, such as cars, cyclists, and miscellaneous objects, even in complex environments. The labels also include confidence scores, indicating the system's certainty about its predictions. The "DontCare" label is used for objects the system recognizes as irrelevant for the tracking or classification task, showcasing its ability to filter out unnecessary information. The diversity of scenes, ranging from urban streets to more open roads, highlights the robustness of the proposed approach across different contexts. The consistent accuracy in detecting and classifying objects, despite variations in lighting, object size, and occlusions, demonstrates the system's effectiveness and reliability. This visual evidence supports the previously discussed quantitative results, confirming that the proposed system performs well in real-world scenarios, maintaining high accuracy and precision.

Fig. 13: Visual Examples of Object Detection and Classification Results Across Diverse Road Scenarios

5. Conclusion

The proposed system addresses several critical challenges, including occlusions, rapid object movements, and varying object scales. Key components of the system, namely the ASCNN, HOF-SG, and SEAT, were developed and optimized to enhance detection accuracy, motion analysis, and tracking consistency. The integration of semantic information into the tracking process, as exemplified by SEAT, proved particularly effective in handling complex scenarios involving occlusions and rapid movements. Additionally, using ASCNN and HOF-SG contributed significantly to the system's ability to adapt to varying object scales and accurately track dynamic objects.

Conflict of Interest

The authors declare that there are no conflicts of interest related to the publication of this paper. The research presented in this manuscript was conducted independently and without any financial or personal relationships that could be perceived as potential sources of bias.

10.48047/jocaaa.2024.33.07.49

Data Availability

The data supporting the findings of this study are not publicly available due to privacy and ethical considerations.

References

[1]. S. K. Pal, A. Pramanik, J. Maiti, and P. Mitra, "Deep learning in multi-object detection and tracking: state of the art," Appl. Intell., vol. 51, no. 9, pp. 6400–6429, Sep. 2021, doi: 10.1007/S10489-021-02293-7/FIGURES/4.

[2]. J. Zhao et al., "Autonomous driving system: A comprehensive survey," Expert Syst. Appl., vol. 242, p. 122836, May 2024, doi: 10.1016/J.ESWA.2023.122836.

[3]. V. Malligere Shivanna and J. I. Guo, "Object Detection, Recognition, and Tracking Algorithms for ADASs—A Study on Recent Trends," Sens. 2024 Vol 24 Page 249, vol. 24, no. 1, p. 249, Dec. 2023, doi: 10.3390/S24010249.

[4]. M. Runz, M. Buffier, and L. Agapito, "MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects," Proc. 2018 IEEE Int. Symp. Mix. Augment. Real. ISMAR 2018, pp. 10–20, Jul. 2018, doi: 10.1109/ISMAR.2018.00024.

[5]. L. Schmid, O. Andersson, A. Sulser, P. Pfreundschuh, and R. Siegwart, "Dynablox: Real-Time Detection of Diverse Dynamic Objects in Complex Environments," IEEE Robot. Autom. Lett., vol. 8, no. 10, pp. 6259–6266, Oct. 2023, doi: 10.1109/LRA.2023.3305239.

[6]. D. Meimetis, I. Daramouskas, I. Perikos, and I. Hatzilygeroudis, "Real-time multiple object tracking using deep learning methods," Neural Comput. Appl., vol. 35, no. 1, pp. 89–118, Jan. 2023, doi: 10.1007/S00521-021-06391-Y/FIGURES/50.

[7]. R. Kaur and S. Singh, "A comprehensive review of object detection with deep learning," Digit. Signal Process., vol. 132, p. 103812, Jan. 2023, doi: 10.1016/J.DSP.2022.103812.

[8]. S. P. Yadav, M. Jindal, P. Rani, V. H. C. de Albuquerque, C. dos Santos Nascimento, and M. Kumar, "An improved deep learning-based optimal object detection system from images," Multimed. Tools Appl., vol. 83, no. 10, pp. 30045–30072, Mar. 2024, doi: 10.1007/S11042-023-16736-5/FIGURES/19.

[9]. M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection," Mach. 2023 Vol 11 Page 677, vol. 11, no. 7, p. 677, Jun. 2023, doi: 10.3390/MACHINES11070677.

[10]. M. Sohan, T. Sai Ram, and Ch. V. Rami Reddy, "A Review on YOLOv8 and Its Advancements," pp. 529–545, 2024, doi: 10.1007/978-981-99-7962-2_39.

[11]. G. P. M. Bhat and D. N. G. Cholli, "A Comprehensive Survey on Computer-Vision Object Detection, Segmentation, Tracking, and Feature Extraction," Int. J. Adv. Sci. Technol., vol. 28, no. 16, Art. no. 16, Dec. 2019.

[12]. R. Pereira, G. Carvalho, L. Garrote, and U. J. Nunes, "Sort and Deep-SORT Based Multi-Object Tracking for Mobile Robotics: Evaluation with New Data Association Metrics," Appl. Sci. 2022 Vol 12 Page 1319, vol. 12, no. 3, p. 1319, Jan. 2022, doi: 10.3390/APP12031319.

[13]. X. Xiao and X. Feng, "Multi-Object Pedestrian Tracking Using Improved YOLOv8 and OC-SORT," Sens. 2023 Vol 23 Page 8439, vol. 23, no. 20, p. 8439, Oct. 2023, doi: 10.3390/S23208439.

[14]. J. S. D. R. G. A. F. Redmon, "(YOLO) You Only Look Once," Cvpr, vol. 2016-December, pp. 779–788, Dec. 2016, doi: 10.1109/CVPR.2016.91.

[15]. Y. Zhang, T. Wang, and X. Zhang, "MOTRv2: Bootstrapping End-to-End Multi-Object Tracking by Pretrained Object Detectors," 2023 IEEECVF Conf. Comput. Vis. Pattern Recognit. CVPR, pp. 22056–22065, Jun. 2023, doi: 10.1109/CVPR52729.2023.02112.

[16]. S. B. Neamah and A. A. Karim, "Real-time Traffic Monitoring System Based on Deep Learning and YOLOv8," ARO- Sci. J. KOYA Univ., vol. 11, no. 2, pp. 137–150, Nov. 2023, doi: 10.14500/aro.11327.

[17]. M. Safaldin, N. Zaghden, and M. Mejdoub, "An Improved YOLOv8 to Detect Moving Objects," IEEE Access, vol. 12, pp. 59782–59806, 2024, doi: 10.1109/ACCESS.2024.3393835.

[18]. N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," Proc. - Int. Conf. Image Process. ICIP, vol. 2017-September, pp. 3645–3649, Jul. 2017, doi: 10.1109/ICIP.2017.8296962.

10.48047/jocaaa.2024.33.07.49

A. Yadav, P. K. Chaturvedi, and S. Rani, "Object Detection and Tracking using YOLOv8 and DeepSORT," Adv. Commun. Syst., pp. 81–90, 2024, doi: 10.56155/978-81-955020-7-3-7.

[19]. L. Shen, B. Lang, and Z. Song, "DS-YOLOv8-Based Object Detection Method for Remote Sensing Images," IEEE Access, vol. 11, pp. 125122–125137, 2023, doi: 10.1109/ACCESS.2023.3330844.

[20]. Z. Huangfu and S. Li, "Lightweight You Only Look Once v8: An Upgraded You Only Look Once v8 Algorithm for Small Object Identification in Unmanned Aerial Vehicle Images," Appl. Sci. 2023 Vol 13 Page 12369, vol. 13, no. 22, p. 12369, Nov. 2023, doi: 10.3390/APP132212369.

[21]. G. Zidani, D. Djarah, A. Benmakhlouf, and L. Khettache, "OPTIMIZING PEDESTRIAN TRACKING FOR ROBUST PERCEPTION WITH YOLOV8 AND DEEPSORT," Appl. Comput. Sci., vol. 20, no. 1, pp. 72–84, 2024, doi: 10.35784/ACS-2024-05.

[22]. N. Nyi Myo, A. Boonkong, K. Khampitak, and D. Hormdee, "Real-Time Surgical Instrument Segmentation Analysis Using YOLOv8 With ByteTrack for Laparoscopic Surgery," IEEE Access, vol. 12, pp. 83091–83103, 2024, doi: 10.1109/ACCESS.2024.3412780.

[23]. F. Hermens, "Automatic object detection for behavioural research using YOLOv8," Behav. Res. Methods, vol. 56, no. 7, pp. 7307–7330, May 2024, doi: 10.3758/S13428-024-02420-5/FIGURES/22.

A. Sahafi, A. Koulaouzidis, and M. Lalinia, "Polypoid Lesion Segmentation Using YOLO-V8 Network in Wireless Video Capsule Endoscopy Images," Diagn. 2024 Vol 14 Page 474, vol. 14, no. 5, p. 474, Feb. 2024, doi: 10.3390/DIAGNOSTICS14050474.

[24]. S. Yue, Z. Wang, and X. Zhang, "DSOMF: A Dynamic Environment Simultaneous Localization and Mapping Technique Based on Machine Learning," Sens. 2024 Vol 24 Page 3063, vol. 24, no. 10, p. 3063, May 2024, doi: 10.3390/S24103063.

[25]. S. Hossain and D. J. Lee, "Deep Learning-Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via a Flying Robot with GPU-Based Embedded Devices," Sens. 2019 Vol 19 Page 3371, vol. 19, no. 15, p. 3371, Jul. 2019, doi: 10.3390/S19153371.

[26]. K. An et al., "An Advanced Approach to Object Detection and Tracking in Robotics and Autonomous Vehicles Using YOLOv8 and LiDAR Data Fusion," Electron. 2024 Vol 13 Page 2250, vol. 13, no. 12, p. 2250, Jun. 2024, doi: 10.3390/ELECTRONICS13122250.

[27]. J. Gómez, O. Aycard, and J. Baber, "Efficient Detection and Tracking of Human Using 3D LiDAR Sensor," Sens. 2023 Vol 23 Page 4720, vol. 23, no. 10, p. 4720, May 2023, doi: 10.3390/S23104720.

[28]. Y. Wang, X. Liu, M. Zhao, and X. Xu, "VIS-SLAM: A Real-Time Dynamic SLAM Algorithm Based on the Fusion of Visual, Inertial, and Semantic Information," ISPRS Int. J. Geo-Inf. 2024 Vol 13 Page 163, vol. 13, no. 5, p. 163, May 2024, doi: 10.3390/IJGI13050163.

[29]. T. Fischer et al., "QDTrack: Quasi-Dense Similarity Learning for Appearance-Only Multiple Object Tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 12, pp. 15380–15393, Oct. 2022, doi: 10.1109/TPAMI.2023.3301975.

[30]. B. Drayer and T. Brox, "Object Detection, Tracking, and Motion Segmentation for Object-level Video Segmentation," arXiv.org, 2016.