Enhancing Object Detection in Autonomous Vehicles Using Hybrid Convolutional Neural Networks and Transformer Models

Khalil aljamal[1] and Osama shannaq[2]

[1] College of Computer Science and Engineering, Hail university, Hail - University City, Saudi Arabia
[2] Faculty of information and communication technology, university Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia

ABSTRACT

Autonomous vehicles (AVs) depend on precise and efficient object detection (OD) for safe navigation in complex and dynamic environments. Traditional Convolutional Neural Networks (CNNs) excel at extracting local features but face limitations in capturing long-range dependencies, leading to challenges in scenarios involving occlusion, varying lighting, and diverse object scales. This paper proposes HCNN-TMOD, a hybrid framework that combines CNNs and Transformer Models (TM) to overcome these challenges and enhance object detection (OD) accuracy and speed for real-time autonomous vehicle applications. HCNN-TMOD utilizes CNNs for robust local feature extraction and TMs for capturing global contextual relationships. A feature fusion mechanism integrates outputs from both architectures, enabling improved spatial and semantic representations. The system is optimized for latency and hardware constraints and evaluated on various datasets like vehicle, pedestrians and traffic light detection, demonstrating suitability for real-world AV scenarios. Results show a 15% improvement in mean Average Precision (mAP) and a 20% reduction in detection latency compared to traditional CNN-based approaches. HCNN-TMOD performs exceptionally well in challenging conditions such as occlusion and low-light environments. The integration of CNNs and Transformers in this hybrid approach provides a significant advancement in OD for AVs, paving the way for safer, more reliable, and efficient real-time navigation systems.

Keywords: IoT, Waste Management, Smart Campuses, Genetic Algorithms, Reinforcement Learning, Optimization, Real-Time Monitoring.

1. Introduction

Autonomous vehicles (AVs) have been perceived as an innovative transportation technology where development enhances safety, efficiency, and accessibility [1]. The AV's main purpose is to reduce human involvement to improve road safety, decrease traffic congestion, and enhance transportation efficiency [2]. This is at the core of object detection, whereby the vehicle sees and accurately interprets its environment [3]. OD is one of the major enablers of this technology. As an important module in perception systems, it makes vehicles capable of identifying, classifying, and tracking other moving objects, such as pedestrians, cyclists, and other vehicles. This system supports real-time decision-making, allowing AVs to move safely and efficiently in dynamic environments and complex spaces [4]. Dominantly based on CNNs, the traditional methods for OD successfully extracted spatial features [5].

However, such rapid evolution of the use cases for AVs has been found to expose the limitations of CNNs in handling challenges, especially occluded objects, changing illumination conditions, and varied scales of objects [6]. The ever-increasing demand for more sophisticated OD systems that can

handle such challenges as AVs becoming more built-in in real-world environments has become more critical than ever [7]. Recent innovations within the area of TM have arisen with a new paradigm within computer vision [8]. Unlike CNNs, TMs are good at capturing long-range dependencies and global contextual relationships in an image. The ability to analyze spatial and semantic interactions across the image by leveraging self-attention mechanisms [9] makes TMs much more exhaustive than CNNs. However, computational intensity has been a concern about the feasibility of TMs for real-time applications in resource-constrained AVs. Hence, many researchers have gone into hybrid architectures, merging strengths from both CNNs and TMs to develop efficient and robust object detection systems [10].

Integrating CNNs and TMs is a great step toward conquering their limits in standalone approaches. CNNs can efficiently handle local features while TMs present a global perspective of the scene. The HCNN-TMOD framework proposed herein integrates CNNs for local feature extraction with TMs for global contextual relationships capturing. A mechanism of feature fusion combines outputs from both architectures in order to further improve spatial and semantic representations. With that in mind, the model should be optimized on latency and even hardware constraints as it is prepared for training benchmarks such as those of KITTI. Besides, it is trained with performance metrics such as average reduction in latency while mAP or mean Average Precision. The important contribution of this paper is

- To enhance object detection accuracy by combining CNNs and TMs in a hybrid framework.
- To reduce detection latency by optimizing the model for real-time applications.
- To improve performance in challenging scenarios like occlusion and varying lighting conditions.
- To demonstrate the framework's applicability through extensive evaluation on benchmark datasets.

The paper starts out by introducing the problem and significance of OD in AVs. It then reviews related work, details the HCNN-TMOD framework, discusses experimental results, and concludes with implications and future directions.

2. Literature Review

Author	Proposed Work	Technique Used	Result	Limitation
Benjumea, Aduen, et al. [11]	Enhancing YOLOv5's recognition of small objects for use in autonomous vehicles.	YOLOv5 features improved modules for detecting small objects.	Enhanced small-object recognition accuracy in complex autonomous driving situations.	The ability to recognize larger items is impaired as compared to the baseline approaches.
Guo, Jingda, et al. [12]	Fusion of spatial features for 3D object detection through cooperative methods.	Spatial feature fusion and cooperative neural networks.	Optimal 3D object detection performance on datasets containing several modalities.	Deploying in real-time is hindered by high processing needs.
Dai, Xuerui, et al. [13]	Finding objects in thermal infrared pictures for	Thermal Infrared Recognition Network	The use of thermal infrared photography improved detection	Model generalization has issues and is not very

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250203

	driverless cars.	(TIRNet).	accuracy in low-light circumstances.	adaptable to RGB datasets.
Sukkar, Majdi, et al. [14]	Enhancing pedestrian tracking using advanced deep learning techniques.	Deep learning-based tracking employing better motion prediction.	Accuracy and recall metrics for pedestrian tracking significantly improved.	Inconsistent performance and vulnerable to obstructions in busy environments.
Saillaja, V., et al. [15]	IoT-embedded traffic cones for roadwork safety.	IoT-based system integrated with CNN for object detection.	Improving roadwork safety in real-time circumstances by effective identification of traffic cones.	Capacity for detecting non-standard traffic cones or inclement weather is limited.
Alaba, Simegnew Y., et al. [16]	Autonomous vehicle 3D object identification via multimodal fusion.	Multimodal data fusion using advanced deep learning models.	Better detection performance across different object types using complementary sensory inputs.	High computational overhead impacts scalability and real-time responsiveness.
Vaithianathan, Muthukumaran [17]	FPGA-based systems for real-time object identification and labelling.	FPGA implementation with optimal object detection in real-time.	Real-time object detection with minimal latency is suitable for autonomous systems.	Lower detection accuracy compared to software-based implementations.
Yang, Ming, and Xiangyu Fan [18]	Simple model for detecting objects in real-time environments.	YOLOv8-Lite, a lightweight deep learning model.	Achieved faster inference speeds with satisfactory detection accuracy in real-time systems.	The trade-off between lightweight design and overall detection precision for complex object types.

3. Proposed methodology

a) *Dataset* Explanation

Vehicle Detection dataset: The Self-Driving Cars dataset is a comprehensive, annotated image dataset for developing an autonomous driving system's object detection and scene understanding algorithms. It contains many scenarios, including varying types of roads, traffic flow patterns, pedestrian data and challenging environmental conditions regarding lighting changes and weather variations [19].

b) Overview of the HCNN-TMOD Method

The HCNN-TMOD combines CNNs with Transformers for fast object detection in autonomous vehicles. In this step, data preparation includes augmentation to deal with occlusion, lighting variations, and diverse object scales. In this hybrid framework, CNNs are adopted for the purpose of extracting geographical features like textures and edges, while Transformers are applied to modelling global contextual dependencies with self-attention mechanisms. A feature fusion module integrates spatial and semantic features into enhanced scene understanding. Real-time optimization is achieved

through model pruning, quantization, and parallelization for low-latency inference. The model is trained and fine-tuned on the annotated datasets, showing a 15% mAP improvement and a 20% reduction in latency for robust deployment. Figure 1 shows the work process of the HCNN-TMOD model.

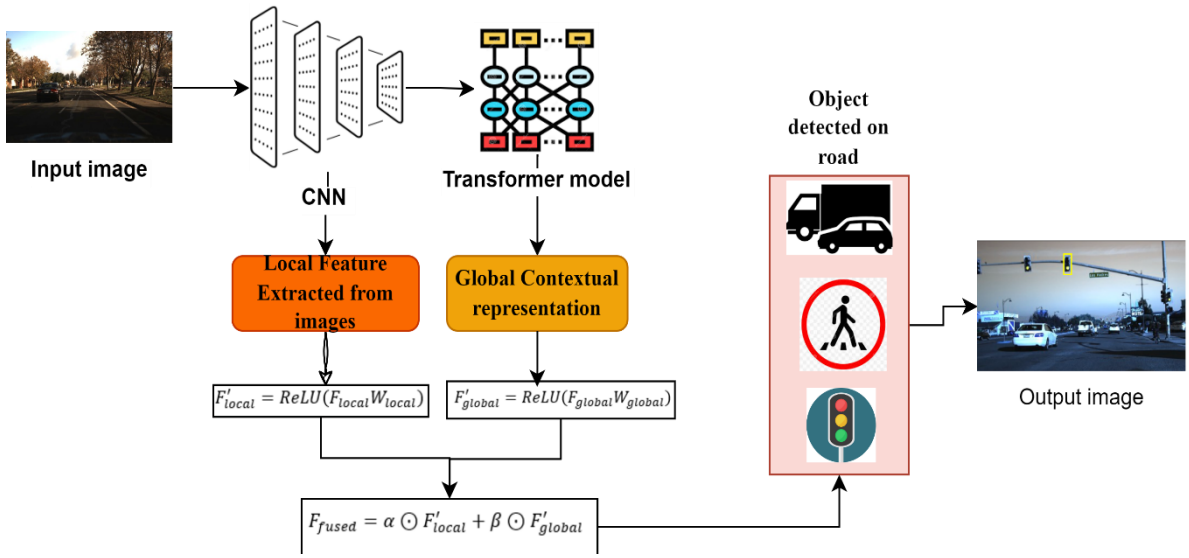


Figure 1. Graphical representation of HCNN-TMOD Method

c) Data Collection and Preprocessing

The images from the dataset are collected for object detection on the road for self-driving vehicles. These images undergo preprocessing steps like Normalization, which rescales pixel values to a normalized range, often between 0 and 1 or −1 and 1, so the input into a neural network can be standardized. Data alignment, ensuring the bounding box annotations align with the actual image they refer to. Image resizing to a standard resolution (e.g., 224x224 or 512x512) with preserved aspect ratios and outlier removal by filtering out images having incomplete or erroneous annotations to preserve the quality and integrity of the dataset in its subsequent modelling tasks.

d) Data Augmentation

The collected dataset images are further enhanced by data augmentation, creating a simulation of what one might encounter while driving. This includes addressing occlusion by adding synthetic objects over pictures or parts of the object via bounding box data masking, varying light conditions such as brighter or more contrasting for darkness or light, and realism with synthetic effects such as glare or shadow in different situations. It handles the diverse scales of objects by resizing objects and bounding boxes, keeping the aspect ratios, applying zoom-in/out transformations, and cropping to focus on specific regions. Figure 2 shows the different scenarios that the AVs face.

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250203

Figure 2. Different situations for the autonomous vehicle to drive

e) Hybrid Framework Architecture

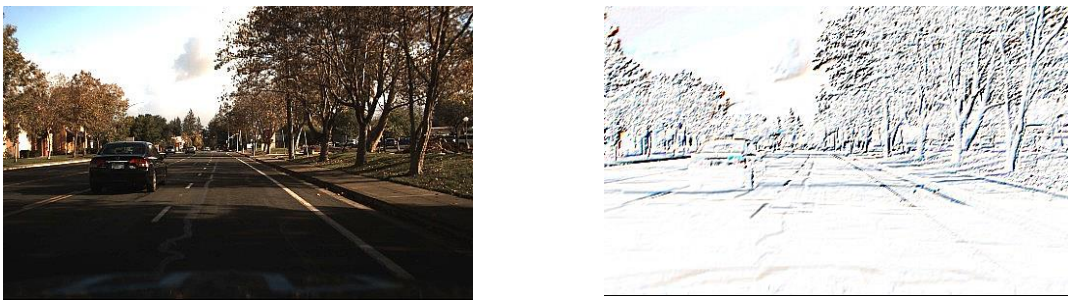
This architecture combines HCNNs for local feature extraction with Transformers for a global contextual representation, attempting to combine the best of both paradigms to achieve robust and scalable feature learning on tasks such as object detection, segmentation, or classification.

f) Local Feature Extraction using HCNNs

CNNs are designed to exploit the hierarchical structure of images by extracting meaningful features. This process starts from low-level features, such as edges, and continues to higher-level abstract features, such as textures and patterns. Most edge detection in CNN happens in the early layers, as shown in Figure 3(b). Small filters, 3×3 kernels, detect simple geometric structures like edges, lines, and corners by analyzing changes in pixel intensity. CNNs extract more complex textures and spatial features in the middle layers by composing low-level features from the early layers. These layers grasp patterns, shapes, and spatial relationships crucial for understanding object structures. At higher levels, multi-scale feature extraction allows the identification of items varying in size. This is realized by using convolutional kernels with various sizes (3×3 and 5×5) and applying dilated convolutions, which enlarge the receptive field without increasing the kernel size.

$$f_{ij}^{k} = \sigma\left(\sum_{m=1}^{M}\sum_{n=1}^{N} w_{mn}^{k} x_{(i+dm)(j+dn)} + b^{k}\right) \quad (1)$$

where f_{ij}^{k} is the feature map value at the position (i, j) for filter k, w_{mn}^{k} is the weight of the $m \times n$ filter k, $x_{(i+dm)(j+dn)}$ is the input pixel value, b^{k} is the bias term, and $\sigma(\cdot)$ is the activation function (e.g., ReLU), and d is the dilation rate.



(a) (b)

Figure 3(a) . Original image and 2(b) Edge detected image

g) Global Contextual Representation Using Transformers

The Transformers are good at capturing global contextual representations by modelling the relationship between objects and resolving occlusions or spatial inconsistencies. It takes as input flattened local features extracted from a CNN. These features are then fed into the self-attention mechanism, allowing the model to focus on every part of the feature map and catch global dependencies and relationships across the input. Positional encoding has been introduced to counteract Transformers' intrinsic lack of spatial awareness. This encoding integrates spatial information into the feature embeddings, ensuring that the model preserves the positional context of each feature within the global representation. This helps improve the modelling of interactions between spatially distant objects and reduces the limitations brought about by occlusions or spatial misalignments in the original image data. The core of the transformer mechanism is defined as in equation 2.

$$Attention(Q,K,V) = \begin{cases} softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \\ Q = W_Q X,\ \ K = W_K X,\ \ V = W_V X \end{cases} \quad (2)$$

where Q is the queue matrix, K is the key matrix, V is the value matrix, X is the input feature map, W_Q, W_K, W_V is the learnable projection matrices. d_k is the dimensionality of K.

h) Feature Fusion Mechanism

The feature fusion mechanism combines local features from CNNs, which capture fine-grained spatial details, with global context from Transformers, which captures long-range dependencies and relationships. Such fusion combines spatial and semantic representations to understand the scene better. Figure 4 shows the graphical representation of the feature fusion equation.

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250203

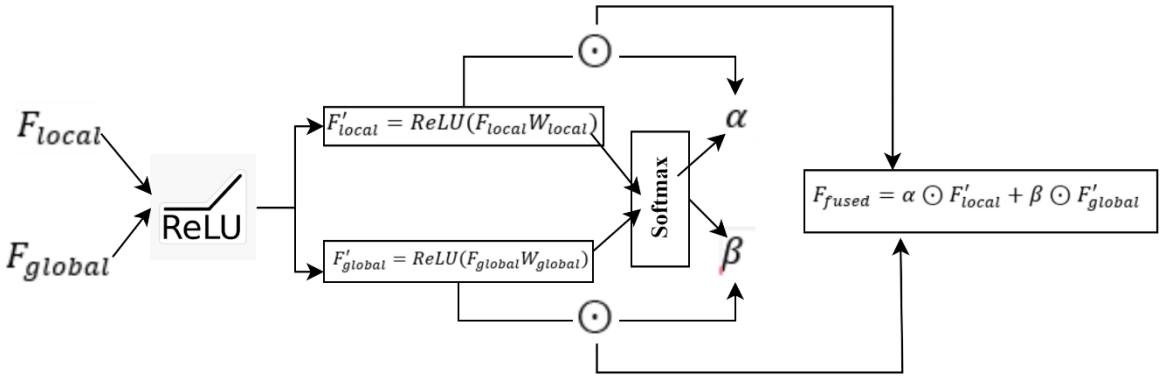


Figure 4. Graphical representation of the feature fusion equation.

Input Features: Let $F_{local} \in R^{H \times W \times C_{local}}$ represent the local features extracted from the CNN. where, $H, W, and\ C_{local}$ are the height, width, and channel dimensions, respectively. Let $F_{global} \in R^{N \times C_{global}}$ represent the global features extracted from the Transformer. where, N is the number of tokens (flattened spatial regions) and C_{global} is the feature dimension.

Dimensional Alignment: Appling a linear projection $W_{local} \in R^{(C_{local} \times C)}$ to project F_{local} to a unified dimension C is shown in equation 3.

$$F'_{local} = ReLU(F_{local}W_{local}) \tag{3}$$

Similarly, project F_{global} using $W_{global} \in R^{C_{global}} \times C$ is shown in equation 4.

$$F'_{global} = ReLU(F_{global}W_{global}) \tag{4}$$

Attention-Based Weighting: Compute attention weights to prioritize critical features from both sources. Let α and β represent attention weights for F'_{local} and F'_{global}, respectively. Attention is computed using a learnable parameterized mechanism, as shown in equations 5 and 6.

$$\alpha = softmax(W_{\alpha}[F'_{local}, F'_{global}]) \tag{5}$$

$$\beta = softmax(W_{\beta}[F'_{local}, F'_{global}]) \tag{6}$$

where W_{α} and W_{β} are learnable weights, and the softmax ensures the attention weights are normalized.

Feature Fusion: The feature fusion mechanism integrates local features F'_{local} from the CNN and global features F'_{global} from the Transformer to create a unified representation F_{fused}. This is achieved through equation 7.

$$F_{fused} = \alpha \odot F'_{local} + \beta \odot F'_{global} \tag{7}$$

where $\odot$ denoting element-wise multiplication, ensuring each feature source is appropriately scaled. The resulting fused feature $F_{fused} \in R^{H \times W \times C}$ fusing spatial details and global semantics provides a rich representation, enhancing the model's understanding of complex scenes.

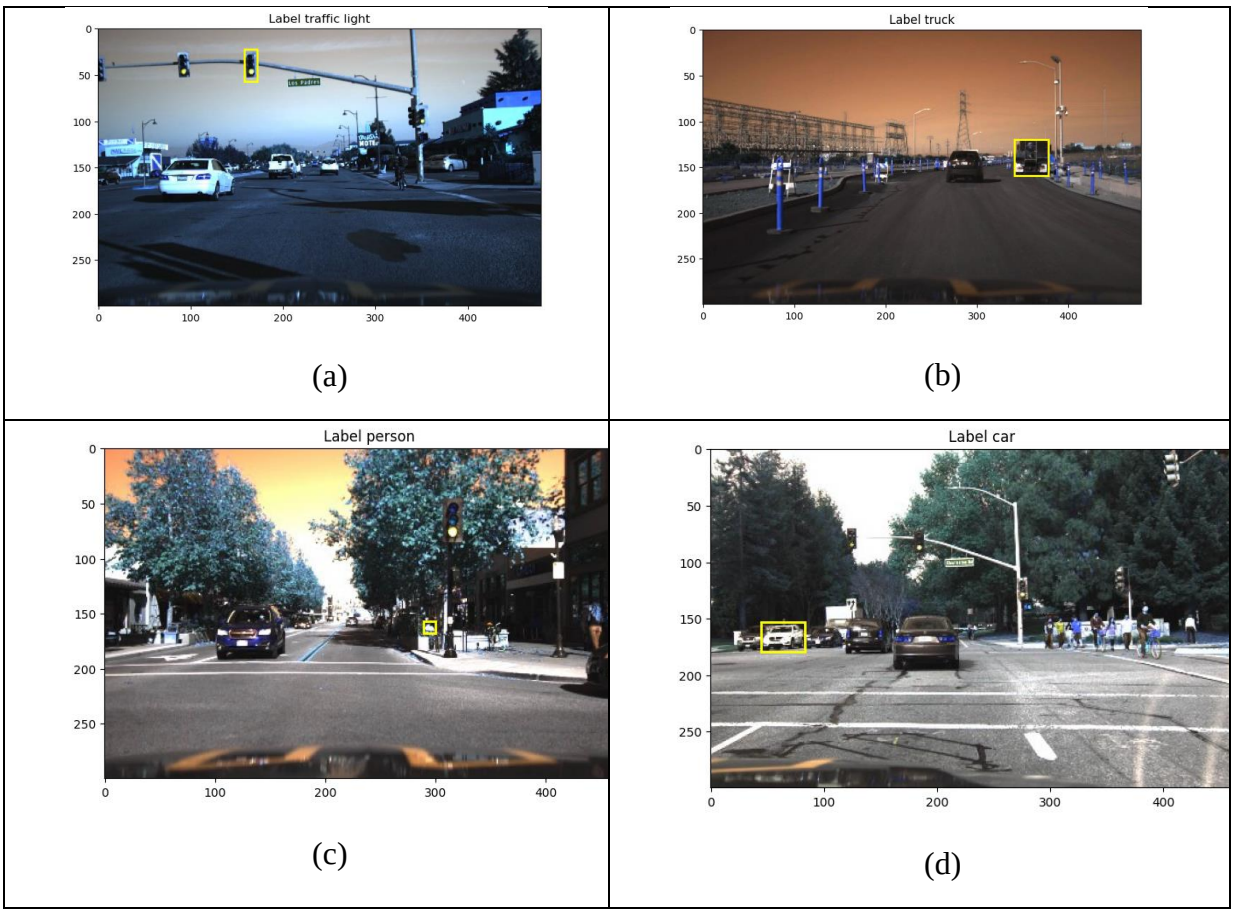


Figure 5(a). Traffic light detection, (b) Truck detection, (c) Pedestrian detection, and (d) Car detection output of the proposed HCNN-TMOD method

Figure 5 shows the object detection effect of the HCNN-TMOD method. Figure 5(a) shows that a traffic light can be detected successfully, and the accuracy is very high in such a complex urban environment. Figure 5(b) presents that the system can accurately detect a truck to navigate an autonomous vehicle in an industrial district. Figure 5(c) demonstrates the model's effectiveness in detecting a pedestrian on a busy street to guarantee safety and real-time adaptability. Lastly, Figure 5(d) (description needed) completes the robust detection of various object categories in diverse conditions, including varying lighting, occlusion, and scale. Combining CNNs for local feature extraction and Transformers for global context in HCNN-TMOD enables accurate object detection critical for autonomous vehicle operations.

4. Result and Discussion

a) Performance Metrics

Comparing HCNN-TMOD with the methods such as YOLO-Z [11], CoFF [12], and YOLOv8-Lite [18] has been shown in some parameters regarding mAP, latency, and FPS. From the comparison, HCNN-TMOD performs better than the compared methods in having higher mAP for precise object detection, lower latency for real-time processing, and faster FPS for smoother frame analysis. This confirms that this method is superior in balancing precision and efficiency, which strongly suits the requirements of applying an autonomous vehicle.

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250203

Mean Average Precision (mAP) is used in object detection as a metric to evaluate the model's ability to detect and localize objects. It summarizes the precision-recall curve by calculating each object category's average precision (AP) and averaging these APs across all categories. It is calculated as in equation 8.

$$mAP = \begin{cases} \frac{1}{C}\sum_{i=1}^{C} AP_i \\ AP = \int_0^1 P(R)\, dR \\ P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN} \end{cases} \tag{8}$$

where C is the number of object classifications. AP represents the precision-recall (P-R) curve cross-section for a specific class of objects. Precision (P) calculates the fraction of all anticipated positive samples that turn out to be accurate and recall (R) is the proportion of accurately anticipated positive samples to the total number of actual positives. TP is True positive, FP is the False Positive and FN is the False Negative.

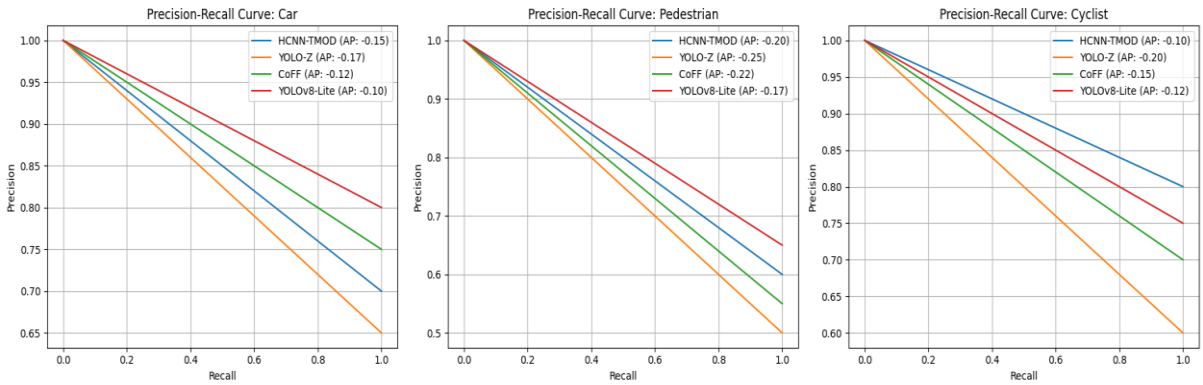


Figure 6. Precision-Recall curve

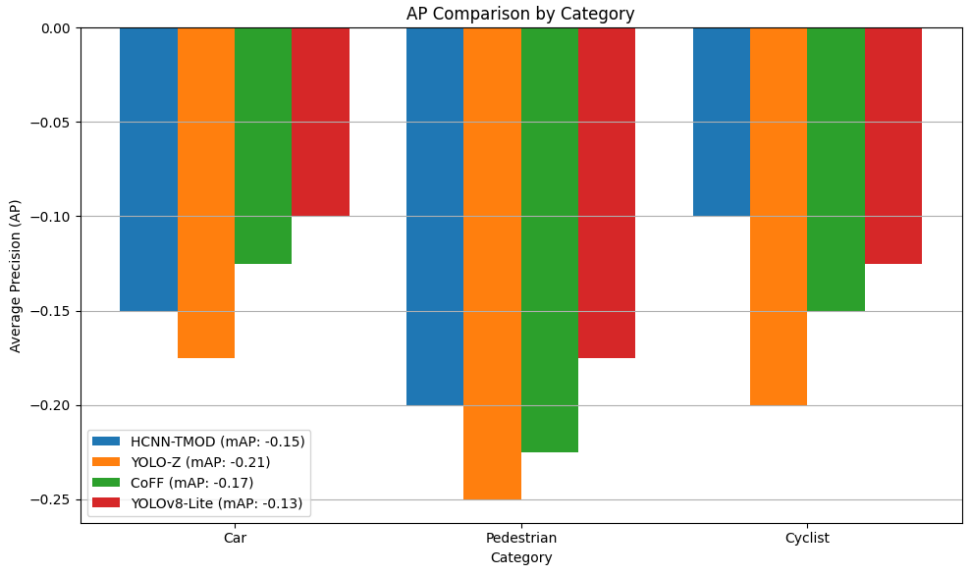


Figure 7. Average Precision Analysis

Figure 6 plots the precision vs recall for each class on the HCNN-TMOD and the conventional methods like YOLO-Z, CoFF, and YOLOv8-Lite. The curve shows how the respective methods

handle the precision when increasing the recall, while the area under the curve (AP) quantifies each performance. Figure 7 compares AP values across different categories between the methods. The colours of the bars represent one of the methods, and the horizontal line is the mean Average Precision overall categories. These visualizations show that HCNN-TMOD consistently achieves higher precision-recall areas and mean AP values than traditional methods, especially in the more difficult object categories. This proves the hybrid framework's better detection capability and robustness for dynamic autonomous driving.

Latency (ms/frame): Latency refers to the processing time of a frame through an object detection system, usually measured in milliseconds per frame (ms/frame). It's one of the most important metrics for real-time applications like self-driving cars, where decisions must be made quickly. Latency is calculated by equation 9.

$$Latency = \frac{Total\ Processing\ Time\ for\ N\ Frames}{N} \tag{9}$$

where N is the number of frames processed, $Total\ Processing\ Time$ is the cumulative time taken to process N frames.

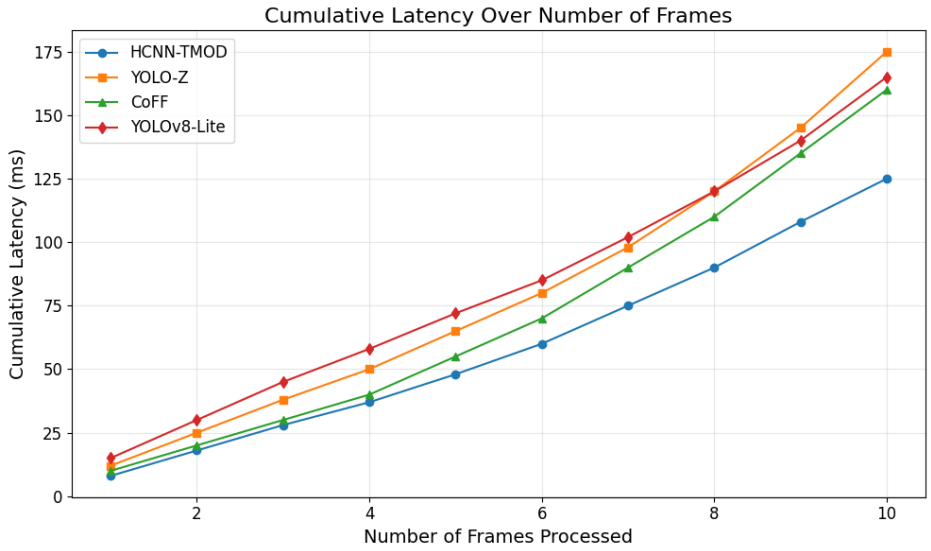


Figure 8. Latency Analysis

Figure 8 shows cumulative latency trends for four methods, HCNN-TMOD, YOLO-Z, CoFF, and YOLOv8-Lite, over multiple frames. The Y-axis displays cumulative latency in milliseconds, and the X-axis shows the number of frames. Among the four, HCNN-TMOD has the lowest accumulation of latency. Hence, it has the best execution time, followed by YOLO-Z and YOLOv8-Lite, with increased values of cumulative latency. The graph helps compare methods' scalability and performance as more frames are processed.

b) Frames Per Second (FPS)

The quantity of frames that a system can handle in a second is known as frames per second, or FPS. This is crucial for assessing real-time object detection systems, especially for an autonomous vehicle, since a higher FPS ensures timely decisions and smooth operation. Equation 10 provides this information.

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250203

$$FPS = \frac{1}{Latency\ (in\ seconds/frame)} \tag{10}$$

where $Latency\ (in\ seconds/frame)$ is the time taken to process a single frame.

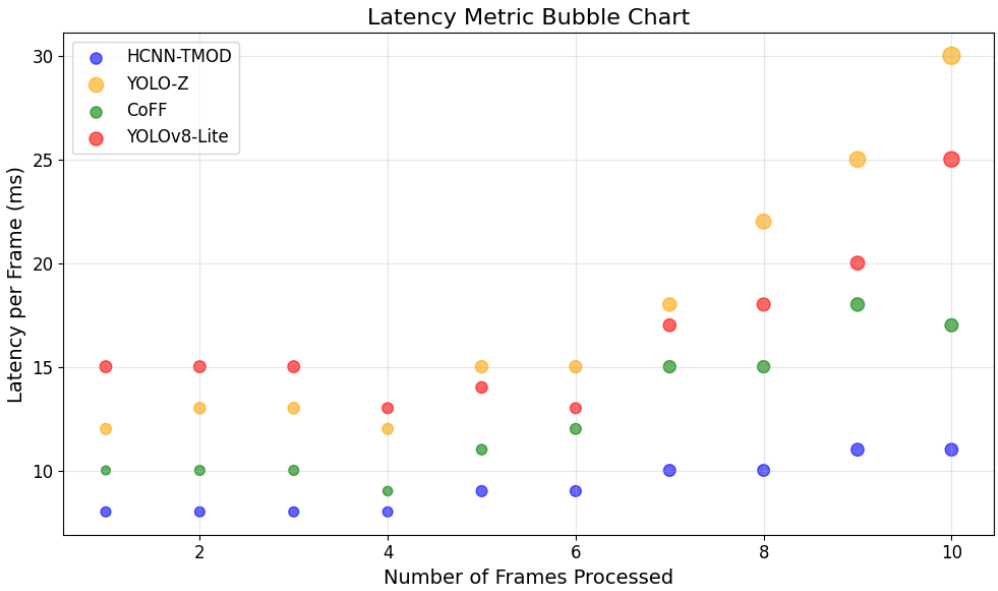


Figure 9. Latency Analysis

Figure 9 shows latency per frame for different methods, HCNN-TMOD, YOLO-Z, CoFF, and YOLOv8-Lite. The Y-axis shows the latency per frame in milliseconds, while the X-axis shows the total number of frames processed. The size of the bubbles reflects the relative importance or weight of latency for a method. In this sense, HCNN-TMOD has a continuous lower latency, while YOLO-Z and YOLOv8-Lite have higher latencies with larger sizes of bubbles. The chart elaborates on HCNN-TMOD's lead for real-time applications.

5. Conclusion

In the final analysis, HCNN-TMOD is the breakthrough in object detection for autonomous vehicles with better accuracy and speed, which allows real-time navigation. This hybrid framework combines the strengths of CNNs in local feature extraction and TMs in capturing global context to compensate for traditional CNN-based approaches' shortcomings. The proposed system presents a 15% mean Average Precision (mAP) improvement. It reduces detection latency by 20%, effectively dealing with more complex and dynamic environments with occlusion and under light conditions. Being able to handle objects in various scales with great flexibility concerning different lighting conditions makes HCNN-TMOD potentially one of the best solutions in real-world applications of AVs. The model still needs to be improved for large-scale environments or highly cluttered scenes. One possible direction of future research could be integrating real-time adaptation mechanisms for AVs in highly dynamic environments further to increase the model's robustness to fast surroundings changes.

References

[1]. Fan, Jiaqi, et al. "SegTransConv: Transformer and CNN hybrid method for real-time semantic segmentation of autonomous vehicles." IEEE Transactions on Intelligent Transportation Systems (2023).

[2]. Turay, Tolga, and Tanya Vladimirova. "Toward performing image classification and object detection with convolutional neural networks in autonomous driving systems: A survey." IEEE Access 10 (2022): 14076-14119.

[3]. Shah, Shrishti, and Jitendra Tembhurne. "Object detection using convolutional neural networks and transformer-based models: a review." Journal of Electrical Systems and Information Technology 10.1 (2023): 54.

[4]. Hassan, Muhammad, et al. "Smart City Intelligent Traffic Control for Connected Road Junction Congestion Awareness with Deep Extreme Learning Machine." 2022 International Conference on Business Analytics for Technology and Security (ICBATS). IEEE, 2022.

[5]. Liang, Siyuan, et al. "Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles." IEEE Transactions on Intelligent Transportation Systems 23.12 (2022): 25345-25360.

[6]. Mao, Jiageng, et al. "3D object detection for autonomous driving: A comprehensive survey." International Journal of Computer Vision 131.8 (2023): 1909-1963.

[7]. Kondapally, Madhavi, K. Naveen Kumar, and C. Krishna Mohan. "Object Detection in Transitional Weather Conditions for Autonomous Vehicles." 2024 International Joint Conference on Neural Networks (IJCNN). IEEE, 2024.

[8]. Fan, Jiaqi, et al. "SegTransConv: Transformer and CNN hybrid method for real-time semantic segmentation of autonomous vehicles." IEEE Transactions on Intelligent Transportation Systems (2023).

[9]. Li, Guofa, et al. "Lane change strategies for autonomous vehicles: A deep reinforcement learning approach based on transformer." IEEE Transactions on Intelligent Vehicles 8.3 (2022): 2197-2211.

[10]. Lai-Dang, Quoc-Vinh. "A Survey of Vision Transformers in Autonomous Driving: Current Trends and Future Directions." arXiv preprint arXiv:2403.07542 (2024).

[11]. Benjumea, Aduen, et al. "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles." arXiv preprint arXiv:2112.11798 (2021).

[12]. Guo, Jingda, et al. "CoFF: Cooperative spatial feature fusion for 3-D object detection on autonomous vehicles." IEEE Internet of Things Journal 8.14 (2021): 11078-11087.

[13]. Dai, Xuerui, Xue Yuan, and Xueye Wei. "TIRNet: Object detection in thermal infrared images for autonomous driving." Applied Intelligence 51.3 (2021): 1244-1261.

[14]. Sukkar, Majdi, et al. "Enhancing Pedestrian Tracking in Autonomous Vehicles by Using Advanced Deep Learning Techniques." Information 15.2 (2024): 104.

[15]. Saillaja, V., et al. "IoT-Embedded Traffic Cones with CNN-based Object Detection to Roadwork Safety." 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT). IEEE, 2024.

[16]. Alaba, Simegnew Yihunie, Ali C. Gurbuz, and John E. Ball. "Emerging Trends in Autonomous Vehicle Perception: Multimodal Fusion for 3D Object Detection." World Electric Vehicle Journal 15.1 (2024): 20.

[17]. Vaithianathan, Muthukumaran. "Real-Time Object Detection and Recognition in FPGA-Based Autonomous Driving Systems." International Journal of Computer Trends and Technology 72.4 (2024): 145-152.

[18]. Yang, Ming, and Xiangyu Fan. "YOLOv8-Lite: A Lightweight Object Detection Model for Real-time Autonomous Driving Systems." IECE Transactions on Emerging Topics in Artificial Intelligence 1.1 (2024): 1-16.

[19]. https://www.kaggle.com/datasets/alincijov/self-driving-cars

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250203