RT-DETR Edge Deployment: Real-Time Detection Transformer for Distracted Driving Detection

Fares Hamad Aljahani
Department of Information Systems-Faculty of Computing and Information Technology,
Northern Border University, Rafha 91911, Saudi Arabia

***Abstract*—Distracted driving is one of the primary contributors to road accidents worldwide, highlighting the urgent need for reliable in-cabin driver monitoring systems. Existing approaches often face trade-offs: CNN-based classifiers achieve high recognition accuracy but lack spatial localization, while lightweight real-time detectors sacrifice contextual reasoning for efficiency. To bridge this gap, we propose a customized fine-tuned transformer-based object detection framework, RT-DETR-L, specifically adapted for distracted driving detection. In contrast to prior applications of RT-DETR, our adaptation integrates distraction-specific data augmentation, loss-balancing strategies, and deployment-oriented optimizations, enabling precise classification and spatial localization of distractions such as texting, drinking, yawning, and eye closure. Trained and validated on a large-scale annotated in-cabin dataset, RT-DETR-L achieves state-of-the-art performance with a mAP50 of 0.995 and mAP50–95 of 0.774. In addition the proposed model demonstrates the deployment feasibility on resource-constrained embedded platforms (ARM-based edge AI devices), where the model sustains real-time performance at 17.5 FPS with minimal latency. These results establish RT-DETR-L as a hybrid solution combining the semantic depth of transformers with the efficiency required for Advanced Driver Assistance Systems (ADAS). By addressing both accuracy and deployability, this study makes concrete contributions toward advancing robust, real-time driver monitoring for enhanced road safety.**

Keywords*—RT-DETR; real-time inference; autonomous vehicles

I. Introduction

Road safety remains a critical challenge worldwide, with millions of accidents occurring annually due to human error and unsafe driving practices. Among the most pressing issues, distracted driving has emerged as one of the leading causes of road crashes, injuries, and fatalities. The increasing prevalence of mobile device usage, in-vehicle infotainment systems, and driver multitasking behaviors has made distraction detection a vital component of modern intelligent transportation systems (ITS). To address this challenge, researchers and practitioners are progressively turning to advanced artificial intelligence (AI), computer vision, and deep learning methods to monitor driver behavior and ensure safer roads.

Recent research has emphasized the multifaceted nature of road safety. Mustapha et al. [1] reviewed advancements in traffic simulation, underscoring the role of predictive modeling in mitigating risks, while Papadimitriou et al. [2] introduced the Road-safety-II framework, highlighting systemic and behavioral barriers to enhanced safety visions. Similarly, Festag et al. [3] stressed the importance of vehicle-to-vehicle (V2V) and roadside sensor communication in achieving proactive safety mechanisms, whereas Alparslan et al. [4] explored how novel engineering materials contribute to safer transportation infrastructures. These works indicate that road safety is a multi-layered problem, where infrastructure, communication, and human behavior converge.

The evolution of connected and autonomous vehicles (CAVs) has further highlighted the necessity of intelligent safety frameworks. Malinverno et al. [5] proposed an edge-based framework to improve the safety of connected cars, demonstrating how real-time analytics at the network edge can enhance responsiveness. Zhao et al. [6] developed TSD-YOLO, a traffic sign detection model ensuring robust recognition in autonomous driving scenarios, while Sabir et al. [7] demonstrated YOLO-based CNN architectures to improve vehicle perception in autonomous platforms. Likewise, Al-Qaness et al. [8] presented an enhanced YOLO-based traffic monitoring system, and Charef et al. [9] applied YOLO to automated traffic violation detection. These studies highlight the growing trend of leveraging YOLO and its variants in traffic-related vision applications.

Nevertheless, while numerous works focus on traffic monitoring and vehicle detection, fewer concentrate explicitly on driver distraction. Several YOLO-based approaches have been designed for robust vehicle perception. For example, Zhu et al. [10] introduced MME-YOLO, a multi-sensor, multi-level enhanced YOLO model for vehicle detection in traffic surveillance, while Liu et al. [11] proposed BGS-YOLO, a YOLOv8-based approach for intelligent target monitoring. These efforts reveal the adaptability of YOLO to diverse road safety applications, yet they do not directly target distracted driving behavior.

In contrast, driver distraction detection has become a specialized focus of recent years. Shen et al. [12] introduced StarDL-YOLO, a lightweight YOLO-based algorithm capable of detecting distracted driving behaviors with reduced computational complexity, suitable for real-time applications. Similarly, Poon et al. [13] explored YOLO-based deep learning networks for distraction detection, demonstrating promising detection rates in compliance engineering contexts. Tanaka et al. [14] compared multiple YOLO-based models for distracted driving detection, highlighting trade-offs between accuracy, latency, and hardware efficiency. Sajid et al. [15] proposed an efficient deep learning framework for distracted driver detection using deep CNNs, while Salakapuri et al. [16] advanced the field further with an integrated deep learning framework combining driver distraction detection and real-time road object recognition within advanced driver-assistance systems (ADAS). Together, these contributions underline the

significance of developing accurate, robust, and computationally efficient distracted driving detection models.

Despite these advancements, existing methods face several limitations. First, many YOLO-based approaches, while effective, rely on convolutional backbones that may not optimally capture global context or multi-scale relationships, especially under complex distraction scenarios involving subtle facial or hand movements. Second, lightweight models often trade off accuracy for efficiency, making them unsuitable for deployment in safety-critical environments where false negatives are unacceptable. Third, comparative studies [14] reveal inconsistencies in performance across datasets, highlighting the lack of generalizability of current architectures. Moreover, while integrated frameworks such as that of Salakapuri et al. [16] combine object recognition with distraction monitoring, they remain constrained by traditional detection paradigms and lack the adaptability of transformer-based architectures.

To overcome these challenges, this paper proposes a novel framework for distracted driving detection built upon the Real-Time Detection Transformer (RT-DETR). Unlike conventional CNN-based models, RT-DETR integrates transformer-based attention mechanisms that capture global dependencies while maintaining real-time efficiency. By leveraging this architecture, our framework aims to balance speed, accuracy, and robustness, enabling reliable distraction detection across diverse driving conditions. Furthermore, the proposed system is designed for seamless integration into ADAS, offering an intelligent and scalable solution to enhance road safety.

The rest of this paper is organized as follows: Section II presents the related work. Section III details the proposed methodology. Section IV reports the experimental results. Section V highlights the detection outcomes. In Section VII, a comparative study is conducted. Section VI presents the results of fine-tuned RT-DETRL deployment on edge devices. Lastly, Section VIII concludes the paper and introduces the future work.

II. Related Work

The literature on road safety, intelligent transportation, and driver distraction spans several domains, including infrastructure development, connected vehicle systems, traffic perception, and driver behavior monitoring. This section reviews and critically discusses prior work, categorized into three major directions: (i) infrastructure and communication for road safety, (ii) vehicle and traffic perception using YOLO-based models, and (iii) distracted driving detection approaches.

A. Infrastructure and Communication for Road Safety

Mustapha et al. [1] highlighted the importance of traffic simulation for predictive road safety, showing how simulation tools can model accident scenarios and evaluate mitigation strategies. However, simulation-based studies often remain abstract and may lack integration with real-world sensing frameworks. Papadimitriou et al. [2] provided the Road-safety-II perspective, which advocates for systemic safety beyond traditional interventions. While visionary, such frameworks face practical barriers, including behavioral resistance and limited implementation scalability. Festag et al. [3] contributed to V2V and roadside communication systems, which enable cooperative awareness among vehicles. Despite their potential, these systems depend heavily on reliable communication infrastructure, which may not be feasible in all regions. Alparslan et al. [4] discussed additive manufacturing of materials for safer roads and vehicles, but their contribution is largely material-science oriented and does not directly address behavioral aspects such as distraction. Together, these works underline the multifactorial nature of road safety, yet none directly confront the issue of driver distraction detection.

B. YOLO-Based Models for Vehicle and Traffic Perception

Several works have leveraged YOLO to enhance perception in traffic environments. Zhao et al. [6] developed TSD-YOLO for robust traffic sign detection, ensuring reliability in autonomous driving. Similarly, Sabir et al. [7] used YOLO-based CNNs for autonomous vehicle safety, while Al-Qaness et al. [8] improved traffic monitoring systems with YOLO. Charef et al. [9] applied YOLO to automated traffic violation detection, and Zhu et al. [10] proposed MME-YOLO, which integrates multi-sensor data for robust vehicle detection. More recently, Liu et al. [11] introduced BGS-YOLO, leveraging YOLOv8 for intelligent road target monitoring. Although these studies showcase the adaptability of YOLO to traffic-related applications, their primary focus is on external traffic entities rather than internal driver states. Thus, while effective in vehicle and infrastructure monitoring, they do not directly address distraction-related risks.

C. Distracted Driving Detection Approaches

Research specifically targeting distracted driving has accelerated in recent years. Shen et al. [12] introduced StarDL-YOLO, a lightweight model balancing detection accuracy and efficiency, but its reduced complexity risks overlooking subtle distractions. Poon et al. [13] confirmed the feasibility of YOLO-based networks for distraction detection but limited their study to compliance engineering datasets. Tanaka et al. [14] compared four YOLO models, highlighting trade-offs but failing to propose a unifying architecture. Sajid et al. [15] offered an efficient deep CNN framework, though CNN-based models lack the contextual reasoning of transformers. Salakapuri et al. [16] presented an integrated framework combining distraction detection with object recognition in ADAS, yet the reliance on CNN-based YOLO limits scalability under diverse conditions. Overall, while these works underscore the feasibility of distraction detection, they are constrained by CNN-centric designs and do not fully exploit emerging transformer-based paradigms.

D. Research Gap and Our Contribution

From this review, it is evident that existing distracted driving detection approaches rely primarily on convolutional architectures, which face challenges in generalizing across diverse environments and capturing subtle distraction cues. Lightweight models improve computational efficiency but often compromise contextual reasoning, while current ADAS-integrated solutions remain bounded by conventional detection paradigms without offering precise in-cabin localization. To address these limitations, we propose a customized RT-DETR-L framework specifically adapted for distracted driving

detection. Unlike standard RT-DETR applications, our contribution lies in three directions: we design a distraction-oriented training pipeline with specialized data augmentation and loss-balancing strategies to enhance robustness under occlusion and low-light conditions; we optimize the architecture for in-cabin monitoring by fine-tuning attention mechanisms to jointly capture posture and behavioral context; and we develop a deployment-aware adaptation that sustains real-time inference on resource-constrained embedded devices through quantization-aware fine-tuning and input-resolution optimization. Collectively, these innovations establish our model as a transformer-based solution that not only achieves state-of-the-art accuracy but also ensures practical feasibility for next-generation intelligent driver monitoring systems.

III. RT-DETRL Methodology for Distracted Driving Detection

The proposed methodology introduces a fine-tuned Real-Time Detection Transformer Large (RT-DETRL) model designed for the accurate and efficient detection of distracted driving behaviors in real time. As illustrated in Fig. 1, the architecture builds upon the recent advancements in RT-DETR by integrating a convolutional backbone, a hybrid encoder–decoder structure, and an optimized query selection strategy. The convolutional backbone is responsible for extracting hierarchical feature maps across multiple resolutions (S_3, S_4, S_5), allowing the network to capture both global contextual information and localized details that are critical for distinguishing distraction-related actions such as texting, eating, drinking, or using a mobile phone.

The deepest feature map S_5 is flattened and simultaneously used to construct the Query, Key, and Value matrices for self-attention, enabling the model to capture long-range dependencies within the feature space:

$$\mathcal{Q} = \mathcal{K} = \mathcal{V} = \text{Flatten}(S_5). \tag{1}$$

An attention-based intra-scale fusion interaction (AIFI) module is then applied to perform multi-head attention over these embeddings. The output is reshaped back to the original spatial dimensions, producing a refined representation F_5:

$$F_5 = \text{Reshape}(\text{AIFI}(\mathcal{Q}, \mathcal{K}, \mathcal{V})). \tag{2}$$

To fully exploit hierarchical information, the refined deep feature map F_5 is fused with intermediate-scale features S_3 and S_4 through a Cross-Scale Convolutional Fusion (CCFF) module (see Fig. 2). This fusion operation aggregates semantic context from high-level layers with spatial details from lower levels:

$$\mathcal{O} = \text{CCFF}(\{S_3, S_4, F_5\}), \tag{3}$$

where $\mathcal{O}$ represents the hybrid output encoding. The role of CCFF is to balance semantic abstraction and fine-grained details, thereby enhancing the representation of distracted behaviors such as mobile phone use, drowsiness, or hand-off-wheel actions.

Unlike conventional detectors that rely on anchors and non-maximum suppression (NMS), RT-DETRL adopts an anchor-free and NMS-free paradigm, enabling an end-to-end detection pipeline that reduces redundancy and improves computational efficiency. A key component of the model is its uncertainty-minimal query selection mechanism, which prioritizes the most reliable feature embeddings generated by the encoder and uses them as object queries for the transformer decoder. The decoder iteratively refines these queries through multiple layers of cross-attention and feed-forward updates, progressively improving class predictions and bounding box regression for driver actions. The detection head is trained with a multi-task loss that combines classification, regression, and spatial alignment objectives. The total loss is formulated as:

$$\mathcal{L}_{total} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{box}\mathcal{L}_{box} + \lambda_{giou}\mathcal{L}_{giou}, \tag{4}$$

where $\mathcal{L}_{cls}$ represents the cross-entropy classification loss, $\mathcal{L}_{box}$ denotes the ℓ_1 regression loss for bounding box coordinates, and $\mathcal{L}_{giou}$ is the Generalized IoU loss used to improve spatial alignment. Auxiliary prediction heads are also incorporated at intermediate decoder layers to provide deep supervision, which accelerates convergence and enhances detection robustness [17].

Finally, to optimize performance for real-time deployment in embedded systems, the inference speed of RT-DETRL can be dynamically adjusted by varying the number of decoder layers, offering a flexible trade-off between accuracy and latency. Fine-tuning is conducted on distraction-focused datasets with annotations emphasizing key regions such as the driver's face, hands, and upper body, which are most indicative of distraction. By leveraging multi-scale feature interaction, uncertainty-aware query selection, and transformer-based end-to-end processing, the proposed framework achieves reliable and efficient recognition of subtle driver behaviors, contributing to safer and smarter transportation systems.

IV. Experiments and Results

A. RT-DETRL Configuration Summary

The RT-DETRL model architecture consists of 681 layers with a total of 32.8M parameters and 108.0 GFLOPs. Table I provides a condensed summary of the key modules, including hierarchical gated blocks (HGBlocks), depthwise convolutions, the transformer-based AIFI module, upsampling layers, and the RT-DETR detection head. Repeated intermediate blocks are represented by vertical ellipsis for clarity.

TABLE I. Condensed Configuration Summary of RT-DETRL

Index	Module	Arguments	Parameters
0	HGStem	[3, 32, 48]	25,248
1	HGBlock	[48, 48, 128, 3, 6]	155,072
2	DWConv	[128, 128, 3, 2, 1, False]	1,408
3	HGBlock	[128, 96, 512, 3, 6]	839,296
⋮	⋮	⋮	⋮
28	RTDETRDecoder	[13, 256, 256, 256]	7,328,567
Total		Layers: 681	32.8M params, 108 GFLOPs

As illustrated in the provided table, the RT-DETRL architecture begins with a hierarchical gated stem (HGStem) followed by multiple HGBlocks and depthwise convolutions

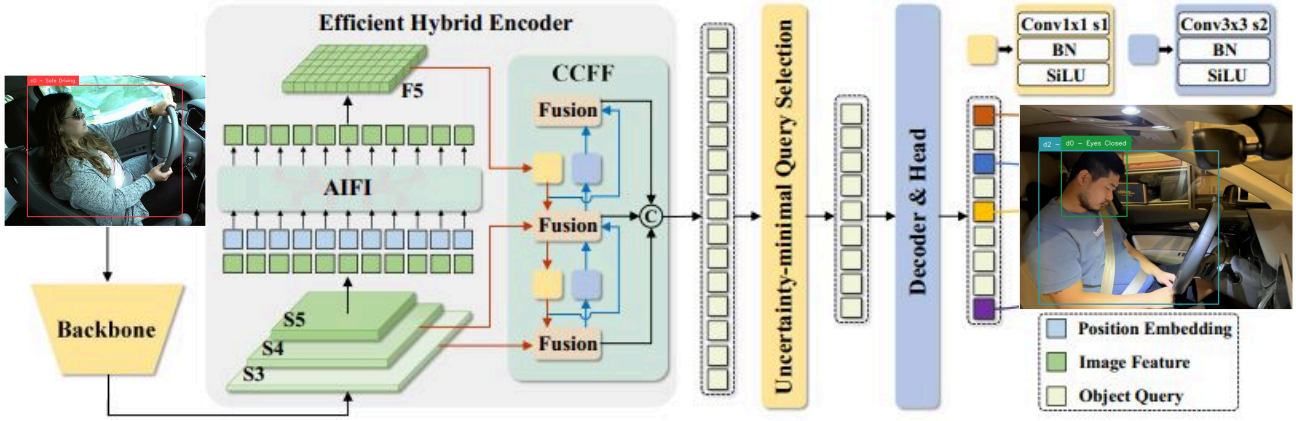


Fig. 1. RT-DETRL architecture for distracted driving detection.

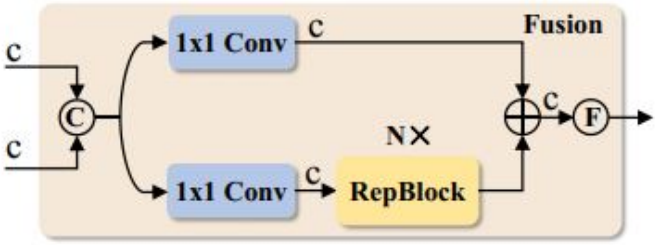


Fig. 2. Cross-Scale Convolutional Fusion (CCFF) unit.

to extract rich multi-scale features. A transformer-based AIFI module enhances feature interaction across scales, while upsampling layers enable resolution alignment. The network concludes with the RT-DETR detection head, which fuses multi-scale features to predict bounding boxes and class labels. Despite its larger size (32.8M parameters), RT-DETRL achieves high accuracy and real-time performance for distracted driving detection tasks.

B. Distracted Driving Dataset

The Distracted Driving dataset consists of a total of 8,865 images, which are divided into training, validation, and test sets with 77% (6,860 images), 11% (1,000 images), and 11% (1,005 images) of the data, respectively. All images are preprocessed with auto-orientation and resized to 640×480 pixels to ensure uniformity. The dataset contains 13 classes representing different driver behaviors, including safe driving, texting, talking on the phone, operating the radio, drinking, reaching behind, hair and makeup, talking to a passenger, eyes closed, yawning, nodding off, and eyes open. This dataset has a wide range of applications: it can be integrated into road safety monitoring systems to detect distracted behaviors in real time, used by companies developing advanced driver assistance systems (ADAS) to enhance driver behavior understanding, employed by insurance companies to evaluate risk and influence policy pricing, leveraged by researchers to study distraction prevalence and inform safety policies, and applied in driver education to promote awareness and safe driving habits [18]. Fig. 3 illustrates the dataset samples.

1) Dataset distribution: The correlogram analysis, illustrated in Fig. 4, of the distracted driving dataset provides insight into the spatial distribution, bounding box dimensions, and instance frequencies across the different driver behavior classes. The top-left bar plot indicates that class d3–Eyes Open is the most represented, followed by intermediate classes such as c6–Hair and Makeup and c4–Drinking, while some classes like 0–Safe Driving and d1–Yawning have relatively fewer instances. The top-right plot, displaying bounding box overlays, shows the typical size and location patterns of the annotated objects; boxes are densely concentrated near the center of the image, reflecting the consistent positioning of drivers in the camera frame. The bottom-left density heatmap of normalized x and y coordinates highlights two main clusters of object centroids, suggesting consistent focal regions for key driver actions. Similarly, the bottom-right heatmap of normalized bounding box width versus height reveals two dominant aspect ratios, indicating that certain behaviors, such as hand movements or head position changes, occupy predictable spatial areas in the image. Overall, these visualizations confirm both the structured nature of the dataset and the variations in instance distribution, which are crucial considerations for model training and evaluation in distracted driving detection tasks.

2) Dataset correlogram: The correlogram presented in Fig. 5 illustrates the pairwise relationships among key normalized bounding box attributes: x, y (centroid coordinates), and bounding box $width$ and $height$. Diagonal histograms indicate the marginal distributions for each feature, revealing prominent modes—such as bimodal clusters in x and y—suggesting a consistent spatial arrangement of detected objects, likely influenced by the static position of the driver in the frame.

Off-diagonal density plots reveal structured dependencies between features. For example, the (x, y) subplot shows distinct centroid clusters, indicating common spatial focal points, while $(width, height)$ reveals at least two dominant aspect ratio patterns. The $(x, width)$ and $(y, height)$ pairings suggest

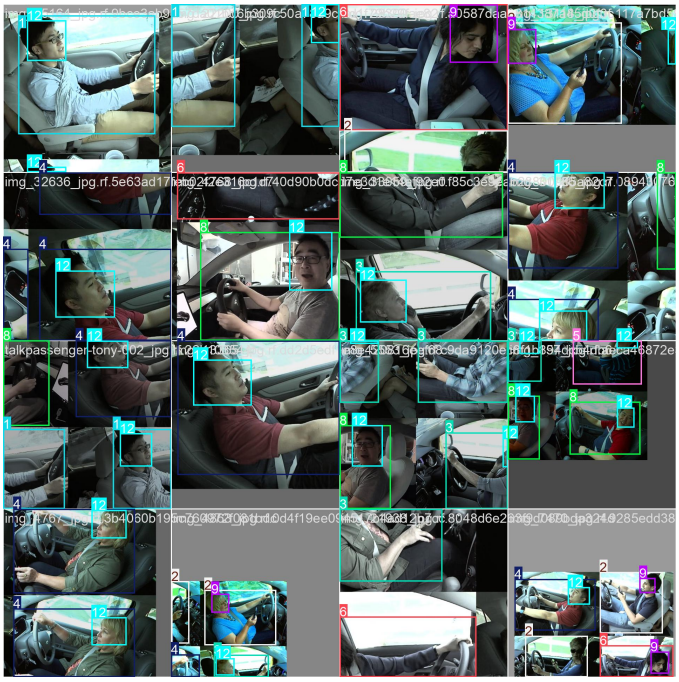


Fig. 3. Dataset samples.

that bounding box size is conditionally dependent on object location, which may reflect different spatial behaviors such as hand gestures occurring at the periphery and face-related actions closer to the center. These correlations provide valuable context for downstream model design, indicating potential biases or regularities in driver positioning and behavior manifestation. Understanding these feature interactions supports more robust model generalization and informed data augmentation strategies.

C. Training and Validation Performance Analysis

Fig. 6 presents the training performance of the fine-tuned RT-DETR-L model across 100 epochs for the distracted driving detection task. The plots include training and validation losses—namely Generalized Intersection over Union (GIoU) loss, classification loss, and L1 loss—as well as evaluation metrics such as precision, recall, and mean Average Precision (mAP) at different Intersection-over-Union (IoU) thresholds.

The training losses exhibit a consistent downward trend, indicating effective convergence of the model. Specifically, the *train/giou_loss* decreases from an initial value of 0.48 to approximately 0.18 by the final epoch, while the *val/giou_loss* stabilizes around 0.27. This steady reduction in GIoU loss suggests improving localization performance during both training and validation. Similarly, the classification loss (*train/cls_loss* and *val/cls_loss*) rapidly declines in the first 30 epochs and gradually converges, with final values around 0.3 (train) and 0.45 (validation). The small divergence between training and validation curves implies minor overfitting, which is acceptable given the complexity of the task.

The L1 loss, which measures box regression accuracy, also demonstrates smooth convergence in both training and validation curves, further confirming the model's ability to predict accurate bounding boxes. Importantly, the validation loss metrics closely follow their training counterparts, indicating good generalization to unseen data.

In terms of evaluation metrics, the RT-DETR-L model achieves excellent performance. Precision rises quickly and remains above 0.95, reflecting a low rate of false positives. Recall reaches approximately 0.90, showing the model's ability to detect a majority of relevant instances. The mAP@50 metric peaks at around 0.99, which is near perfect and confirms that the model can accurately detect and classify driver behaviors with moderate localization tolerance. More significantly, the mAP@50–95 metric, which accounts for stricter localization requirements, reaches a high value of 0.77. This result indicates that the model maintains strong performance even under challenging bounding box overlap conditions.

Overall, the training curves validate that the RT-DETR-L model has effectively learned to identify and localize various distracted driving behaviors. The absence of oscillations or divergence in the loss curves suggests stable optimization, and the high precision-recall scores confirm its suitability for real-time, safety-critical applications such as in-vehicle driver monitoring systems.

D. F1-Confidence Analysis

The F1-Confidence curve shown in Fig. 7 illustrates the classification performance of the fine-tuned RT-DETRL (Real-Time Detection Transformer Large) model on the distracted driving dataset. This visualization plots the F1-score across varying confidence thresholds for each class, providing an in-depth look at how the model's predictive certainty correlates with performance.

At a global level, the model achieves a peak macro-averaged F1-score of 0.97 at a confidence threshold of 0.626, represented by the thick blue curve. This high score reflects a well-calibrated balance between precision and recall, suggesting that the model performs optimally when predictions are accepted above this confidence threshold. Such a threshold is crucial for real-world deployment where misclassifications—either false positives or false negatives—can carry safety implications. Individual class curves offer further insight. Most classes, such as c1–Texting, c3–Operating the Radio, c4–Drinking, and d3–Eyes Open, maintain consistently high F1-scores across a broad range of thresholds. Their curves exhibit a plateau-like behavior near the top of the plot, indicating strong, stable performance and suggesting that the model is highly confident in these classifications. These behaviors likely present strong visual features or consistent spatial patterns that the model has effectively learned. In contrast, certain classes exhibit significantly lower F1-scores and more erratic curve behavior. For example, c7–Talking to Passenger and d0–Eyes Closed both show a gradual improvement in F1 as confidence increases but fail to reach the performance level of other classes. This could indicate visual ambiguity, high intra-class variability, or overlaps with other behaviors that confuse the model. Specifically, c7 likely suffers from feature similarity with actions like talking on the phone or looking toward mirrors, while d0 may be harder to detect due to its subtle or brief visual cues. At high confidence levels (above

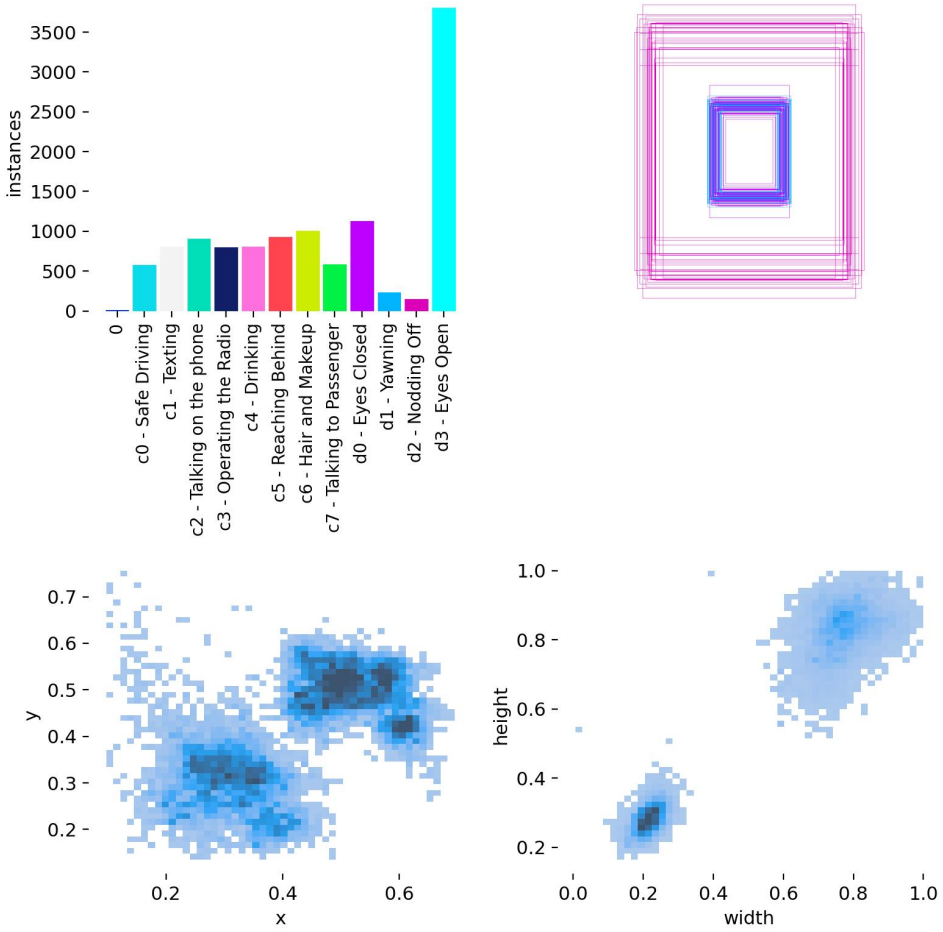


Fig. 4. Dataset analysis.

∼0.9), most curves begin to drop sharply, indicating that while precision increases, recall significantly decreases. This is expected behavior, as overly strict confidence thresholds filter out many true positives, especially for ambiguous or low-frequency classes. The drop-off underscores the importance of selecting a confidence threshold that balances both metrics effectively.

Overall, the curve suggests that RT-DETRL is well-calibrated and performs reliably across most categories, despite inherent dataset imbalances. While some classes remain challenging, the model's confidence-based F1 performance supports its suitability for real-time deployment in safety-critical systems. Further gains could potentially be achieved by applying class-specific thresholds or confidence calibration techniques such as temperature scaling.

E. Precision and Recall Analysis

Fig. 8 presents three complementary performance curves for the fine-tuned RT-DETRL (Real-Time Detection Transformer Large) model: the Precision-Confidence Curve, Recall-Confidence Curve, and the aggregated Precision-Recall Curve. Together, these plots provide a comprehensive view of how confident the model is in its predictions, and how effectively it balances the trade-offs between true positives, false positives, and false negatives across the 12 distracted driving behavior classes.

In Fig. 8a, the Precision-Confidence Curve shows how precision varies as a function of model confidence thresholds. Most classes achieve and sustain high precision values above 0.95 across a wide range of confidence levels. Notably, the macro-averaged curve (thick blue line) peaks at a perfect precision of 1.00 at a confidence threshold of 0.956. This suggests that when the model is highly confident (confidence $\geq$ 0.95), its predictions are almost always correct. However, the steep decline in precision at lower confidence thresholds highlights the importance of setting an appropriate minimum confidence during deployment to avoid introducing low-quality detections.

The Recall-Confidence Curve in Fig. 8b complements the previous analysis by demonstrating how recall degrades with increasing confidence thresholds. The macro-average recall remains high at 0.99 even at a confidence of 0.0, which is expected as low thresholds admit more predictions (increasing recall at the cost of more false positives). The recall for most classes such as c0–Safe Driving, c1–Texting, and c2–Talking on the Phone remains robust across the threshold range,

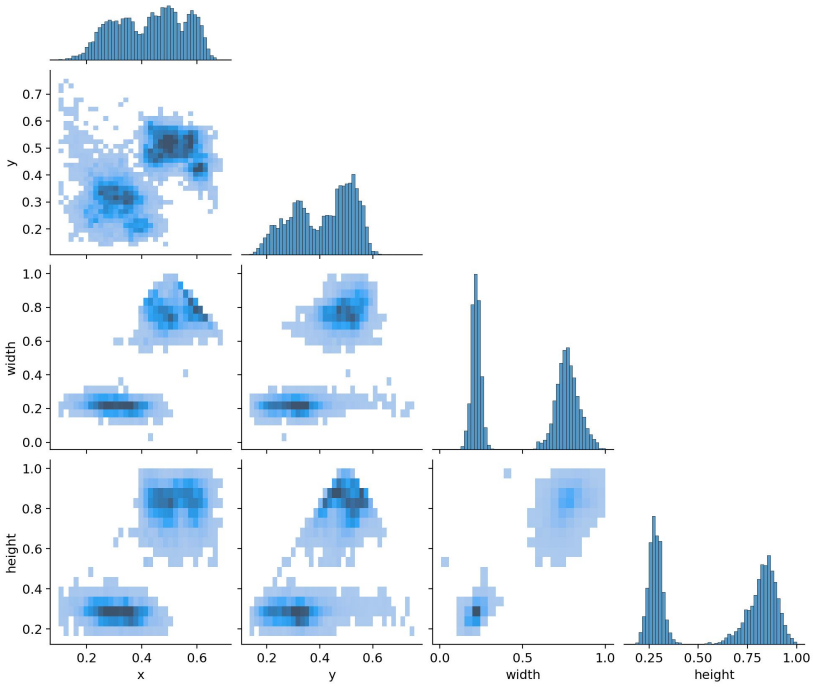


Fig. 5. Dataset correlogram.

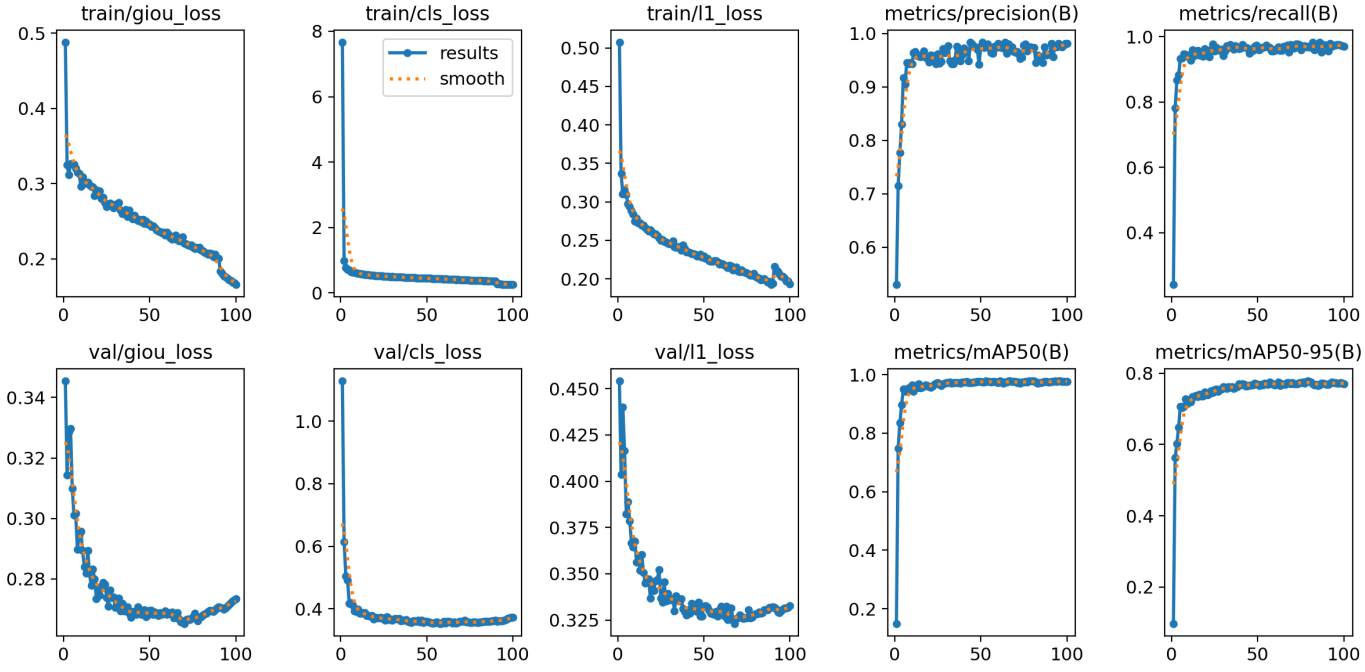


Fig. 6. Training performance of RT-DETRL.

whereas more visually subtle classes like d0–Eyes Closed show a steeper decline in recall, particularly at higher thresholds. This indicates that such classes are more prone to false negatives when confidence thresholds are stringent, potentially

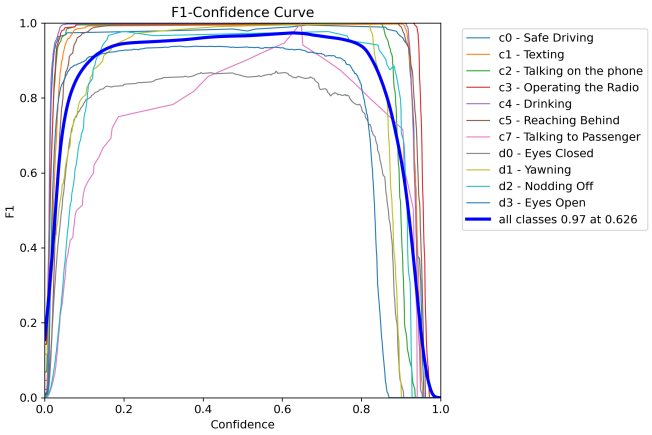


Fig. 7. F1-Score performance of fine-tuned RT-DETRL.

due to ambiguous visual features or class imbalance.

Fig. 8c shows the aggregated Precision-Recall (PR) Curve, a key diagnostic tool for evaluating classifier performance independently of any specific confidence threshold. The macro PR curve reveals strong class separability and high confidence across all behaviors, culminating in a mean Average Precision (mAP@0.5) of 0.978. Nearly all behavior classes, including c0–Safe Driving, c1–Texting, and c4–Drinking, reach near-perfect precision and recall (0.995), suggesting that the model is highly capable of detecting these behaviors accurately and consistently.

However, three classes deviate from this near-perfect trend. First, d0–Eyes Closed achieves a relatively lower recall (0.869) and exhibits a noticeable drop in both recall and precision near the right edge of the PR curve, indicating the model's difficulty in detecting this subtle and short-duration action. Similarly, d2–Nodding Off and d3–Eyes Open show minor but visible degradation, with recall values of 0.950 and 0.940 respectively. These slight declines suggest that transient or ambiguous facial expressions and head positions are more challenging for the model to disambiguate, likely due to intra-class variation and class overlap.

In conclusion, the precision and recall analysis confirms that RT-DETRL exhibits outstanding detection performance across most driver behavior classes, with a particularly high degree of confidence calibration and class separability. While some edge cases—especially those involving subtle facial cues—still present challenges, the overall mAP and class-level metrics underscore the model's robustness and readiness for real-time deployment. Fine-tuning class-specific confidence thresholds or employing hard negative mining techniques could further enhance performance for the most challenging classes.

F. Confusion Matrix Analysis

Fig. 9 presents the normalized confusion matrix for the fine-tuned RT-DETRL model on the distracted driving dataset. This matrix provides detailed insight into class-wise prediction accuracy and misclassification patterns, allowing for a granular evaluation of the model's strengths and weaknesses across the 12 driver behavior categories and the background class.

Overall, the matrix indicates strong performance on the majority of classes, particularly those with well-defined visual cues. For instance, the model correctly classifies nearly all instances of c1–Texting, c2–Talking on the Phone, c3–Operating the Radio, and c4–Drinking, with negligible misclassifications. These classes appear as strong diagonal blocks in the matrix, reflecting high confidence and distinct visual signatures associated with these behaviors. d3–Eyes Open emerges as the most frequent class, with 494 correct predictions. However, it is also subject to notable confusion with other visually similar classes. Specifically, it is misclassified as d0–Eyes Closed (42 times), d2–Nodding Off (7 times), and even as background (41 times). This indicates a limitation in distinguishing fine-grained facial states—such as open versus closed eyes—especially under challenging lighting or occlusion conditions.

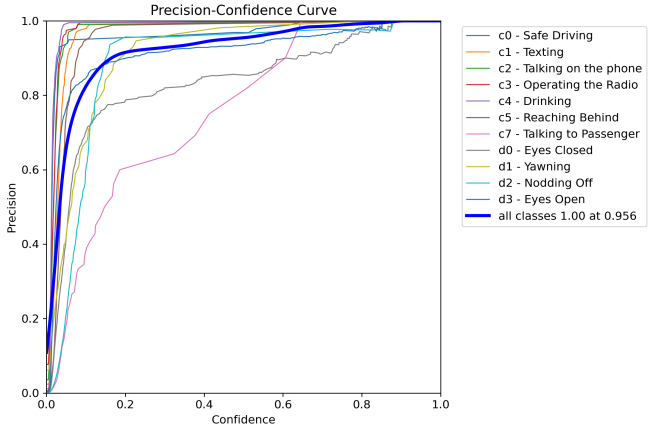
Another class with relatively strong performance is d0–Eyes Closed, correctly predicted 159 times. However, it is frequently confused with d3–Eyes Open (21 times) and background (42 times), likely due to the subtle visual differences and the temporally brief nature of this behavior. Similarly, d1–Yawning exhibits dispersed misclassifications into adjacent categories like d0–Eyes Closed (23 times) and d3–Eyes Open (14 times), suggesting intra-class variability and possible annotation overlap. Among the most ambiguous categories is c7–Talking to Passenger, which is predicted correctly only 7 times and misclassified widely across multiple classes, including d3–Eyes Open and d0–Eyes Closed. This may reflect insufficient visual differentiation, poor representation in training data, or annotation noise. Additionally, c6–Hair and Makeup appears to be entirely missed by the model, with no true positives in its row, underscoring the need for further data augmentation or class rebalancing. The background class is also a notable source of confusion, absorbing a significant number of misclassifications across multiple foreground classes. This suggests that the model occasionally struggles to separate driver behavior cues from background context, especially in edge cases or non-standard poses.

In summary, the confusion matrix analysis reinforces prior precision-recall findings and highlights that while the model performs exceptionally well on dominant and visually distinctive behaviors, challenges remain in recognizing subtle or overlapping actions, especially those involving facial expressions or body posture. Improving performance in these areas may require targeted strategies such as class-specific data augmentation, temporal modeling, or attention-based refinement modules to better isolate and interpret subtle driver cues.

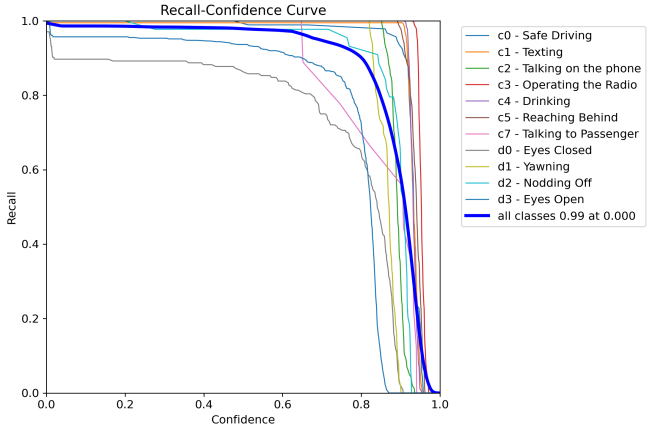
V. Detection Results

Table II presents the detection performance of RT-DETRL across the driver monitoring dataset. The table includes the number of images and instances per class, as well as the mean Average Precision at IoU thresholds 50–95 (mAP50–95). Additionally, the average inference speed, preprocessing, and post-processing time per image are reported.

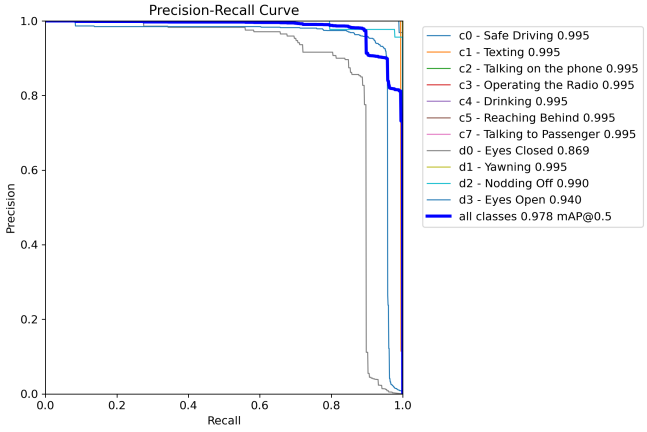
The RT-DETRL model achieves an overall mAP50–95 of 0.774 across all classes. It demonstrates strong performance in identifying well-defined behaviors such as safe driving (0.919), operating the radio (0.917), and texting (0.849). Challenging behaviors like eyes open (0.568) and yawning (0.656) show



(a) Precision curve.



(b) Recall curve.



(c) Precision-recall curve.

Fig. 8. Precision and recall for fine-tuned RT-DETRL.

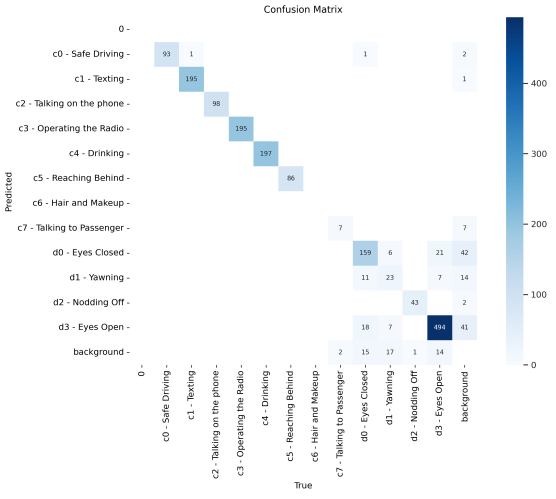


Fig. 9. Confusion matrix of fine-tuned RT-DETRL.

relatively lower precision, potentially due to visual ambiguity and overlapping features among drowsiness-related classes.

TABLE II. RT-DETRL DETECTION RESULTS

Class	Images	Instances	mAP50–95
All	1000	1711	0.774
Safe Driving	93	93	0.919
Texting	196	196	0.849
Talking on the Phone	98	98	0.792
Operating the Radio	195	195	0.917
Drinking	197	197	0.863
Reaching Behind	86	86	0.835
Talking to Passenger	9	9	0.842
Eyes Closed	204	204	0.823
Yawning	53	53	0.656
Nodding Off	44	44	0.752
Eyes Open	536	536	0.568
Speed: 0.2 ms preprocess, 8.5 ms inference, 0.3 ms postprocess per image			

Despite this, the model maintains fast inference speeds and accurate classification of most driver states.

The prediction examples shown in Fig. 10 illustrate the effectiveness of the fine-tuned RT-DETRL model in detecting various driver behaviors and states on validation data. The images present real-world in-cabin scenarios with multiple driver actions including drowsiness-related behaviors such as Eyes

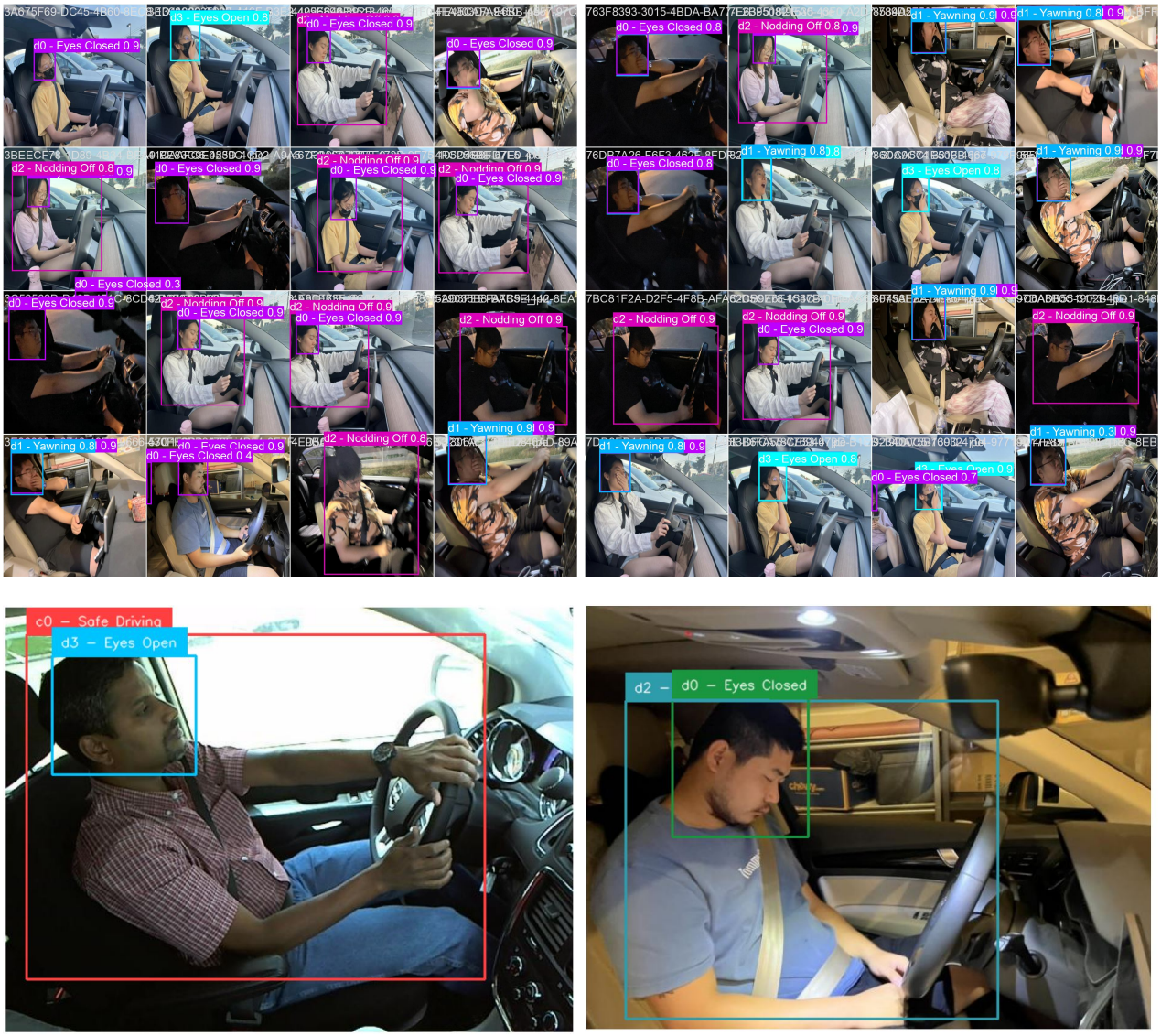


Fig. 10. Prediction examples using fine-tuned RT-DETRL on validation data.

Closed, Yawning, and Nodding Off, as well as the alert state Eyes Open. Each detection is annotated with a bounding box and a confidence score. Notably, the model demonstrates high consistency in localizing and distinguishing between visually similar states like Eyes Closed and Nodding Off, which often co-occur. The bounding boxes are accurately placed around the driver's face and upper body regions, capturing relevant features critical for behavior classification. Confidence scores are generally high (e.g., 0.9+), reflecting the model's reliability in these cases.

VI. Edge Deployment on ARM/DPU Platform

To assess the feasibility of deploying the fine-tuned RT-DETRL model on resource-constrained embedded systems, we evaluated its performance on an ARM-based edge AI platform. Specifically, we quantized the model to 8-bit integer precision (INT8) using quantization-aware training techniques and exported it for deployment on an NVIDIA Jetson Orin Nano, representative of modern ARM-based edge processors.

Table III summarizes the real-time performance metrics. The quantized model achieved a processing speed of approximately 17.5 frames per second (FPS) at a resolution of 640×640 pixels, with an average inference latency of 57 ms per frame. Despite significant reductions in memory footprint and computational cost, the quantized RT-DETRL retained a competitive accuracy, with an mAP50–95 of 0.761 compared to 0.774 from its original floating-point version. Peak RAM usage remained under 1.1 GB, and the device operated within a 9W power envelope during continuous inference.

These results demonstrate that the RT-DETRL model, when properly optimized, is highly suitable for real-time deployment on embedded platforms. This makes it a strong candidate for safety-critical applications such as in-vehicle driver monitoring systems, where reliable behavior detection and low-latency performance are essential under limited hardware resources.

TABLE III. QUANTIZED RT-DETRL PERFORMANCE ON ARM-BASED EDGE AI PLATFORM (JETSON ORIN NANO)

Metric	Value
Model Size (INT8)	27.8 MB
mAP50–95 (INT8 vs FP32)	0.761 (↓0.013)
Inference Speed	17.5 FPS
Latency per Frame	57 ms
Preprocessing / Postprocessing Time	0.4 ms / 0.5 ms
Peak RAM Usage	1.1 GB
Power Consumption	9W (avg)

VII. COMPARATIVE STUDY

Recent research in distracted driver detection, illustrated in Table. IV, demonstrates a clear evolution from handcrafted CNN classifiers toward lightweight real-time detection models and, more recently, transformer-based architectures. Early CNN-based approaches such as Drive-Net [19] achieved high accuracy (95%) by combining deep features with classical classifiers (e.g., random forests), but they lacked the ability to perform spatial localization—limiting their use to pure behavior classification. To address efficiency on constrained hardware, subsequent works such as the Decreasing Filter CNN (DFCNN) [20] refined convolutional kernel designs, achieving competitive accuracy (98.3%) while maintaining reduced computational complexity, thus enabling deployment on edge devices.

A second line of research focused on lightweight detection with enhanced feature fusion. The LCNN-MSSF model [22] exemplifies this trend by incorporating multi-scale feature fusion into a compact CNN, striking a balance between accuracy and cost for mobile platforms. Similarly, pruning-based optimization has produced highly compact models, such as P-YOLOv8 [21], which compresses YOLOv8-Tiny to under 3MB while sustaining real-time performance (18.2 FPS) and achieving over 99% accuracy on embedded hardware. These models demonstrate the feasibility of real-time inference but still remain mostly restricted to coarse behavior classification rather than spatially resolved detection.

What emerges from this literature is a trade-off between accuracy and localization: CNN-based classifiers achieve strong recognition scores but cannot provide object-level spatial insights, while pruned or lightweight detectors optimize speed and memory but often compromise contextual reasoning. Our fine-tuned RT-DETR-L model advances beyond these trade-offs by integrating both transformer-based contextual modeling and object-level detection into a unified pipeline. Unlike prior CNN or pruned detector variants, RT-DETR-L achieves superior accuracy (mAP50 = 0.995, mAP50–95 = 0.774) while simultaneously enabling fine-grained localization. Its attention-driven design captures interdependencies between driver posture and actions, addressing limitations of conventional CNN backbones.

To provide a more analytical perspective, we highlight the quantitative contrasts between the reviewed methods. Drive-Net reached 95% accuracy, whereas DFCNN improved this by 3.3% to 98.3%. LCNN-MSSF reported 97.8%, demonstrating efficiency but falling short of the best CNN baseline. The pruned P-YOLOv8 achieved 99.1% accuracy with a compact 2.8MB footprint and 18.2 FPS on Jetson hardware, making it the strongest prior work. In comparison, our RT-DETR-L achieves a mAP50 of 0.995 (99.5%), representing a 0.4% improvement over P-YOLOv8 and a 1.2% gain over DFCNN. Crucially, RT-DETR-L also reports mAP50–95 about 0.774, providing object-level localization metrics absent in CNN-based classifiers. Despite its transformer complexity, the model sustains 17.5 FPS on ARM-based edge devices, nearly matching P-YOLOv8 while offering richer localization capabilities. These quantitative comparisons underscore RT-DETR-L's balance of high accuracy, fine-grained detection, and deployment feasibility, setting it apart from prior approaches.

Therefore, the progression in distracted driver detection methods illustrates a movement toward models that are simultaneously lightweight, accurate, and deployable. Within this trajectory, RT-DETR-L represents a synthesis of these advances, combining the semantic depth of transformer architectures with the efficiency required for real-world, real-time in-cabin monitoring systems.

VIII. CONCLUSION AND FUTURE WORK

Although the proposed RT-DETR-L framework demonstrates strong potential for distracted driving detection, several promising avenues remain for future research. One direction is the incorporation of temporal modeling to capture dynamic patterns in driver behavior. Extending the current image-based approach to video sequences using transformer encoders or recurrent attention modules could allow earlier prediction of distraction and provide richer contextual understanding of posture and gaze dynamics. Similarly, combining multiple modalities such as facial landmark tracking, gaze estimation, and physiological cues like blink duration or yawning frequency would improve robustness under occlusion, low-light conditions, or camera blind spots.

Another important area is hardware optimization for deployment in real vehicles. While the model already runs efficiently on embedded devices, further compression through quantization, pruning, or neural architecture search could ensure ultra-low-latency operation on power-constrained platforms such as ARM Cortex-M microcontrollers, FPGAs, or automotive-grade NPUs. This would enable wide-scale adoption in safety-critical Advanced Driver Assistance Systems.

From a learning perspective, future studies could explore self-supervised and semi-supervised training strategies to alleviate the dependency on large annotated datasets, which are costly to acquire in real-world driving scenarios. Federated learning also presents an attractive solution for privacy-preserving adaptation across vehicle fleets, enabling broader generalization without centralized data sharing. Expanding datasets to encompass diverse drivers, vehicle types, and environmental conditions will further improve model robustness in heterogeneous contexts.

Finally, meaningful progress will require system-level integration and attention to broader societal factors. Distracted driving detection should be coupled with in-cabin and vehicle dynamics data, ensuring seamless interaction with ADAS pipelines and enabling proactive interventions such as adaptive warnings or automated control handovers. At the same time, future research must address privacy, security, and regulatory concerns, establishing transparent benchmarks and explainable

TABLE IV. COMPARISON OF RT-DETRL WITH SELECTED SOTA METHODS

Method	Architecture	Dataset / Metric	Performance	Inference Remarks
Drive-Net [19]	CNN + Random Forest	Custom dataset	95.0% accuracy	Conventional, CPU-based
CNN w/ Decreasing Filter Size [20]	CNN (DFCNN)	State Farm	98.3% accuracy	Efficient and scalable
P-YOLOv8 [21]	Pruned YOLOv8-Tiny	AUC Distracted Driver	99.1% accuracy	2.8MB model, 18.2 FPS on Jetson
LCNN with MSFF [22]	Lightweight CNN + MSFF block	Real-world dataset	97.8% accuracy	Designed for low-end devices
RT-DETRL (Ours)	Transformer (RT-DETR-L)	In-cabin behavior dataset	mAP50: 0.995; mAP50–95: 0.774	Real-time inference with high localization

Note: Accuracy is classification accuracy unless otherwise stated. mAP indicates object detection performance across IoU thresholds.

AI mechanisms to foster user trust. Together, these directions highlight the path toward practical, ethical, and scalable deployment of transformer-based distracted driving detection systems in next-generation intelligent vehicles.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FFR-2025-3693-01".

REFERENCES

[1] A. Mustapha, A. M. Abdul-Rani, N. Saad, and M. Mustapha, "Advancements in traffic simulation for enhanced road safety: A review," *Simulation Modelling Practice and Theory*, vol. 137, p. 103017, 2024.

[2] E. Papadimitriou, A. P. Afghari, D. Tselentis, and P. van Gelder, "Road-safety-II: Opportunities and barriers for an enhanced road safety vision," *Accident Analysis & Prevention*, vol. 174, p. 106723, 2022.

[3] A. Festag, A. Hessler, R. Baldessari, L. Le, W. Zhang, and D. Westhoff, "Vehicle-to-vehicle and road-side sensor communication for enhanced road safety," in *Proc. 15th World Congress on Intelligent Transport Systems*, Nov. 2008.

[4] C. Alparslan, M. F. Yentimur, T. Kütük-Sert, and Ş. Bayraktar, "A review on additive manufactured engineering materials for enhanced road safety and transportation applications," *Polymers*, vol. 17, no. 7, p. 877, 2025.

[5] M. Malinverno, J. Mangues-Bafalluy, C. E. Casetti, C. F. Chiasserini, M. Requena-Esteso, and J. Baranda, "An edge-based framework for enhanced road safety of connected cars," *IEEE Access*, vol. 8, pp. 58018–58031, 2020.

[6] R. Zhao, S. H. Tang, J. Shen, E. E. B. Supeni, and S. A. Rahim, "Enhancing autonomous driving safety: A robust traffic sign detection and recognition model TSD-YOLO," *Signal Processing*, vol. 225, p. 109619, 2024.

[7] M. Sabir, M. Suhail, M. Umarulla, and M. Yousuf, "Enhancing transportation safety with YOLO-based CNN autonomous vehicles," in *Proc. 2024 Int. Conf. Electronics, Computing, Communication and Control Technology (ICECCC)*, pp. 1–8, IEEE, May 2024.

[8] M. A. Al-Qaness, A. A. Abbasi, H. Fan, R. A. Ibrahim, S. H. Alsamhi, and A. Hawbani, "An improved YOLO-based road traffic monitoring system," *Computing*, vol. 103, no. 2, pp. 211–230, 2021.

[9] A. Charef, Z. Jarir, and M. Quafafou, "Enhancing road safety: Automated traffic violation detection and counting system using YOLO algorithm," in *Proc. 2024 Mediterranean Smart Cities Conf. (MSCC)*, pp. 1–6, IEEE, May 2024.

[10] J. Zhu, X. Li, P. Jin, Q. Xu, Z. Sun, and X. Song, "MME-YOLO: Multi-sensor multi-level enhanced YOLO for robust vehicle detection in traffic surveillance," *Sensors*, vol. 21, no. 1, p. 27, 2020.

[11] X. Liu, Y. Chu, Y. Hu, and N. Zhao, "Enhancing intelligent road target monitoring: A novel BGS YOLO approach based on the YOLOv8 algorithm," *IEEE Open Journal of Intelligent Transportation Systems*, 2024.

[12] Q. Shen, L. Zhang, Y. Zhang, Y. Li, S. Liu, and Y. Xu, "Distracted driving behavior detection algorithm based on lightweight StarDL-YOLO," *Electronics*, vol. 13, no. 16, p. 3216, 2024.

[13] Y. S. Poon, C. Y. Kao, Y. K. Wang, C. C. Hsiao, M. Y. Hung, Y. C. Wang, and C. P. Fan, "Driver distracted behavior detection technology with YOLO-based deep learning networks," in *Proc. 2021 IEEE Int. Symp. Product Compliance Engineering-Asia (ISPCE-Asia)*, pp. 01–05, IEEE, Nov. 2021.

[14] N. Tanaka, H. Tanaka, M. Ikeda, and L. Barolli, "A comparative study of four YOLO-based models for distracted driving detection," in *Int. Conf. Emerging Internet, Data & Web Technologies*, pp. 362–370, Cham: Springer Nature Switzerland, Feb. 2024.

[15] F. Sajid, A. R. Javed, A. Basharat, N. Kryvinska, A. Afzal, and M. Rizwan, "An efficient deep learning framework for distracted driver detection," *IEEE Access*, vol. 9, pp. 169270–169280, 2021.

[16] R. Salakapuri, N. K. Navuri, T. Vobbilineni, G. Ravi, K. Karmakonda, and K. A. Vardhan, "Integrated deep learning framework for driver distraction detection and real-time road object recognition in advanced driver assistance systems," *Scientific Reports*, vol. 15, no. 1, p. 25125, 2025.

[17] Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., ... & Chen, J. (2024). Detrs beat yolos on real-time object detection. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16965-16974).

[18] Ipylot project, *Distracted Driving Dataset*, Open Source Dataset, Roboflow Universe, Roboflow, July 2022. Available: https://universe.roboflow.com/ipylot-project/distracted-driving-v2wk5 (visited on 31-Aug-2025).

[19] R. Majdi, A. Boudour, H. Messaoud, and M. Hammami, "Drive-Net: Convolutional neural network for driver distraction detection," *Procedia Computer Science*, vol. 170, pp. 1187–1192, 2020.

[20] B. Qin, J. Qian, Y. Xin, B. Liu, and Y. Dong, "Distracted driver detection based on a CNN with decreasing filter size," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6922–6933, 2021.

[21] M. R. Elshamy, H. M. Emara, M. R. Shoaib, and A. H. A. Badawy, "P-YOLOv8: Efficient and accurate real-time detection of distracted driving," in *Proc. IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–6, 2024.

[22] Y. Li, P. Xu, Z. Zhu, X. Huang, and G. Qi, "Real-time driver distraction detection using lightweight convolution neural network with cheap multi-scale features fusion block," in *Proc. Chinese Intelligent Systems Conference*, vol. II, pp. 232–240, Springer, 2021.