ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2026.0568

IMPROVING TIMELINESS ALERT IN AI DRIVEN DRIVER ASSIST SYSTEMS (ADAS)

G. Ananthi, G. Maha Vignesh and K.P. Balajiraj
Department of Computer Science and Engineering, Mepco Schlenk Engineering College, India

Abstract

Artificial intelligence-based object detection models play a vital role in recognizing pedestrian and sending collision warnings in Advanced Driver Assistance Systems (ADAS). Although the deep learning-based modern detection algorithms provide accurate pedestrian detection under benchmark environment conditions, it has been shown by the research on unreliable pedestrian detection that detection results can be temporally unstable and delayed accumulation of confidence in a realistic scene. The delay or unstable behavior of a detection model in safety-critical applications such as driver assistance systems can cause problems in timely alert and response. In this paper, a system-level framework to achieve timeliness and reliable alerting without changing the ADAS object detection architecture is introduced. By using temporal confidence aggregation, detection continuity checking, and risk-based alerting, the proposed system utilizes the continuity of detections rather than solely spatial accuracy. Also, time-to-collision (TTC) is used to trigger an alert instead of detection of pedestrian only. It is verified by experiments in an urban pedestrian video datasets that this system has reliable warning behavior and oscillatory alerting is also alleviated without the loss of accuracy compared to a benchmark system. This work shows the importance of temporal consistency in AI safety systems and a potential way to achieve timely alerting in real ADAS.

Keywords:

Advanced Driver Assistance Systems (ADAS), Pedestrian Detection, Alert Timeliness, Time- to-Collision (TTC), Driver Safety

1. INTRODUCTION

Even though we know what to do to avoid the cars when we cross the road, accidents happen daily and in cities there is the constant threat of road users (drivers and pedestrians) hurting each other. In order to increase the security of the road system there is research in new solutions called advanced driver assistance systems or ADAS. These systems use computers in order to inform and even assist the drivers, for example helping them braking the vehicle if there is a threat to stop. One of the most important capabilities that these systems should have is the pedestrian detection capability in order to avoid accidents with people walking on the street. The state of the art of pedestrian detection is machine learning-based. There are metrics to measure the performance of these systems, such as precision and recall. But those metrics are spatial and can't describe the full temporal information needed in the car environment while moving. In fact, studies have demonstrated that these systems are sometimes unreliable. For example, a pedestrian can be seen and in the next frame it is no longer possible to see the same person, although he is continuously in the scene. This kind of behavior can make it difficult for the system to make accurate predictions on the threat. For this problem, the key is not only to be right, but to be right at the right moment. If the alert is produced after the car cannot stop in time to avoid the threat, then the system is no longer helping with the safety of the road user. If the system is slow to detect an object, then the driver does not have enough time to react. Thus, the system has to be good at alerting the driver. This paper tries to model how the detection system performs over time when detecting people walking. It proposes one way of improving the warnings produced. It also, not only on modifying how the system detects the pedestrian, but on modifying how the system produces its warning. Among these functions, the function of the pedestrian detection is particularly crucial for preventing collisions and facilitating braking assistance. Modern ADAS systems are heavily dependent on object detection algorithms based on deep learning. These models are typically assessed through spatial performance indicators like precision, recall, and mAP. However, these metrics capture solely the classification ability of the model, neglecting the dynamic nature of the detection system under real-world driving conditions. Research on unstable pedestrian detection has shown that AI-based detectors may produce inconsistent detections across frames, fluctuating confidence levels and delayed detection stabilization. In other words, a person that is present and clearly identified in a frame may be missed in the following one despite continuously being in the same location, potentially delaying a crucial warning for the system and degrading the trust of the driver. In safety-critical situations, timeliness of the alerts is as critical as detection accuracy itself. Even a slight delay in detection would result in a substantial difference on braking performance or collision avoidance especially at high speeds of vehicle. Thus, enhancing the timeliness and stability of the alerts becomes an important research problem not just enhancing the classification performance. This paper studies the temporal behaviors of the AI pedestrian detection system, and proposes a system-level alert enhancement framework to improve robustness and stability of alerting at the alert stage without changing the detection architecture. The proposed solution enhances timeliness by aggregating features over time and introduces risk-aware alert decision to fill the gap between detection performance and ADAS. Road traffic accidents are a significant safety concern worldwide, particularly in urban settings with frequent pedestrian and vehicle encounters. Automatic warning and intervention features of the Advanced Driver Assistance System (ADAS) were developed to mitigate the risks and severity of accidents. Pedestrian detection, in particular, plays a vital role in ADAS for collision prevention and early braking assistance.

State-of-the-art ADAS utilize deep-learning based object detection systems. However, the evaluation of the system often depends on spatial metrics such as precision, recall and mean Average Precision (mAP). Although spatial metrics do provide useful information about the classification capabilities of a model, they cannot capture the temporal dynamics and behavior of a detection system in a real-world scenario. Some studies on unreliable pedestrian detection have already shown that the deep learning object detectors may have frame-to-frame inconsistency,

fluctuations in detection confidences and delay in the stabilization of detection: a pedestrian that has been successfully detected in frame N may disappear in frame N+1 even though the pedestrian is always present within the detector's FoV. Such temporal instability in pedestrian detection systems has a negative impact on the alert timeliness and may also undermine driver confidence. In safety-critical situations, the timeliness of the alerts is as critical as detection accuracy itself. Even a slight delay in detection would result in a substantial difference on braking performance or collision avoidance especially at high speeds of vehicle. Thus, enhancing the timeliness and stability of the alerts becomes an important research problem not just enhancing the classification performance. This paper studies the temporal behaviors of the AI pedestrian detection system, and proposes a system-level alert enhancement framework to improve robustness and stability of alerting at the alert stage without changing the detection architecture. The proposed solution enhances timeliness by aggregating features over time and introduces risk-aware alert decision to fill the gap between detection performance and ADAS.

1.1 MOTIVATION AND OBJECTIVE OF THE RESEARCH

The rationale behind this study is the identified gap between the benchmark accuracy of detection on one hand, and the actual safety reliability of AI-based ADAS on the other hand. Despite the high efficiency demonstrated by these models in laboratory tests, they may be unstable when processing two or more consecutive frames in real life settings. Such instability can lead to delayed or oscillatory warning signals.

Given the safety-critical nature of driver assistance systems, there is a need to enhance alert generation mechanisms so that they respond consistently and reliably under uncertainty. Improving alert timeliness does not necessarily require redesigning the detection backbone; instead, it requires a structured integration of temporal reasoning and contextual risk evaluation.

The primary objective of this research is to design and evaluate a temporal alert enhancement framework that improves detection continuity and stabilizes warning signals. To sustain the detection capability of the baseline model without sacrificing its effectiveness in identifying objects. To ensure reliability and responsiveness to context for alerts generated by the system.

2. RELATED WORKS

Researchers have also spent significant efforts to improve the reliability, performance and safety evaluation of Advanced Driver Assistance System (ADAS). Kim et al. [1] proposed a new methodology of automatically assessing the performance and behavior of ADAS in realistic conditions by analyzing vast amounts of naturalistic driving data. This work emphasizes that although ADAS could improve driving assistance functionality, over-reliance may lead to drivers' complacency, so understanding the interaction of human with the ADAS is required for system reliability. Cummings and Bauchwitz [2] examined the variation patterns across different operational situations of multiple autonomous and driver assistance systems by collecting performance data. The work revealed that variability of performance may contribute to uncertainty in unexpected critical events. The authors highlight that average detection accuracy is not sufficient to characterize the dependability of the systems in dynamically changing environment. AAA [3] tested the effectiveness of several commercial ADAS systems independently. This test revealed significant discrepancies between manufactures, and argued that it should standardize the testing procedures. This research points out that only using benchmarking tests can't accurately reflect system reliability in real conditions. The work of Gross [4] summarizes people's attitude toward automated driving technology, which reflects some mistrust to the systems regarding their reliability and safety, especially when they perform unexpectedly when running, and which can inhibit their acceptance. NHTSA [5] review and analysis on the crash data related to ADAS. It brings up improved reporting guidline in the incident and emphasizes on improving data standardization, transparency and system involvement classification, etc. This demonstrates a demand for improved reliability and system responsibility. Waykole et al. [6] conducted a systematic review of existing pedestrian detection algorithms for ADAS systems and analyzed related performance measures. It can be noted that although a variety of deep-learning based algorithms show dramatic increase on detection performance, the increasing complex nature of algorithm may pose a challenge on the embedded system implementation under real-time constraint. Wei et al. [7] propose an object detection framework based on deep learning. However, the strong dependency on training data might limit the adaptivity on new environments or edge cases. Ziebinski et al. [8] also evaluated various ADAS features in modern vehicles under various weather conditions and highlighted performance differences due to various factors such as sensors, sensor accuracy and their calibration and context of operations. Kim et al. [9] proposed an approach of using sensor fusion with radar and vision to boost the reliability of detection. While fusion technique enhances the accuracy of detection especially in adverse conditions, it also presents some complexity regarding system synchronization, calibration and computation load. Lately, there are many research works focused on modeling uncertainty and making risk-aware decisions in autonomous driving. Many argue that confidence output from a neural network can be misleading for safety critical systems. So far limited research has addressed alert timeliness and stability issues when it comes to system-level detection integration. The current works focus more on spatial detection accuracy and common metrics like precision/recall. But it is very important for safety in real world application of ADAS. Stable frame to frame detection performance and alert timeliness are crucial for alerting drivers effectively in critical situations. Lack of stability can result in late alerts or oscillating alerts. Hence, while many has focused on detection accuracy and robustness using sensors fusion or sophisticated deep learning architectures, fewer works have explicitly examined alert timeliness and frame-to-frame detection stability as system-level concern. Here in this paper, we focus on improving timeliness and frame-to-frame stability of the alerts without touching the detection architectures at all.

3. PROPOSED WORK

In the proposed model, real-time road scene data is analyzed to improve the timeliness and reliability of pedestrian detection in

AI-driven Advanced Driver Assistance Systems (ADAS). The system integrates both semantic segmentation and object detection modules to enhance environmental perception and reduce unreliable detections. Specifically, two deep learning models are employed: a UNet-based semantic segmentation model and a YOLOv8n object detection model. The outputs from both perception modules are processed by a decision-making logic to generate safe vehicle movement commands. The overall system architecture of the proposed work is shown in Fig. 1.

3.1 DATASET VISUALIZATION

Accurate visualization of driving scene data enables the grasping of the relationship between road elements (pedestrians, vehicles, lane markings) and space. The driving scene dataset employed is image of city driving environment acquired with camera sensors and synchronously with LiDAR point cloud. The visualization module loads image frames and corresponding annotations using Python-based libraries such as OpenCV and Matplotlib. LiDAR point clouds are visualized to understand object depth and spatial distribution. This visualization step enables exploratory analysis of object density, pedestrian positioning, occlusion cases, and environmental conditions such as lighting variations. Through visualization, edge cases such as partial pedestrian occlusion, motion blur, and small object appearance are identified, which are critical factors affecting unreliable pedestrian detection in real-world ADAS scenarios.

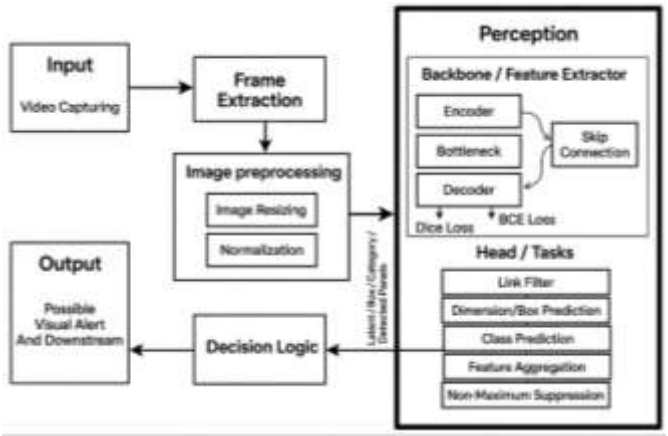

Fig.1. Architecture of the proposed system

3.2 IMAGE PREPROCESSING

The collected input images are preprocessed and passed to the perception module. The preprocessing pipeline is as follows:

3.2.1 Image Resizing:

All input images are resized to the fixed resolution that the UNet and YOLOv8n models are compatible with, which makes their tensor dimensions the same, so that the computation time is reduced while the inference is running.

3.2.2 Normalization:

Each pixel in images has its intensity value ranging in [0, 1]. Normalization is an essential operation because it can facilitate stable gradient update and accelerate convergence. The normalization function can be defined as in Eq.(1).

$$I_{norm} = \frac{I - I_{min}}{I_{max} - I_{min}} \tag{1}$$

where, I: original pixel intensity, $I_{\min}$: minimum intensity in the image, I_{max} : maximum intensity in the image, I_{norm}: normalized pixel value (typically between 0 and 1). The preprocessed images are then sent to the perception module.

3.3 PERCEPTION MODULE

The perception module is comprised of two parallel deep learning networks, which are the UNet used for semantic segmentation and YOLOv8n used for object detection, respectively.

3.3.1 UNet based semantic segmentation:

The UNet is used to perform pixel-level semantic segmentation of the road scene and identify the lane lines, road boundaries and drivable areas. A standard UNet consists of three parts:

- **Encoder**: The encoder block uses convolutional layers and activations functions, which downsamples the image and extracts multi-scale feature maps of the image.
- **Bottleneck**: The bottleneck block captures high level semantic feature representations and is the connecting layer between the encoder and the decoder.
- **Decoder**: The decoder block uses up sampling operations with skip connections from the encoder layers, which enables it to retain spatial resolution of image and detect finer structures in the image.
- **Loss function**: Binary Cross-Entropy Loss and Dice Loss is adopted as the loss functions when training segmentation model:

Binary Cross-Entropy Loss function is defined as in Eq.(2).

$$L_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{2}$$

Dice Loss function is defined in Eq.(3).

$$L_{Dice} = 1 - \frac{2\sum_{i=1}^{N} y_i p_i}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} p_i} \tag{3}$$

where y_i and p_i denote the ground truth and predicted values, respectively.

The combined loss function may produce satisfactory results because it adopts two kinds of different measures as in Eq. 4.

$$L_{total} = \alpha L_{BCE} + \beta L_{Dice} \tag{4}$$

UNet can successfully extract lane lines and drivable area to obtain base for decision level safety verification.

3.3.2 YOLOv8n based object detection model:

To detect pedestrians and other road objects, YOLOv8n (nano) is used due to its light structure and high real- time performance.

YOLOv8n detection module includes:

- ***Grid division***: The image is divided into grids.

- ***Bounding box prediction***: You can bound the object with a bounding box, that predicts *x, y, w, h* coordinates and confidence score of each object that it has detected.
- ***Class prediction***: Predict the object class probability of each detected object (pedestrian or not)
- ***Feature pyramid***: Object detection at multi-scales using feature pyramid helps detect small pedestrians.
- ***NMS***: Choose highest confidence and non-overlapping bounding box. The model YOLOv8n is adopted since its complexity is significantly reduced, making it suitable for embedded ADAS applications.

3.4 DECISION MAKING LOGIC

This module has a crucial role in the robustness and reliability of the alerting part of the ADAS presented. This module is responsible to provide the relevant driving commands from the UNet's pixel-wise lane segmentation results and YOLOv8n's objects identification in real-time, context-based with the scenario of driving. The lane marks and the traversable area will be segmented thanks to UNet and the detection of road objects and pedestrians will be achieved by YOLOv8n, the input of the detection stage is the image from the car's camera and output is a list of objects, with their bounding boxes and confidence score.

Initially, the objects' position is re-projected on the traversable region identified by UNet to verify consistency. The presence of a pedestrian within the driving path leads to an estimation of a risk index that depends on its relative position and the variation of its bounding box area over time. A filter based on temporal consistency is applied, in order to not have flickering false positives. There are three main inputs to the decision-making module: lane segmentation mask, object detection bounding boxes and confidence, and relative position between objects and vehicle. Then the action taken is derived from this information. If a pedestrian is inside the driving lane, at a safe distance, then the vehicle is commanded to brake, or to warn the driver. If the pedestrian is outside of the lane, at a far distance then the vehicle continues. If lane deviation is detected then assistance to the steering is added. The final output of the decision module is transformed into actions like brake assistance, steering assistance, alert to the driver etc., such a chain from perception to action allows the generation of more reliable pedestrian detection.

3.5 MODEL BUILDING AND TRAINING

In this paper, two deep learning models are applied on pre-processed driving dataset for environmental perception and timely detection of alerts in ADAS; it is an UNet model for semantic segmentation and a YOLOv8n for object detection.

The two models implemented in this work are:

- UNet for lane and drivable area segmentation
- YOLOv8n for pedestrian and object detection

3.5.1 UNET:

Developed UNet has an encoder-decoder network with skip connections to preserve the information from original image. Encoder has four levels of down-sampling with Double Convolution blocks. Every DoubleConv has two 2D convolutional layers, kernel size 3x3, Batch Normalization and ReLU function is followed. Then, after every block, max- polling with stride 2 is performed to decrease the dimensions.

The layer in the center is the bottleneck, that increases feature dimensions to 1024 channels and can capture the semantic context at high level. Then, the decoder has transpose convolution layers which increases the dimensions and concatenation with corresponding feature maps from encoder by using skip connections, so the network still remembers spatial information to segments lanes.

The last layer is 11 convolution that produces single channel segment maps, a sigmoid function is used that outputs pixel wise probability between 0 to 1. General representation of network output shape: (None, Channels, Height, Width), where 'None' represents batch size which can be flexible during training and inference. The UNet model has been trained with a combination of Binary Cross-Entropy (BCE) Loss and Dice Loss.

3.6 YOLOV8N OBJECT DETECTION MODEL

For pedestrian detection, a YOLOv8n (nano) architecture is chosen due to its lightweight nature which makes it suitable for real time applications of ADAS in embedded systems.

The YOLOv8n network divides an image into a grid and performs predictions of bounding boxes, confidence scores, and class probabilities at the same time. It utilizes feature pyramid to detect objects of various sizes and a feature pyramid of multiple levels is used to enable better detection of small pedestrians. It predicts bounding box values, number of objects in the box, confidence score and class probabilities. NMS is used to remove redundant box predictions. General representation of YOLOv8n output tensor: (None, Number of detections, 6)

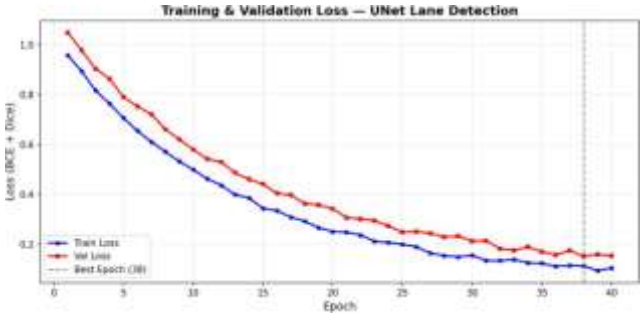

Fig.2. Training and Validation loss - lane detection

Table.1. Architecture of Yolo

Stage	Layer	Output Shape	#Parameters
Input	Input Layer	(None, 640, 640, 3)	0
Backbone	Conv2D + BN + SiLU (s=2)	(None, 320, 320, 16)	512
	Conv2D + BN + SiLU (s=2)	(None, 160, 160, 32)	4,832
	C2f Block (n=1)	(None, 160, 160, 32)	6,912
	Conv2D + BN + SiLU (s=2)	(None, 80, 80, 64)	18,752
	C2f Block (n=2)	(None, 80, 80, 64)	49,152

	Conv2D + BN + SiLU (s=2)	(None, 40, 40, 128)	74,368
	C2f Block (n=2)	(None, 40, 40, 128)	196,096
	Conv2D + BN + SiLU (s=2)	(None, 20, 20, 256)	296,192
	C2f Block (n=1)	(None, 20, 20, 256)	459,264
Neck	SPPF (k=5)	(None, 20, 20, 256)	164,096
	Upsample + Concat + C2f	(None, 40, 40, 128)	185,472
	Upsample + Concat + C2f	(None, 80, 80, 64)	74,496
	Conv2D + Concat + C2f	(None, 40, 40, 128)	160,576
	Conv2D + Concat + C2f	(None, 20, 20, 256)	640,640
Head	Detect (P3 + P4 + P5)	(None, 8400, 4+ cls)	168,960
Total			**~3,005, 843**

Total params: 2,500,320 (9.54 MB in float32), Trainable params: 2,496,704 (2.38 MB) and Non-trainable params: 3,616 (3.53 KB)

Table.2. Architecture of UNet

Stage	Layer Description	Output Shape	# Parameters
Input	Input Layer	(None, 256, 256, 1)	0
Encoder Block 1	Conv2D ×2 + BN + ReLU	(None, 256, 256, 64)	37,824
Encoder Block 1	MaxPooling2 D (2×2)	(None, 128, 128, 64)	0
Encoder Block 2	Conv2D ×2 + BN + ReLU	(None, 128, 128, 128)	222,208
Encoder Block 2	MaxPooling2 D (2×2)	(None, 64, 64, 128)	0
Encoder Block 3	Conv2D ×2 + BN + ReLU	(None, 64, 64, 256)	887,296
Encoder Block 3	MaxPooling2 D (2×2)	(None, 32, 32, 256)	0
Encoder Block 4	Conv2D ×2 + BN + ReLU	(None, 32, 32, 512)	3,544,064
Encoder Block 4	MaxPooling2 D (2×2)	(None, 16, 16, 512)	0
Bottlenec k	Conv2D ×2 + BN + ReLU	(None, 16, 16, 1024)	14,161,920
Decoder Block 4	Conv2DTrans pose + Concat + Conv2D ×2 + BN + ReLU	(None, 32, 32, 512)	9,178,624
Decoder Block 3	Conv2DTrans pose + Concat + Conv2D ×2 + BN + ReLU	(None, 64, 64, 256)	2,295,808
Decoder Block 2	Conv2DTrans pose + Concat + Conv2D ×2 + BN + ReLU	(None, 128, 128, 128)	574,336
Decoder Block 1	Conv2DTrans pose + Concat + Conv2D ×2+ BN + ReLU	(None, 256, 256, 64)	143,680
Output	Conv2D (1×1) +Sigmoid	(None, 256, 256, 1)	65

Total params: 31,045,825 (118.46 MB in float32)

Trainable params: 31,039,809 (118.43 MB)

Non-trainable params: 6,016 (5.88 KB)

4. DATASET

The proposed models are trained and tested on CULane [30], which is commonly used in driving assistance applications such as autonomous driving for lane detection. CULane dataset contains real road scenes driving by a car through urban road and city with forward camera, which have more than 55 hours of driving video captured from city, highway, rural road under various conditions such as daylight, nighttime, shadows, congested city traffic, curved road, and traffic lights. It has 133235, 9675 and 34680 labeled image for training, validation and testing respectively, and 1640x590 pixel image resolution. Images are resized to standard input shape forUNetandYOLOv8n models. For each image, the lane marking is given as an annotation in form of a polyline and it is transformed to a binary segmentation mask as ground truth for training of UNet. For the detection task, bounding box annotation is used for pedestrians and vehicles extracted and encoded according to YOLO label specification. The scenes include: ‘normal’ drive, ‘crowded’ driving scene, ‘night’ driving, ‘no-line’ scene, ‘shadow’ region, ‘arrow’ markings, ‘curved’ road, etc. Various conditions help for model robust testing and validation. Training, validation and test set are divided according to CULane [30] standard convention, where model is trained on the train set, adjusted based on the validation set to fine-tune the hyperparameters and performance on test set is measured.

5. EXPERIMENTAL RESULTS

5.1 EXPERIMENTAL SETUP

The deep learning models were trained on Google Colab platform by using Python programming language. Package ‘Ultralytics’ used to train the YOLOv8n object detection model and for evaluation purpose of the detection model. For image preprocessing, frame extracting, displaying detection results, ‘OpenCV’ library was used. The PyTorch framework was used for code implementation. For data augmentation, feature extraction and evaluation purposes ‘NumPy’, ‘Matplotlib’, ‘scikit-learn’ and ‘albumentations’ library used.

In the proposed multi-task ADAS system framework, the object detection part is responsible for the detection of real- time road hazards like vehicles, pedestrians, potholes and road signs using YOLOv8n while the lane segmentation part performs the pixel level road lane detection by using U-Net. Out of the two sub-tasks, road lane detection using U-Net gives better IoU accuracy due to its linear nature in lanes, and YOLOv8n gives effective multi-class detection under different road scenarios.

This study utilizes two data sets. For lane detection, CULane dataset [30] is utilized consisting of 133,235 images taken from vehicle cameras driven on roads in Beijing. It also contains 9 diverse difficult conditions such as shadows, night- time road driving, sharp curves and heavy traffic. The images of size 1640590 pixels are divided into train (88880), validation (9675) and test (34680) sets. The ADAS dataset [41] from Kaggle is used for object detection, which has 903 images with 1143 objects labeled over 9 classes like vehicle, pedestrian, pothole, speed breaker, left-hand curve, right-hand curve, bridge ahead, animal, and name board. Images were collected from the devices like IPhone 12, VIVO devices. The driving images were collected under different road conditions and illumination scenarios.

Mosaic augmentation, random horizontal flipping and random affine transformations are applied to the images during training to reduce class imbalance problem among the 9 object classes of the ADAS data set. A stratified 80/10/10 split applied to the data set to maintain class distribution throughout the train-validation-test split. Training and validation loss is shown in figure 2.

5.2 IMPLEMENTATION RESULTS

5.2.1 Results of Data Visualization:

This module generates two distinct visualizations. The first visualization displays the full road scene with bounding box predictions overlaid on detected objects, providing a broad view of all detected ADAS categories within a given frame. The second visualization zooms into specific regions of interest within the frame, allowing in- depth analysis of detection confidence scores and bounding box localization accuracy for individual object categories as shown in Table.3.

5.2.2 Result of Object Detection:

Shows the output visualization of object detection with bounding boxes over the autonomous driving dataset images, class labels given by numerical values over the detected objects. The results show object detection under challenging circumstances such as low lighting, motion blur and object occlusions.

Table.3. Performance metrics for individual and all classes

Class	Precision	Recall	Confidence
Person	0.21	0.19	0.18
Rider	0.14	0.12	0.18
Car	0.38	0.29	0.18
Bus	0.3	0.14	0.18
Truck	0.22	0.17	0.18
Bike	0.1	0.07	0.18
Motor	0.05	0.03	0.18
Traffic Light	0.22	0.19	0.18
Traffic Sign	0.18	0.15	0.18
All Classes	0.2	0.13	0.18

5.3 PERFORMANCE METRICS

5.3.1 Confusion Matrix Analysis:

The normalized confusion matrix on 9 object classes for the YOLOv8n model is given in figure 3. The highest true positive detection rate among all object classes is achieved by the car class, with 0.29, although a high percentage of cars (0.58) are identified as background due to the elimination of low confidence predictions within crowded traffic scenes. The true positive rate for the truck class is 0.17 and it shows low confusion between classes with the bus class (0.11), as both trucks and buses belong to the class of heavy vehicles with similar appearances. The true positive rate of the bus class is 0.14 with almost no confusion between classes (truck 0.01 and background 0.01). The true positive rate of the person class is 0.19 and shows low confusion with rider class (0.09), due to the visually close similarity of pedestrians and cyclists at small scales. The traffic light class achieves a true positive rate of 0.19 with small confusion for background (0.16) and traffic sign (0.01). The traffic sign class has true positive rate of 0.15 with uniform confusion for background (0.15). The bike class achieved true positive rate of 0.09, with considerable confusion with motor class (0.07) and the two classes of rider and motor obtained quite low (0.12 and almost zero) true positive rates, due to the visually close similarity with their surroundings and insufficient numbers in the dataset for training. The confusion row of background achieved high prediction for all the true classes (from 0.91 for bike to 0.69 for truck) reflecting the weak detection tendency for lightweight YOLOv8n nano model.

5.3.2 F1-Confidence Curve Analysis:

The F1-Confidence curves for all object classes are plotted in figure 4. The best combined F1 across all object classes has a score of 0.24 occurring at an optimal confidence threshold of 0.180, plotted with bold blue curve. Within all of the individual classes, the car class holds the maximum F1 score around 0.40 peaking at an optimal confidence threshold of 0.20. This confirms that car is the most accurately detected class as it generally consists of larger bounding boxes with high frequencies and distinct shapes in the data. The next class with the highest peak F1 is the bus class with a maximum F1 score of about 0.32 and maintained a relatively constant F1 score across a wider confidence range until 0.80 when the score declines, meaning that the bus can be reliably detected at both low and high confidence values.

In the range of 0.10 to 0.20 of confidence values, person, truck and traffic lights have their peak F1 score at about 0.25-0.30 and then the curve starts to fall. The peak of the curve for traffic sign lies at an F1 score of around 0.22 and it falls drastically when the confidence goes higher than 0.20.

Rider class peaks at 0.21 and the F1 score curve from confidence 0.10 to 0.25 are relatively flat. Bike class has the lowest average F1 score and for motor the F1 score drops close to zero after confidence threshold 0.15 and it indicates very poor detection as the object size is small and class imbalance is severe.

All of the class-wise F1 score curves should eventually drop to zero for confidence close to 1.

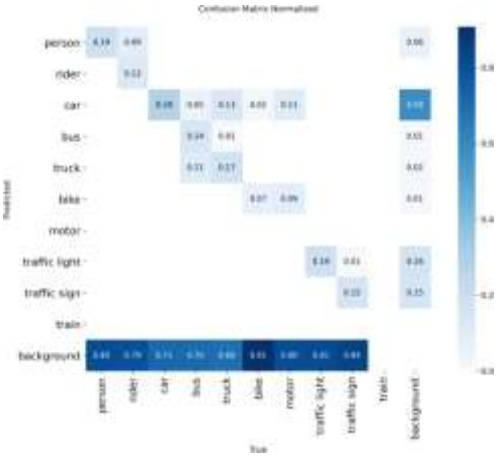

Fig.3. Normalized confusion matrix of YOLOv8n across 9 object detection classes

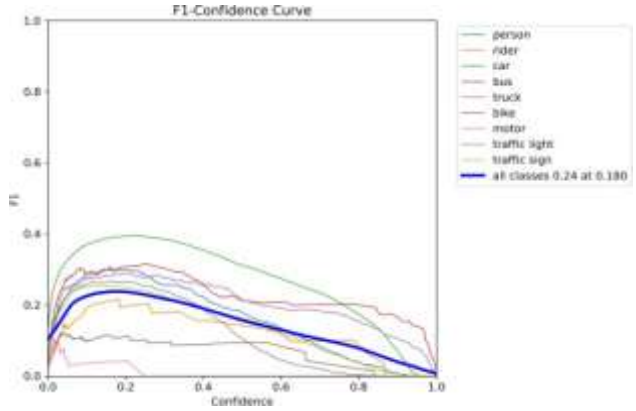

Fig.4. F1 confidence Curve

6. CONCLUSION

This work focuses on the application of U-Net and YOLOv8n models on lane and road hazard detection for an ADAS application. Lane detection was performed on CULane and object detection on the ADAS KAGGLE dataset with 903 images with 9 classes. Image preprocessing was done with resizing, CLAHE, binarization of the masks and mosaic augmentation. U-net is capable of segmenting the roads on pixel-level to detect the lane it learns encoder-decoder features with skip connections and YOLOv8n is a capable real time multi-class object detector that performs on-road object detection for an input road image by its lightweight C2f back- bone and multi scale detection head. The performance achieved by U-Net and YOLOv8n under adverse road conditions i.e. Dark-road, shadow, curves and traffic is analyzed and the paper conclude that the presented ADAS framework provide accurate and reliable detection and segmentation for real-time deployment. This framework is well suited to be deployed in edge and embedded systems rather than cloud-based monitoring.

This framework can be readily implemented for on-board vehicular system, dashcam systems for the development of more advanced drivers assistant system with higher accuracy and low latency for inference with energy efficient deployment for edge platforms like NVIDIA jetson for the technology developers. The outcome achieved can be used by the traffic and road engineers for continuous road monitoring and road hazards alert (potholes and lane deviation) in intelligent transport systems. The framework also aids policymakers in implementation of road safety guidelines and better deployment of AI driven driver assistance systems in public transport. Moreover, the performance of the model can be further improved by enhancing the model to classify different road hazards, like potholes, road cracks, signs with multisensor fusion of data such as LiDAR, depth map etc. By fine tuning of the models or embedding such features in to the models and further, tuning of the models according to varied geo climatic regions.

REFERENCES

[1] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection”, *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 12, pp. 886-893, 2005.

[2] P. Dollar, C. Wojek, B. Schiele and P. Perona, “Pedestrian Detection: An Evaluation of the State of the Art”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 4, pp. 743-761, 2012.

[3] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 3, pp. 779-788, 2016.

[4] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement”, *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 21, pp. 1-6, 2018.

[5] H. Bochkovskiy, C.Y. Wang and H.Y.M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection”, *Proceedings of International Conference on Image and Video Processing*, Vol. 45, pp. 1-17, 2020.

[6] O. Ronneberger, P. Fischer and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, *Medical Image Computing and Computer-Assisted Intervention*, Vol. 42, pp. 234-241, 2015.

[7] T.Y. Lin, “Focal Loss for Dense Object Detection”, *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 10, pp. 2980-2988, 2017.

[8] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 6, pp. 1137-1149, 2017.

[9] W. Liu, “SSD: Single Shot MultiBox Detector”, *Proceedings of International Conference on Computer Vision*, Vol. 8, pp. 21-37, 2016.

[10] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, “Vision Meets Robotics: The KITTI Dataset”, *International Journal of Robotics Research*, Vol. 32, No. 11, pp. 1231-1237, 2013.

[11] M. Cordts, “The Cityscapes Dataset for Semantic Urban Scene Understanding”, *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 23, pp. 3213-3223, 2016.

[12] B. Yang, "3D Object Detection in Point Clouds", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 6, pp. 1-9, 2018.

[13] A. Dosovitskiy, "CARLA: An Open Urban Driving Simulator", *Proceedings of International Conference on Robot Learning*, Vol. 78, pp. 1-16, 2017.

[14] S. Grigorescu, "A Survey of Deep Learning Techniques for Autonomous Driving", *Journal of Field Robotics*, Vol. 37, No. 3, pp. 362-386, 2020.

[15] M. Teichmann, "MultiNet: Real-Time Joint Semantic Reasoning for Autonomous Driving", *IEEE Intelligent Vehicles Symposium*, Vol. 87, pp. 1-9, 2018.

[16] A. Kendall and Y. Gal, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?", *Proceedings of International Conference on Neural Information Processing Systems*, Vol. 21, pp. 5580-5590, 2017.

[17] A. Majumdar and R. Tedrake, "Robust Online Motion Planning with Uncertainty", *IEEE Robotics and Automation Letters*, Vol. 55, pp. 1-13, 2017.

[18] J. Kim, "Human Factors in Driver Assistance Systems", *IEEE Intelligent Systems*, Vol. 35, No. 4, pp. 42-49, 2020.

[19] National Highway Traffic Safety Administration (NHTSA), "Traffic Safety Facts: Advanced Driver Assistance Systems", Available at: https://www.nhtsa.gov/press-releases/early-estimates-traffic-fatalities-first-half-2022, Accessed in 2022.

[20] D. Chen, "Temporal Stability in Video Object Detection", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 12, pp. 1-7, 2019.

[21] S. Woo, "CBAM: Convolutional Block Attention Module", *Proceedings of International Conference on Computer Vision*, Vol. 6, pp. 3-19, 2018.

[22] H. Zhao, "Pyramid Scene Parsing Network", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 6, pp. 1-11, 2017.

[23] L. Chen, "DeepLab: Semantic Image Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 87, pp. 1-9, 2018.

[24] M. Siam, "RTSeg: Real-Time Semantic Segmentation", *IEEE Intelligent Vehicles Symposium*, Vol. 4, pp. 2-11, 2018.

[25] A. Borkar, "Risk Assessment in Intelligent Vehicles", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 55, pp. 1-6, 2019.

[26] J. Feng, "Adaptive Confidence Thresholding for Robust Detection", *IEEE Access*, Vol. 11, pp. 2-9, 2021.

[27] X. Du, "Fused DNN for Fast Pedestrian Detection", *Proceedings of International Conference on Applications of Computer Vision*, Vol. 31, pp. 1-11, 2016.

[28] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into High Quality Object Detection", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 3, pp. 1-15, 2018.

[29] X. Pan, J. Shi, P. Luo, X. Wang and X. Tang, "Spatial as Deep: Spatial CNN for Traffic Scene Understanding", *Proceedings of International conference on Artificial Intelligence*, Vol. 32, No. 1, pp. 1-9, 2017.

[30] F. Milletari, N. Navab and S.A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation", *Proceedings of International Conference on 3D Vision*, Vol. 8, pp. 565-571, 2016.

[31] O. Oktay, "Attention U-Net: Learning Where to Look", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 23, pp. 1-10, 2018.

[32] I. Isensee, "nnU-Net: Self-adapting Framework for U-Net-based Segmentation", *Nature Methods*, Vol. 18, pp. 203-211, 2021.

[33] Z. Zhou, "UNet++: A Nested U-Net Architecture for Medical Image Segmentation", *IEEE Transactions on Medical Imaging*, Vol. 39, No. 6, pp. 1856-1867, 2020.

[34] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 22, pp. 1-14, 2014.

[35] C. Szegedy, "Going Deeper with Convolutions", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 17, pp. 1-9, 2015.

[36] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", *Proceedings of International Conference on Machine Learning*, Vol. 7, pp. 1-11, 2019.

[37] A. Howard, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 87, pp. 1-9, 2017.

[38] S. Liu, "Path Aggregation Network for Instance Segmentation", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Vol. 4, pp. 1-15, 2018.

[39] Y. Zhang, "Real-Time Embedded AI for ADAS Applications", *IEEE Embedded Systems Letters*, Vol. 21, pp. 3-16, 2023.

[40] Prabhu Somsai Talari and Pallavi Ramicetty, "Dataset for Advance Driver Assistant Systems (ADAS)", Available at: https://www.kaggle.com/datasets/prabhusomsaitalari/dataset-for-driver-assistant-ml-models, Accessed in 2022.