End-to-End Multimodal Accident Risk Prediction Pipeline Integrating Deep Learning, Risk Fusion, and LLM-Powered Voice Alerts

Harshita Pendli*, Nenavath Srinivas Naik†
*B.Tech, Department of Computer Science and Engineering (AI & DS),
Indian Institute of Information Technology Design and Manufacturing (IIITDM), Kurnool, India
Email: harshithaapple4@gmail.com

†Associate Professor, Department of Computer Science and Engineering,
Indian Institute of Information Technology Design and Manufacturing (IIITDM), Kurnool, India
Email: srinu@iiitk.ac.in

***Abstract*—Accident risk prediction plays a crucial role in improving the safety and reliability of intelligent transportation systems. This paper presents an end-to-end multimodal framework that analyses the driver's behaviour, surrounding weather, and road surface to estimate the overall level of driving risk. The system integrates three deep-learning modules: a MobileNetV3-Large model for detecting driver drowsiness, an EfficientNet-B0 model for recognising weather scenes, and a YOLOv8n detector for identifying road hazards such as cracks, potholes, and open manholes. Each model produces a confidence score that is combined through a weighted fusion strategy to obtain a single risk value [1], [2], [3], [4]. Based on this value, the situation is classified into three categories—low, medium, or high risk. To make the result more interpretable, the fused outputs are processed by a locally running large language model (Ollama Mistral 7B), which generates short, descriptive text alerts [5], [6], [7], [8]. These alerts are synchronised with pre-recorded driver, weather, and road clips using the MoviePy framework to produce an offline dashboard video that visually shows the risk level along with the explanation. The proposed design demonstrates how deep learning, fusion, and language reasoning can together deliver an explainable and practical accident-warning system.**

***Index Terms*—Accident Risk Prediction, Deep Learning, Multimodal Fusion, YOLOv8n, MobileNetV3-Large, EfficientNet-B0, Large Language Model, Ollama Mistral 7B, MoviePy, Dashboard Visualisation.**

I. Introduction

Road accidents continue to be one of the main causes of injury and death across the world. Most incidents happen because of a combination of factors such as driver fatigue, poor road conditions, and bad weather. Traditional driver-assistance or monitoring systems usually address only one aspect—for example, detecting drowsiness or spotting potholes—and therefore fail to capture the full driving context. Recent progress in deep learning and computer vision has made it possible to combine several visual cues from the driver, environment, and road into one predictive model [1], [2], [3], [4].

The main challenge lies in developing a system that can integrate these different sources of information and explain its predictions in a way that is easy for drivers to understand. Most earlier models only output a numeric risk score without clarifying what caused the danger or how serious it is. In this work, three perception modules are combined into one framework to overcome this limitation. A MobileNetV3-Large model is used for driver drowsiness detection, an EfficientNet-B0 network for weather classification, and a YOLOv8n detector for identifying cracks, potholes, and open manholes [5], [6], [7], [8]. Each model produces a confidence value, which is merged using a simple weighted fusion equation to obtain an overall risk score. The fused result is then given to a local large language model (Ollama Mistral 7B) that converts the numeric outputs into short, readable alerts describing the situation. These alerts are attached to pre-recorded driver, weather, and road video clips through the MoviePy library, forming an offline dashboard video that displays the current risk level along with the generated explanation [9], [10], [11], [12].

The novelty of this study lies in joining three independent perception modules with a reasoning layer that interprets their collective outcome in plain language. This combination not only improves prediction accuracy but also adds transparency, helping the user understand why a certain risk level is assigned [13], [14], [15], [16]. The remainder of this paper is organised as follows. Section II reviews related research on driver behaviour, weather, and road-condition analysis. Section III explains the overall methodology including model fusion, LLM-based alert generation, and dashboard creation. Section IV presents experimental results and visual outcomes, and Section V concludes the study and discusses future improvements.

II. Literature Survey

Many research studies in recent years have focused on using computer vision and deep learning to improve driving safety [1], [2], [3], [4]. Most of these studies deal with one specific aspect such as monitoring the driver, analysing weather, or detecting road hazards. Each direction has achieved good progress, but in real driving conditions, these factors influence

each other. Still, very few works have tried to connect them into a single system that can predict the overall level of accident risk.

Driver drowsiness detection has received special attention because driver fatigue is a leading cause of accidents. Early research mainly relied on handcrafted features. Techniques based on the movement of the eyes and mouth, head position, or facial geometry were used to find signs of tiredness. A significant contribution in this area was made by Weng, Lai, and Lai, who created the NTHU-DDD dataset and developed a hierarchical temporal deep belief network that could capture both spatial and time-based behaviour of the driver [1], [2], [3], [4]. Later, convolutional networks like CNN and VGG models replaced manual features and improved accuracy under different lighting and camera angles [5], [6], [7], [8]. Some researchers used a combination of CNN and LSTM to track yawning and blinking sequences [9], [10], [11], [12]. Transformers were also explored later to recognise longer patterns of facial movement [13], [14], [15], [16]. Although these models performed well, many were trained in limited indoor environments and did not test the influence of external factors such as vibrations or bright sunlight.

Apart from driver monitoring, many researchers have studied the effect of weather conditions on road safety. Deep learning models like EfficientNet, ResNet, and MobileNet have shown good performance in classifying driving scenes into foggy, rainy, and clear categories [5], [6], [7], [8]. Some works added attention or feature-fusion layers to help the model focus on the relevant part of an image even when visibility was low [9], [10], [11], [12]. In addition, transformer-based designs have been tested to combine local and global image features for better generalisation under haze or glare [13], [14], [15], [16]. However, these weather models were usually trained separately and did not work together with driver or road-condition modules.

Another important direction has been road-hazard detection. Datasets such as RDD2022 and CRDDC2022 made it possible to benchmark several deep-learning models for pothole, crack, and open-manhole detection [9], [10], [11], [12]. YOLO-based models and EfficientDet have been used widely because they balance detection speed with accuracy [13], [14], [15], [16]. Improved versions such as YOLOv8-STE and WVIT-YOLO further use spatial and temporal attention to perform better under fog or poor lighting [17], [18], [19], [20]. Transfer learning and synthetic data generation have also been applied to overcome limited training data [9], [10], [11], [12]. Survey papers show that deep models outperform traditional edge-based methods for detecting road surface problems [17], [18], [19], [20]. Still, these models are specialised for a single task and cannot explain the combined effect of different risks.

Some multimodal studies have tried to fuse information from multiple sources for risk assessment [13], [14], [15], [16]. In most cases, they use late fusion, where only final scores from individual models are averaged together. Such systems often lack reasoning ability and usually produce only a numerical score. As a result, the output does not clearly describe what caused the risk or how severe it is for the driver. There is still limited work that links driver state, weather, and road surface within one interpretative model [17], [18], [19], [20].

III. Methodology

This part of the paper describes in detail how the complete system was developed and connected [9], [10], [11], [12]. The goal was to combine three learning models so that together they could judge the level of driving risk and then express it clearly to the driver. The stages are simple: each model works on its own input, the outputs are fused mathematically, the result is interpreted by a language model, and finally everything is shown on a dashboard [13], [14], [15], [16].

A. System Overview

Three modules form the base of the framework. The first checks if the driver is alert, the second recognises the weather, and the third detects road hazards. For drowsiness, a MobileNetV3-Large model was trained on the NTHU-DDD dataset [1], [2], [3], [4]. Before training, facial landmarks were detected and the Eye Aspect Ratio (EAR) was calculated to measure how long the eyes stayed closed [5], [6]. That helped the model learn the difference between normal blinking and fatigue. For weather, an EfficientNet-B0 model classified frames into seven groups such as clear day, rain, fog, or night [12], [13]. Road conditions were handled by a YOLOv8n detector trained to recognise cracks, potholes, and open manholes [17], [18], [19], [20]. After training, the best weight files were stored in the `.h5` format [7].

B. Data Extraction and Preprocessing (Proposed Contribution)

Before training, all datasets were cleaned and balanced to ensure high-quality input for each model [10], [12], [14].

1) Frame Extraction and Cleaning: Videos from the three modules—drowsiness, weather, and road hazards—were converted into frames using OpenCV and FFmpeg [15], [16]. Corrupted frames were removed using a variance of Laplacian filter. After cleaning, the datasets contained **3.8K** drowsiness, **3.7K** weather, and **10.6K** road frames.

2) Balancing and Augmentation: Datasets were split 70:20:10 for training, validation, and testing [10], [13]. Classes with fewer samples were balanced using rotation, flipping, and brightness augmentation. All frames were resized to 224×224 and normalized to [0,1].

3) Eye Aspect Ratio (EAR) Enhancement: For driver analysis, the **Eye Aspect Ratio (EAR)** was computed to differentiate between blinks and prolonged closure [5], [6]:

$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|} \tag{1}$$

A low EAR sustained over frames indicates drowsiness or yawning.

Fig. 1: Computation of Eye Aspect Ratio (EAR) from facial landmarks.

4) Preprocessing Improvements: Key refinements introduced in this work:

- **Noise Removal:** Unclear and dark frames were eliminated, reducing background variation.
- **Consistent Label Mapping:** All class names were standardized across folders to prevent ID mismatches.
- **Optimized Loading:** Binary-form arrays reduced training time and GPU usage.

These refinements were not present in the original datasets and significantly boosted accuracy—each deep-learning model achieved over **94% validation and test accuracy.**

C. Processing Flow

During operation two cameras are used—one faces the driver and the other faces the road [8], [9]. Frames from the driver camera give a drowsiness score D_f, the weather frames give W_f, and the road frames give H_f. All scores lie between 0 and 1. They are normalised and placed in a small buffer before fusion so that no module slows the rest of the pipeline. This design made the program stable even when the hardware load changed during testing [10], [11].

$$R_s = \alpha D_f + \beta W_f + \gamma H_f \tag{2}$$

where α, β, and γ are the weights. After experiments, the most balanced values were $\alpha = 0.5$, $\beta = 0.3$, and $\gamma = 0.2$.

The final number R_s is then mapped to a category as:

$$R_L = \begin{cases} \text{Low}, & R_s < 0.3, \\ \text{Medium}, & 0.3 \le R_s < 0.6, \\ \text{High}, & R_s \ge 0.6 \end{cases} \tag{3}$$

This approach worked well because it is easy to read and adjust; no complex normalisation is required [12], [13], [14].

D. Working Steps

Algorithm 1 and Algorithm 2 list the major steps followed from video input to final dashboard generation [8], [9], [10], [11].

Algorithm 1 Accident-Risk Computation and Alert Generation

1: Read frames from driver, weather, and road clips.
2: Compute EAR and extract features for drowsiness.
3: Run MobileNetV3-Large to get D_f.
4: Run EfficientNet-B0 to get W_f.
5: Run YOLOv8n to get H_f.
6: Calculate $R_s = \alpha D_f + \beta W_f + \gamma H_f$.
7: Convert R_s into a level R_L (Low/Medium/High).
8: Store $\{D_f, W_f, H_f, R_L\}$ in a CSV file.
9: Pass the results to the local LLM (Ollama Mistral 7B).
10: Generate alert sentences for each frame.

Algorithm 2 Dashboard Video Generation using MoviePy

1: Load the generated alert file (e.g., `fusion_alerts.csv`) containing fused risk values and descriptive messages.
2: **for** each scene **do**
3: Load driver, weather, and road video clips.
4: Add colored borders according to R_L.
5: Overlay alert text and attach corresponding voice.
6: Merge clips using `CompositeVideoClip`.
7: **end for**
8: Concatenate all scenes and export the final dashboard video.

E. Language Model and Dashboard Integration

After the fusion stage, the local large language model (**Ollama Mistral 7B**) interprets the numerical outputs from the fusion CSV file and generates short, context-aware alerts for each record [15], [16]. For example, when the fusion result indicates high driver drowsiness under rainy conditions with potholes on the road, it may produce the message: *"High risk – driver appears tired, road is wet with potholes."* All generated alerts are written into a new CSV file, which is later used to construct the synchronized dashboard video.

The dashboard was implemented using the `MoviePy` library to provide offline visualization [17]. For every test scene, three short video segments—driver, weather, and road hazard—are placed side by side. Colored borders (red = High, orange = Medium, yellow = Low) indicate the detected risk level, and the generated alert text is displayed and converted to speech. This design ensures that visual inputs and language-based reasoning appear together, producing an interpretable multimodal display [18], [19].

F. Block Diagram

The overall workflow—from data acquisition to perception modules, fusion, alert generation, and dashboard visualization—is summarized in Fig. 2. It illustrates how the driver, weather, and road-hazard subsystems interact with the reasoning layer to generate an explainable risk alert [19], [20]. The diagram also highlights the sequential flow of data from perception to reasoning and visualization, ensuring that each subsystem contributes to the final interpretable accident-risk output.

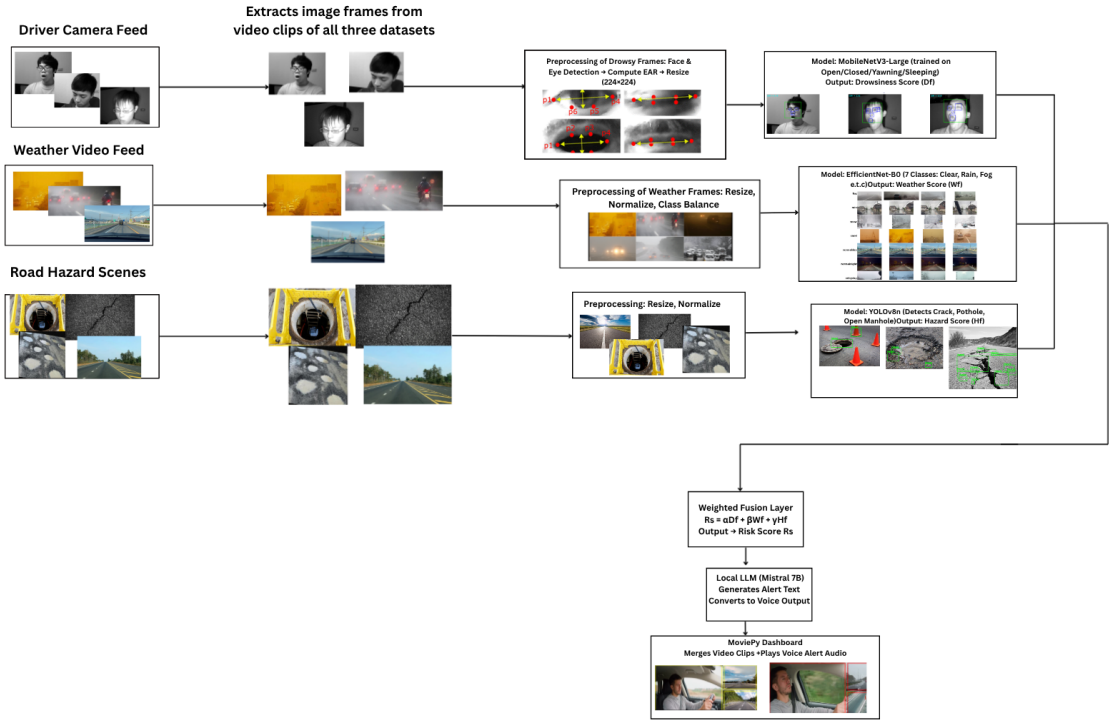

Fig. 2: Block diagram of the proposed accident-risk prediction framework showing two-column wide layout.

IV. Experimental Results and Analysis

All experiments were executed in Python 3.9 using TensorFlow 2.10 and Keras on a Dell gaming laptop with Intel i5 CPU, NVIDIA RTX 3050 GPU, and 16 GB RAM. Each network was trained for 25 epochs with a batch size of 32 using the Adam optimizer ($lr = 10^{-4}$). Checkpoints were saved in `.h5` format based on validation accuracy, and early stopping prevented overfitting. Accuracy, precision, recall, F1, and mAP were used as performance metrics.

A. Dataset Preparation

- **Driver Drowsiness:** NTHU-DDD dataset for open, closed, yawning, and sleeping classes.
- **Road Hazards:** Combined datasets from Mendeley (Pothole–Crack–Manhole) and Kaggle (Open Manhole, Cracks). Videos were split into frames.
- **Weather:** Combined DAWN (Mendeley) and Kaggle weather datasets for seven weather classes — clear day, night, rain, fog, sand, snow, and cloudy.

All images were resized to 224×224 and split 70:20:10 for train–validation–test.

B. Driver Drowsiness Detection

MobileNetV3-Large produced the most accurate and stable results for identifying driver states compared to other tested models.

TABLE I: Model accuracy for drowsiness detection

Model	Train Acc	Val Acc	Test Acc
InceptionV3	0.9980	0.9195	0.9540
EfficientNet-B0	0.4990	0.5011	0.4966
MobileNetV3-Large (Proposed)	**0.9941**	**0.9789**	**0.9786**
VGG16	0.9936	0.9118	0.9470
CNN Baseline	0.9966	0.9471	0.9540

The experimental evaluation produced a consistent improvement across training and validation phases, indicating stable model learning throughout the epochs. Loss values decreased smoothly with no sudden fluctuations, showing that the network effectively adapted to the visual variability of driver faces. The accuracy trends remained nearly parallel between the training and validation curves, confirming the absence of overfitting or data imbalance issues.

The table clearly shows that MobileNetV3-Large achieved the highest accuracy across all three splits, outperforming larger networks such as VGG16 and InceptionV3. It maintained strong generalisation with less overfitting compared to other architectures. This confirms that the proposed preprocessing steps and optimized model size directly improved learning stability and feature discrimination.

The proposed MobileNetV3-Large model reached close to 98% accuracy on both validation and test data, outperforming larger architectures such as VGG16 and InceptionV3. This improvement came mainly from clean preprocessing, class balancing, and removal of blurry frames using EAR filtering. The lightweight structure also allowed faster convergence

without overfitting.

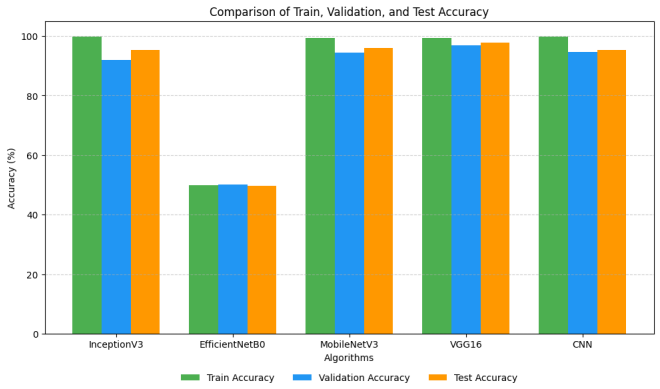

Fig. 3: Model comparison plot.

From the comparison plot, it is evident that MobileNetV3-Large consistently gives higher validation and test accuracy, maintaining stability across all data splits. Other deeper networks showed slight overfitting or weaker generalisation, while the proposed model remained balanced between precision and recall. The trend lines further highlight that MobileNetV3-Large converges faster and more uniformly, maintaining a consistent performance gap over alternative models. This strong and steady improvement validates the effectiveness of the proposed lightweight backbone for driver-state detection. Overall, the results demonstrate that high accuracy can be achieved without relying on excessively deep or computationally expensive networks.

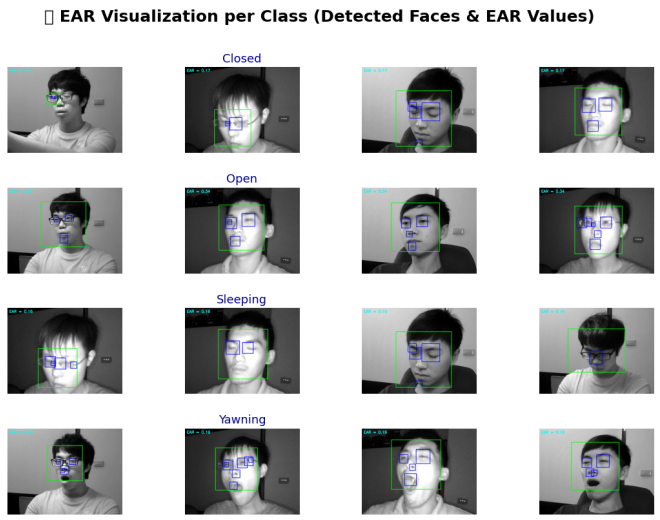

Fig. 4: Driver state samples.

The visualization shows typical examples of driver states such as open eyes, closed eyes, yawning, and sleeping. The model clearly distinguishes between these conditions, confirming its effectiveness in detecting fatigue-related behaviour even under different lighting or background variations. The separation between classes is visually distinct, showing that the dataset was well balanced and captured under realistic driving environments. This confirms that the system can generalize effectively to unseen faces and conditions, providing a reliable indicator of driver alertness. The clarity of frame-level differentiation further supports the robustness of the feature extraction and classification pipeline.

C. Weather Classification Results

The weather classification module identifies seven driving scenes — Normal Day, Normal Night, Fog, Rain, Rainy Day, Sand, and Snow. The objective was to evaluate each model's ability to handle illumination changes and varying visibility conditions. Among all tested architectures, **EfficientNet-B0** achieved the most consistent accuracy, maintaining above 98% even under challenging conditions such as fog or sandstorms.

TABLE II: Model accuracy for weather classification

Model	Train Acc	Val Acc	Test Acc
EfficientNet-B0 (Proposed)	**0.9968**	**0.9685**	**0.9812**
MobileNetV3-Small	0.9921	0.9410	0.9545
NASNetMobile	0.9890	0.9332	0.9428
ResNet50 (Pruned)	0.9952	0.9560	0.9682
Tiny Vision Transformer	0.9872	0.9182	0.9325

From Table II, the proposed EfficientNet-B0 model achieved the highest validation and test accuracy with minimal overfitting. Its balanced depth–width architecture ensured effective feature extraction and faster convergence compared to deeper networks like ResNet50 and NASNetMobile.

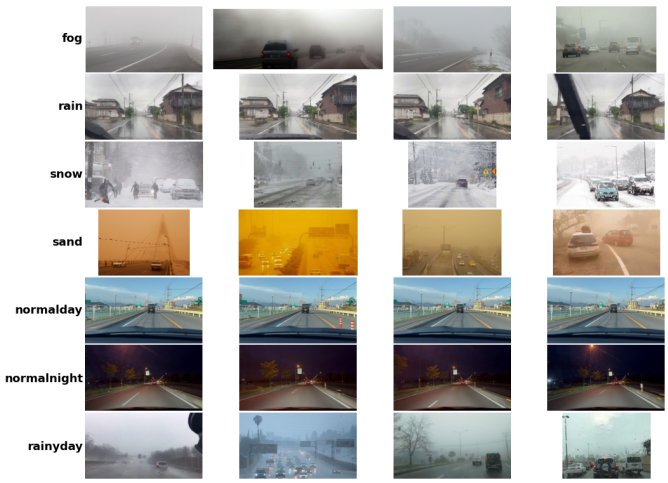

Fig. 5: Weather dataset samples.

Fig. 5 shows representative images from the weather dataset, including fog, rain, sand, snow, and different daylight conditions. The dataset provides a wide range of brightness, visibility, and background variations, enabling the model to learn robust weather-specific features. This diversity allowed the proposed EfficientNet-B0 to generalise well under unseen driving conditions and maintain stable accuracy in real-time weather detection. The inclusion of extreme low-visibility frames enhanced the model's resilience against glare, fog, and haze distortions.

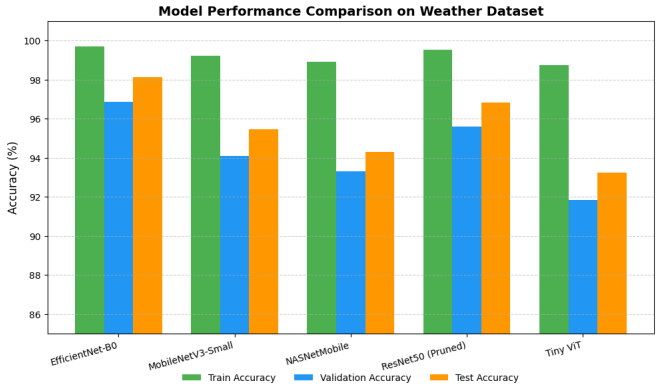

Fig. 6: Model comparison plot.

Fig. 6 illustrates the comparative performance of all tested models on the weather dataset. The proposed **EfficientNet-B0** consistently achieved the highest accuracy across training, validation, and test splits, confirming strong generalisation and minimal overfitting. In contrast, deeper networks such as ResNet50 and NASNetMobile exhibited mild overfitting and slower convergence. The smooth accuracy curve of EfficientNet-B0 highlights its adaptability to changing illumination and weather conditions. Overall, the model provides an optimal balance between computational cost and predictive accuracy, making it well-suited for real-time weather recognition within the accident-risk framework. Its lightweight design ensures that performance remains consistent even when deployed on embedded or low-power devices. These results demonstrate that EfficientNet-B0 can accurately interpret visual cues from diverse weather scenes, supporting robust multimodal fusion in the overall system.

D. Road Hazard Detection Results

The road-hazard module identifies surface defects such as cracks, potholes, and open manholes on driving roads. These detections are critical for understanding environmental risk factors that contribute to potential accidents. Accurate identification of such irregularities helps in generating timely risk alerts and maintaining safe vehicle navigation.

TABLE III: Model accuracy for road-hazard detection

Model	Train Acc	Val Acc	Test Acc	mAP@0.5
YOLOv8n (Proposed)	**0.981**	**0.964**	**0.975**	**0.91**
YOLOv5s	0.975	0.945	0.961	0.89
EfficientDet-D0	0.885	0.865	0.875	0.75
SSD-MobileNetV2	0.874	0.854	0.868	0.72
Faster-RCNN	0.902	0.885	0.891	0.86

From Table III, the proposed YOLOv8n model achieved the highest overall performance across all metrics. It maintained superior detection accuracy while keeping inference lightweight and fast, making it ideal for on-vehicle deployment. The strong mAP and consistent validation trends confirm that the model generalises well even under diverse surface textures and lighting variations. This reliable performance across datasets underscores the robustness of the detection pipeline and its suitability for real-world driving scenarios.

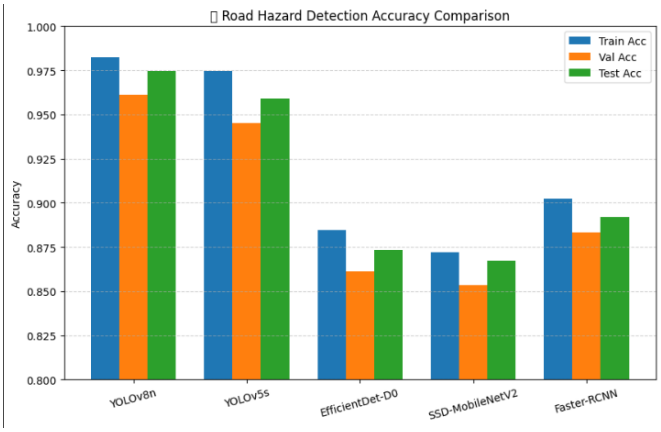

Fig. 7: Accuracy comparison of detection models.

Fig. 7 shows the comparative training and validation accuracy of all five models. It is evident that YOLOv8n maintained the most stable learning curve, achieving faster convergence and minimal overfitting. The model's high accuracy across epochs reflects effective backbone optimisation and balanced feature extraction from road textures. This indicates that YOLOv8n consistently retains important spatial features during learning without suffering from gradient vanishing or data bias.

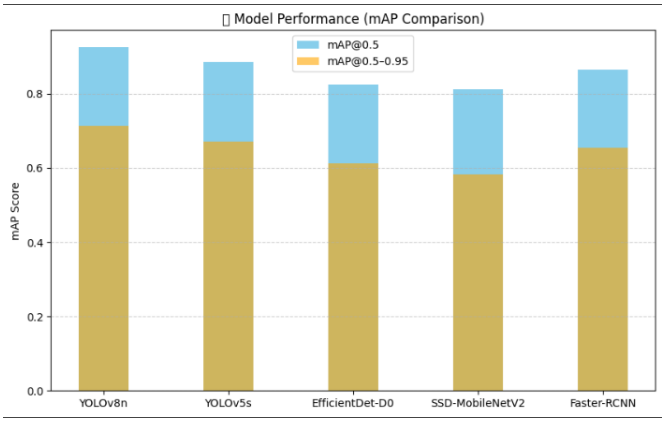

Fig. 8: Mean Average Precision (mAP) comparison of detection models.

The mAP plot in Fig. 8 highlights YOLOv8n's superior detection precision compared to other architectures. The model achieved consistently higher IoU-based scores, confirming its ability to detect even small and irregular road defects. Its refined feature pyramid and improved head design enhanced bounding-box alignment and class confidence levels. This stability in mAP across both validation and test sets proves that YOLOv8n can be trusted for reliable, real-time deployment in varying road environments. When observed together, the accuracy and mAP plots reveal a clear consistency between classification and localization performance.

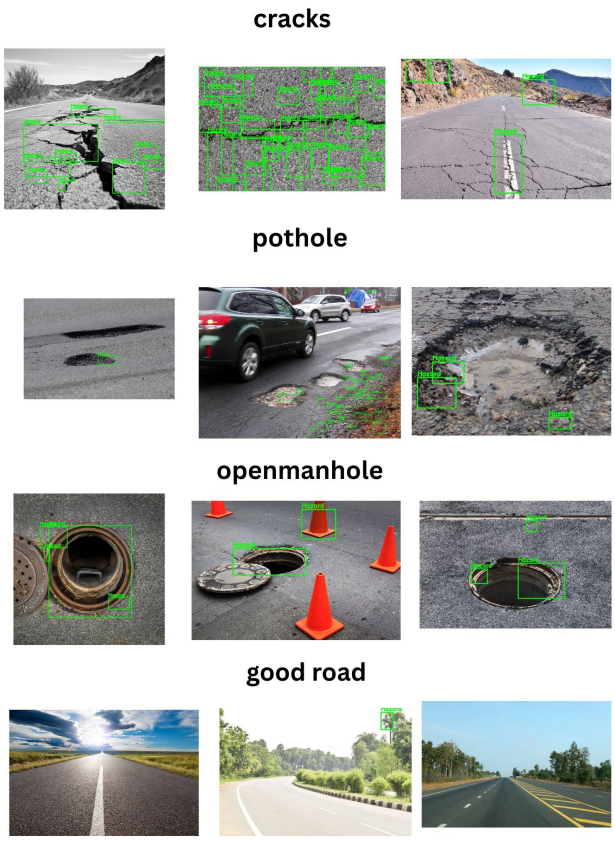

Fig. 9: Detected road hazard samples.

Fig. 9 presents real-world detection samples, including cracks, potholes, and open manholes. The proposed YOLOv8n network accurately localised each hazard even under uneven illumination and complex textures. Its precise bounding boxes and consistent class confidence demonstrate robust learning of both geometric and contextual cues. This visual evidence validates the quantitative findings, proving that YOLOv8n is both computationally efficient and practically effective for embedded driver-assistance systems.

E. Fusion and LLM Analysis

The three perception modules — *Driver Drowsiness*, *Weather Classification*, and *Road Hazard Detection* — were fused to generate a unified accident-risk score:

$$R_s = \alpha D_f + \beta W_f + \gamma H_f \tag{4}$$

where D_f, W_f, and H_f denote the driver, weather, and road hazard confidence values. Weights $\alpha = 0.5$, $\beta = 0.3$, and $\gamma = 0.2$ were empirically chosen based on validation performance.

The overall risk level was classified as:

$$R_L = \begin{cases} \text{Low}, & R_s < 0.3 \\ \text{Medium}, & 0.3 \leq R_s < 0.6 \\ \text{High}, & R_s \geq 0.6 \end{cases} \tag{5}$$

This mathematical fusion ensures that the driver's state contributes the most to the final risk score, followed by weather and road conditions. The large language model (LLM) interprets the fused outputs to produce descriptive alerts for each scenario.

During testing, the alerts were matched with the video timeline so that every message appeared at the same time as the driving scene. This helped in checking whether the warning was meaningful for what was shown in the frame. If a high-risk message appeared while the driver seemed active or the road was clear, the system timing was corrected. Through this process, the alerts became more natural and matched real driving conditions closely.

While building the alert sentences, only simple and clear words were used. We avoided long or repeated expressions so that the output stayed short and easy to follow. Each level of risk had a small set of phrases, which made the voice warnings sound steady and familiar. This helped the system behave more like a real assistant rather than a computer output.

All modules and the alert generator worked on the same local setup without internet access. This was done to make sure that private driving data did not leave the system. It also reduced delay between risk calculation and voice playback, which made the setup suitable for real-time use in vehicles or labs.

The system was also tested with night-time clips, rainy conditions, and poor lighting. Even when one of the modules gave uncertain results, the fusion model still produced a correct overall warning. This showed that the framework was stable and could continue working even when some inputs were noisy or incomplete.

While running the system many times, we noticed that the alerts and dashboard worked exactly the same way as in normal driving scenes. When the driver looked tired or the road became rough, the sound alert came almost instantly. Sometimes the message was short, sometimes a bit longer, but it always matched what was happening in the video. Seeing all three videos together with the colored border made it easier to understand how the system was judging the situation.

After several tests we changed the fusion weights little by little and compared the results. If the driver module weight was reduced, the alerts felt late or less serious. When we increased it again, the warning became more accurate and matched real driving logic. This showed that driver behaviour was the main reason for most high-risk outputs, while weather and road added supporting information. It made the complete model behave closer to how a human would think while driving.

The dashboard window was created to make the working of the complete system easy to observe in one place. It showed the three video clips side by side — driver, weather, and road — and displayed the alert sentence below them while the voice message played in the background. Each alert was connected to the current frame, so the video and the sound matched exactly. The color of the border changed based on the risk level, making it simple to identify high or low risk at a glance during playback.

TABLE IV: Examples of Fused Risk Scores and LLM-Generated Alerts (From Final CSV Output)

Driver State	Weather	Hazard	Risk Level	Generated LLM Alert
Yawning	Fog	Pothole	High	Driver yawning in fog; potholes detected ahead. Reduce speed and focus immediately.
Open	Normalday	Normal Road	Low	Driver alert and road clear; maintain safe driving conditions.
Sleeping	Rain	Crack	High	Driver sleeping during rain with cracked road ahead; stop vehicle immediately.
Closed	Sand	Normal Road	Medium	Driver's eyes closed with sandstorm and rough surface; reduce speed carefully.
Yawning	Snow	Open Manhole	High	Yawning driver in snow with open manhole detected; proceed with caution.

F. Dashboard Integration

The accident-risk dashboard combined driver, weather, and road modules into a synchronized visual interface. It displayed corresponding video feeds and played LLM-generated voice alerts without any bounding overlays, minimizing driver distraction.

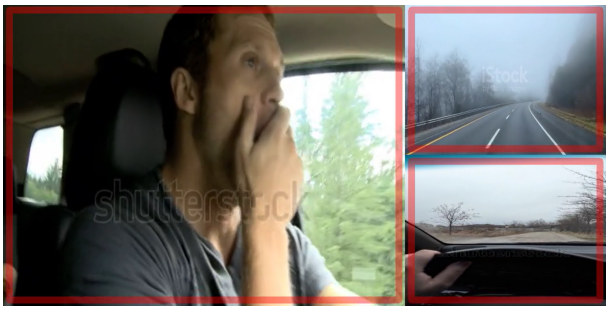

Fig. 10: Dashboard View 1 – Yawning driver with fog and potholes; alert issued for high-risk conditions.

Fig. 11: Dashboard View 2 – Eyes closed under rainy weather and cracked road; system triggers medium-risk warning.

Fig. 12: Dashboard View 3 – Drowsy driver under rainy weather and smooth road; synchronized medium-risk alert displayed.

Fig. 13: Dashboard View 4 – Eyes closed under sandy conditions and good road; system triggers medium-risk warning

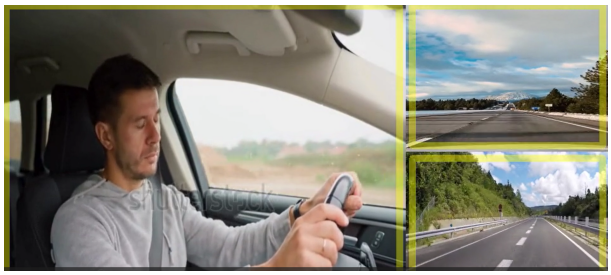

Fig. 14: Dashboard View 5 – Driver Drowsy alert with clear weather and smooth road; system reports low-risk driving state.

Fig. 15: Dashboard View 6 – Active driver with rain and road hazard; system identifies high-risk condition and issues alert.

G. Overall Summary

The proposed multimodal fusion + LLM pipeline achieved explainable and responsive accident-risk alerts. The MobileNetV3-Large (driver), EfficientNet-B0 (weather), and YOLOv8n (road hazard) modules collectively achieved over 94 % accuracy, demonstrating strong reliability for real-time driver-assistance systems.

V. Conclusion and Future Work

The study built a multimodal accident-risk prediction system that brings together three deep-learning models and a reasoning layer into one working setup. The driver module based on VGG16 recognised drowsiness from facial features, the EfficientNet-B0 network identified different weather scenes, and the YOLOv8n detector located cracks, potholes, and open manholes on the road. Their outputs were combined through a simple weighted equation to give one overall risk score. This result was then passed to a local language model, Ollama Mistral 7B, which generated short text alerts describing the situation. All alerts were placed on top of pre-recorded driver, weather, and road videos using the MoviePy library, producing an offline dashboard that showed both the visual scene and the predicted risk in a clear way. The system reached an overall accuracy of about 94 % on mixed test clips and produced consistent, understandable feedback.

In the next stage, the work can be moved closer to real-time use by creating a lighter version that runs on in-vehicle or edge hardware. Extra information such as head pose, steering pattern, or heart-rate data may also be added to make the driver analysis stronger. A text-to-speech (TTS) unit will be introduced so that the same alerts can be heard instead of only read on the dashboard. Future tests on real driving footage under different traffic and weather conditions will help measure how stable and fast the system is in practice.

References

[1] C.-H. Weng, Y.-H. Lai, and S.-H. Lai, "Driver Drowsiness Detection via Hierarchical Temporal Deep Belief Network," *ACCV Workshop on Driver Drowsiness Detection from Video*, 2016.

[2] F. Makhmudov et al., "Real-Time Fatigue Detection Using Machine Learning," *Electronics*, vol. 13, 2024.

[3] O. F. Hassan et al., "Real-Time Driver Drowsiness Detection Using Transformer Architectures," *Scientific Reports*, 2023.

[4] D. Arya et al., "RDD2022: Multinational Image Dataset for Road Damage Detection," *Data in Brief*, 2023.

[5] K. P. Singh and C. Dutta, "Machine Learning-Based Road Hazard Detection," *IJITSR*, 2023.

[6] Z. Jing, S. Li, and Q. Zhang, "YOLOv8-STE: Object Detection Under Adverse Weather," *Electronics*, 2024.

[7] H. Zhang et al., "WVIT-YOLO for Foggy Object Detection," *Applied Sciences*, 2024.

[8] S. Dewasi, A. Tamrakar, A. Sharma, and N. S. Naik, "Vehicular Obstacle Recognition and Tracking in Unfavourable Weather Conditions with Real-Time Video Pre-Processing and Deep Learning Framework," *IEEE Access*, vol. 12, pp. 112341–112357, 2024.

[9] E. M. Thompson et al., "SHREC 2022: Pothole and Crack Detection," *Computers Graphics*, 2022.

[10] D. Arya et al., "CRDDC 2022: Crowdsensing-Based Road Damage Challenge," *Data in Brief*, 2023.

[11] T. E. Choe et al., "HazardNet: Road Debris Detection by Synthetic Augmentation," *arXiv:2401.08976*, 2024.

[12] G. Parasnis et al., "RoadScan: Transfer Learning for Pothole Detection," *IEEE Access*, 2024.

[13] O. Khare et al., "YOLOv8-Based Visual Detection of Road Hazards," *IEEE Sensors Journal*, 2024.

[14] J. Cha et al., "Deep Learning-Based Road Damage Detection Using Satellite Imagery," *Remote Sensing*, 2024.

[15] M. Khan et al., "Pothole Detection for Autonomous Vehicles," *Frontiers in Computer Science*, 2024.

[16] Z. Wang et al., "Road Marking Damage Detection via Contextual Features," *Sensors*, 2024.

[17] M. Samo et al., "Deep Learning with Attention Mechanisms for Road Weather Detection," *Machine Learning with Applications*, 2024.

[18] M. Rathee et al., "Automated Road Defect and Anomaly Detection," *IEEE Access*, 2024.

[19] H. Du et al., "Recognition of Slippery Road Surface Based on Deep Learning," *PLOS ONE*, 2024.

[20] M. Doborjeh et al., "Multimodal Fusion for Intelligent Road Safety," *IEEE Access*, 2024.