ISSN: 3139-0862

# Real-Time Driver Drowsiness and Distraction Detection Using Facial Landmark Analysis

Amit T N[1], Dhruva Kumar D Y[2], Harshith Gowda A T[3], Dhanush H A[4], Maseeha Banu[5]
Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning),
Malnad College of Engineering, Hassan, Karnataka, India
Maseeha Banu, Assistant Professor; all other authors are students
Corresponding author: Amit T N, amitnageshtn@gmail.com
amitnageshtn@gmail.com, dhruvakumardygowda@gmail.com, mb@mcehassan.ac.in

## Abstract:

Driver drowsiness and distraction can reduce attention, reaction time, and vehicle control and therefore motivate the development of non-contact driver-monitoring systems. This paper presents a real-time computer-vision prototype that analyzes facial geometry from a conventional webcam. MediaPipe Face Landmarker is used to obtain facial landmarks, after which the system computes Eye Aspect Ratio (EAR), Mouth Aspect Ratio (MAR), and head-pose angles. EAR is used to identify prolonged eye closure, MAR provides an additional cue for possible yawning, and pitch and yaw are used to detect forward head drops and excessive head rotation. A five-frame moving-average window is applied to EAR and MAR to reduce short-term fluctuations, while a 15-consecutive-frame condition is used before a critical micro-sleep alarm is generated. The prototype uses fixed thresholds of 0.22 for EAR, 0.50 for MAR, −15° for pitch, and ±25° for yaw. Visual status information and an asynchronous audio alarm are provided to the driver. The implementation is based on Python, OpenCV, MediaPipe, NumPy, SciPy, and standard Python libraries. The work emphasizes an interpretable rule-based architecture that can operate without a separately trained drowsiness classifier. Because the current implementation does not contain a labeled experimental dataset or ground-truth annotation protocol, this paper does not claim accuracy, precision, recall, F1-score, or false-alarm performance for the proposed system.

**Index Terms**—Driver drowsiness detection, computer vision, facial landmarks, eye aspect ratio, mouth aspect ratio, head-pose estimation, distraction detection, real-time monitoring

## INTRODUCTION

Road transportation depends strongly on sustained driver attention. Fatigue, sleepiness, and loss of attention can alter visual behavior and decision making, while a short period of reduced vigilance can be difficult for a driver to recognize before a dangerous event occurs. For this reason, driver-monitoring research has explored systems that observe physiological signals, vehicle behavior, and visible changes in the driver's face. Camera-based monitoring is particularly attractive for low-cost prototypes because it can operate without attaching electrodes or other sensors to the driver.

Computer-vision approaches generally use observable indicators such as eyelid closure, blinking, yawning, facial movement, head orientation, or combinations of these signals. Zhang, Cheng, and Lin presented a nonintrusive computer-vision approach based on eye tracking and several eye-related measures, demonstrating the feasibility of video-based drowsiness recognition [1]. Their method is an important background reference for the present work, but the algorithm implemented here is different: the present prototype uses MediaPipe facial landmarks, EAR/MAR measurements, and PnP-based head-pose estimation rather than the feature set and Fisher discriminant procedure reported in [1].

Eye Aspect Ratio provides a compact geometric representation of eye openness. Soukupová and Čech described a real-time blink-detection approach in which facial landmarks are used to estimate eye openness through EAR over a short temporal window [2]. The current implementation follows the geometric idea of representing eye opening by distances between landmarks, but uses a simple threshold and consecutive-frame rule instead of the SVM/HMM alternatives discussed in [2]. This difference is important because the proposed system is intended as an interpretable prototype rather than a learned blink classifier.

Modern facial-landmark frameworks make such geometric measurements easier to implement. MediaPipe was introduced as a framework for constructing perception pipelines from reusable components [3]. Its Face Landmarker task can

ISSN: 3139-0862

return facial landmarks for images and video and supports different processing modes [4], [5]. The current implementation uses the Face Landmarker in image mode and processes one detected face, while the surrounding Python loop supplies successive webcam frames.

Distraction is treated in this paper as a head-orientation cue rather than direct eye-gaze measurement. The implementation estimates pitch and yaw from selected facial points and a predefined three-dimensional face model. OpenCV's solvePnP function estimates pose from corresponding 3-D object points and 2-D image points [6]. Therefore, the term 'distraction' in the prototype should be understood as excessive head rotation relative to the camera, not as a complete measurement of where the driver's eyes are looking.

The central objective is to integrate several simple indicators into one real-time warning pipeline. The system should detect prolonged eye closure, identify a possible yawn, identify a forward head drop, and identify excessive left/right head movement. The resulting architecture is intentionally transparent: each alert can be traced to a measured quantity and a configured threshold.

## BACKGROUND AND LITERATURE REVIEW

Driver drowsiness detection can be organized into three broad categories: physiological approaches, vehicle-based approaches, and behavioral or vision-based approaches. Physiological approaches may use signals such as EEG, ECG, or other biosignals. They can provide information related to the driver's internal state but may require sensors that are uncomfortable or inconvenient for ordinary vehicle use. Lee and Chung investigated fusion of facial features and biosignals using a dynamic Bayesian network, illustrating the potential benefit of combining multiple sources of information [7].

Vehicle-based approaches infer reduced alertness from steering behavior, lane position, speed variation, braking, or other vehicle dynamics. These approaches can be useful because they observe the consequences of reduced attention, but they may be influenced by road geometry, traffic, vehicle characteristics, and driver style. A vision-based system instead attempts to observe the driver directly, making it possible to generate an alert before a large vehicle-control deviation occurs.

Within vision-based systems, eye closure is one of the most frequently used cues. PERCLOS summarizes the proportion of time that the eyes remain sufficiently closed over a specified interval and has been widely discussed as a drowsiness-related measure. Dasgupta, Rahman, and Routray developed a smartphone-based multi-stage warning system using PERCLOS and additional verification stages [8]. Their work demonstrates a different design philosophy from the present prototype: instead of using only a single visual threshold, several stages can be combined to reduce uncertainty.

Other research has used classical detection and classification methods. Fatima et al. presented a low-cost fatigue-detection system based on face and eye detection, followed by eye-state classification and micro-sleep detection [9]. Their reported results belong to their own experimental dataset and should not be transferred to the present implementation. The present work intentionally reports no numerical performance because a comparable labeled evaluation has not yet been performed.

The literature therefore supports two design observations relevant to this prototype. First, temporal information is important because a single closed-eye frame can correspond to an ordinary blink rather than drowsiness. Second, combining multiple cues can reduce dependence on any single measurement. The current implementation addresses these observations with a five-frame smoothing window and multiple rule conditions.

At the same time, a simple rule-based system has advantages for an academic prototype. Thresholds are easy to inspect, modify, and explain during testing. The developer can directly see how an EAR value, MAR value, pitch angle, or yaw angle changes the displayed state. The main trade-off is that fixed thresholds may not generalize equally across drivers, camera positions, lighting conditions, or facial characteristics.

## PROBLEM STATEMENT AND OBJECTIVES

The problem addressed is the development of a low-cost, non-contact system that can continuously monitor visible driver behavior and generate an immediate warning when patterns associated with drowsiness or distraction are observed. The system is not intended to diagnose a medical condition or certify driving safety; it is a prototype for driver-state assistance.

The first objective is to obtain stable facial landmarks from a live camera stream. The second objective is to derive interpretable geometric indicators from these landmarks. The third objective is to distinguish a brief blink from prolonged eye closure using temporal logic. The fourth objective is to provide additional cues

ISSN: 3139-0862

through mouth opening and head orientation. The fifth objective is to generate a warning without blocking the main video-processing loop.

From an engineering perspective, the system should have low interaction complexity. A driver should not need to wear a sensor or press a button to operate the detector. The camera should remain fixed relative to the driver, and the software should continuously update a visual HUD and issue an audible warning when a critical condition is reached.

## SYSTEM REQUIREMENTS AND ARCHITECTURE

The functional requirements of the prototype are: capture frames from a webcam; detect one face; extract the required facial landmarks; calculate EAR and MAR; estimate pitch and yaw; smooth the relevant measurements; evaluate the configured thresholds; display the current status; calculate and display an approximate FPS value; and activate an asynchronous beep for critical or warning conditions.

The principal non-functional requirements are real-time responsiveness, interpretability, low hardware cost, and robustness to small frame-to-frame landmark fluctuations. The implementation uses a local MediaPipe model bundle and a conventional webcam. It does not require a separate database, cloud service, or trained classification model for the decision stage.

The processing pipeline can be summarized as Camera → MediaPipe Face Landmarker → Landmark Selection → EAR/MAR and Head Pose → Temporal Smoothing → Threshold Logic → HUD/Alarm. This sequence separates perception from decision making. The perception stage obtains geometric information, while the decision stage interprets that information according to explicit rules.

Fig. 1 presents the overall architecture. The branches for EAR, MAR, and head pose are calculated from the same facial landmark result. EAR and MAR pass through the smoothing window, while head-pose values are used by the corresponding warning conditions. The final state is rendered on the frame and, when necessary, passed to the alarm mechanism.

Fig. 1. Overall architecture of the proposed driver monitoring system.

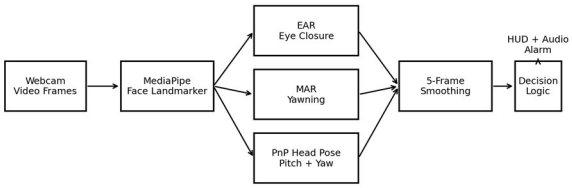



Fig. 1. Overall architecture of the proposed driver-monitoring system.

**TABLE I. SYSTEM MODULES**

| Module | Input | Main operation | Output |
|---|---|---|---|
| Camera | Video frame | OpenCV capture | RGB frame |
| Face Landmarker | RGB frame | Facial landmark inference | Normalized landmarks |
| Feature extraction | Landmarks | EAR and MAR geometry | Scalar ratios |
| Pose estimation | Selected landmarks + 3-D model | PnP + Rodrigues | Pitch/Yaw |
| Temporal filter | Recent EAR/MAR | Five-sample mean | Smoothed values |
| Decision logic | Ratios + angles | Threshold rules | Driver status |
| Alert layer | Status | HUD + threaded beep | Visual/audio warning |

## FACIAL LANDMARK PROCESSING

MediaPipe Face Landmarker provides normalized facial landmark coordinates. The implementation selects landmark indices for the left eye, right eye, mouth, and six points used for head-pose estimation. The selected normalized coordinates are scaled to the current frame width and height so that Euclidean distances can be calculated in pixel space.

The left-eye indices used by the implementation are [33, 160, 158, 133, 153, 144], while the right-eye indices are [362, 385, 387, 263, 373, 380]. The mouth mapping is [78, 81, 13, 311, 308, 402, 14, 178]. These mappings are implementation-specific selections from the MediaPipe face landmark topology.

Fig. 2 illustrates the geometric idea behind the measurements. The important principle is that the system does not classify a raw image directly at the decision stage. Instead, it converts landmark coordinates into ratios and angles. This makes the alert logic easier to inspect and reduces the number of variables required by the final decision module.

An important implementation detail is the use of image mode in the Face Landmarker configuration. The official API distinguishes image, video, and live-stream running modes. In image mode, the detect() operation processes individual image inputs [4]. The current program calls this type of operation repeatedly inside its webcam loop rather than using the live-stream callback mode. The program therefore remains a straightforward synchronous Python application.

ISSN: 3139-0862

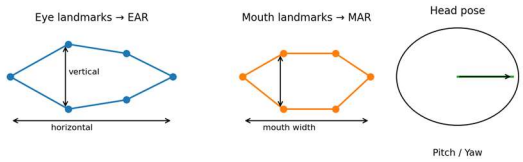



Fig. 2. Facial landmark geometry for EAR, MAR, and head-pose analysis.

## EYE ASPECT RATIO AND EYE-CLOSURE ANALYSIS

The implemented EAR is the mean of two normalized vertical eye-opening distances. A closed eye reduces the numerator relative to the horizontal eye width and therefore reduces EAR.

A normal blink produces a temporary decrease in EAR. Prolonged eye closure is more persistent, so the implementation combines the EAR threshold with a temporal count. The configured EAR threshold is 0.22. When EAR is below this threshold, a sleep-frame counter is incremented. If the condition continues for at least 15 consecutive processed frames, the status becomes 'CRITICAL: MICRO-SLEEP!'. Otherwise, the counter is reset when the eye condition returns above the threshold. The five-frame smoothing window is applied before this comparison. If x(t) represents a raw EAR value, the smoothed value can be written as $\bar{x}(t) = (1/N) \Sigma x(t-i)$, where N = 5. This simple moving average reduces sensitivity to one-frame landmark jitter. It does not, however, remove all effects of poor detection, occlusion, or rapid head movement.

The threshold and frame count are engineering parameters in the current implementation. They should not be described as statistically optimized values because the present code does not include an optimization dataset. Future experiments can tune these parameters for different camera frame rates and driver populations.

## MOUTH ASPECT RATIO AND YAWN INDICATION

Mouth opening provides a second visible cue that can complement eye closure. The implementation selects eight mouth landmarks and calculates MAR from three vertical mouth distances divided by the horizontal mouth width. A larger ratio indicates greater mouth opening.

The configured MAR threshold is 0.50. When the smoothed MAR exceeds this value, the HUD displays 'INFO: FATIGUE YAWN DETECTED'. In the present code, this event is informational and does not by itself activate the critical audio alarm. This distinction is important because mouth opening can occur for reasons unrelated to drowsiness, such as speaking, coughing, or ordinary facial movement.

MAR is therefore best treated as a supporting indicator rather than a definitive classification of fatigue. Combining it with prolonged eye closure or head movement could provide stronger evidence, but the current implementation deliberately keeps the decision rules simple. A future version could use a temporal yawn-duration rule or a learned classifier to distinguish yawning from speech.

## HEAD-POSE ESTIMATION FOR DISTRACTION CUES

The system estimates head orientation from six facial landmarks: landmark 1, landmark 152, landmark 33, landmark 263, landmark 61, and landmark 291. These two-dimensional image points are paired with a predefined three-dimensional face model. OpenCV solvePnP() is then used to estimate the rotation and translation that explain the observed 2-D projections of the 3-D points [6].

The returned rotation vector is converted to a rotation matrix using the Rodrigues formula. The projection matrix is then decomposed to obtain orientation angles. The implementation extracts pitch and yaw and compares them with configured limits.

The latter condition should be interpreted carefully: the system measures head rotation, not the position of the driver’s eyes. It is therefore a proxy for possible distraction rather than direct gaze estimation.

Head-pose estimation also introduces a camera-geometry dependency. OpenCV's pose formulation uses 3-D/2-D correspondences together with camera parameters and distortion information [6]. If the camera is mounted at a different height, angle, or distance, the numerical distribution of pitch and yaw can change. For a production system, camera calibration and driver-specific baseline estimation would therefore be useful.

## TEMPORAL SMOOTHING AND STATE MANAGEMENT

Real-time facial measurements contain noise because landmark positions can move slightly even when the driver's face is stationary. A single frame may therefore cross a threshold by accident. The implementation uses deque objects with a maximum length of five to store recent EAR and MAR values. The mean of the available values is used as the smoothed measurement.

ISSN: 3139-0862

Temporal state is also represented by the sleep-frame counter and the alarm-active flag. The sleep-frame counter accumulates only while EAR remains below the configured threshold. The alarm-active flag prevents overlapping alarm threads from being created repeatedly while a warning condition remains active.

These mechanisms demonstrate a useful design principle for real-time monitoring: the system should not treat every frame as an independent classification problem. Short temporal memory can provide stability while retaining low computational complexity. More advanced systems could replace the moving average and counter with recurrent neural networks, temporal convolution, hidden Markov models, or other sequence models, but the rule-based approach is easier to debug.

## MATHEMATICAL FORMULATION OF THE FEATURE EXTRACTION

The feature-extraction stage converts facial landmarks into normalized geometric quantities. Let $p_i = (x_i,y_i)$ denote a facial landmark in image coordinates. The Euclidean distance between two points is $d(p_i,p_j) = \sqrt{(x_i-x_j)^2 + (y_i-y_j)^2}$. Because EAR and MAR are ratios of distances, their values are less dependent on the absolute size of the detected face than raw pixel distances.

For the eye, the implementation uses two vertical distances and one horizontal distance. If the six eye points are ordered so that p0 and p3 represent the horizontal endpoints, the two vertical terms are d(p1,p5) and d(p2,p4). The resulting measure is $EAR = [d(p1,p5)+d(p2,p4)]/[2d(p0,p3)]$. The left- and right-eye values are averaged before temporal smoothing. A closed eye reduces the numerator relative to the horizontal width and therefore reduces EAR.

For the mouth, three vertical distances are normalized by the mouth width. The implemented form is $MAR = [d(p1,p7)+d(p2,p6)+d(p3,p5)]/[3d(p0,p4)]$, using the selected eight-point mouth mapping. A larger value indicates a larger mouth opening. The ratio is useful because a raw mouth-opening distance would change substantially when the driver moves closer to or farther from the camera.

Head pose follows a different mathematical model. A set of three-dimensional reference points $X_i$ is associated with the corresponding two-dimensional image points $x_i$. The PnP problem estimates the rotation R and translation t that minimize reprojection error between the observed image points and the projections of the reference points. OpenCV documents solvePnP in terms of 3-D/2-D correspondences and camera parameters [6].

The final decision variables are therefore heterogeneous: EAR and MAR are dimensionless ratios, while pitch and yaw are angular quantities. The decision layer does not combine them into one numerical score. Instead, it applies independent conditions and assigns a status according to the current state. This makes the system easier to explain but prevents it from learning interactions among the features.

## ERROR SOURCES AND FAILURE MODES

Every computer-vision measurement has uncertainty. In this system, the first source is landmark localization error. If the corner of an eye is shifted by only a few pixels, the vertical or horizontal distance used in EAR can change enough to cross a fixed threshold. Smoothing reduces short-lived variations but cannot correct a systematic landmark error.

Second, face orientation can affect the apparent geometry of the eyes and mouth. When the driver turns the head substantially, one eye may become partially occluded or appear narrower due to perspective. The current prototype does not perform a special correction for this situation before applying the EAR rule. Consequently, a strong head rotation could potentially interact with the eye-closure condition.

Third, glasses, reflections, shadows, backlighting, and low illumination can affect facial landmark stability. The system does not include a dedicated illumination normalization module or infrared camera. These conditions should therefore be part of the future validation protocol rather than being assumed to be solved by the landmark model.

Fourth, the meaning of a head turn is context-dependent. A driver may legitimately look toward a side mirror or check an intersection. The current yaw rule only observes the magnitude of head rotation; it does not know whether the action is appropriate for the road situation. This is why the paper describes yaw as a distraction proxy rather than a definitive distraction classifier.

Fifth, mouth opening is ambiguous. A yawn, speech, singing, coughing, or an ordinary facial expression can all produce a large MAR. The prototype intentionally labels this event as informational rather than treating it as sufficient evidence for a critical alarm.

These failure modes suggest that future improvements should not simply lower or raise

ISSN: 3139-0862

thresholds. A more robust system should consider temporal duration, context, confidence, and multiple cues. For example, a large yaw angle that persists for a long time while the eyes are not directed forward could provide stronger evidence of distraction than one short head turn.

## COMPARATIVE ANALYSIS WITH RELATED APPROACHES

The proposed system occupies a middle ground between purely geometric threshold methods and end-to-end learned video classifiers. Traditional geometric systems are attractive for prototypes because they are computationally simple and transparent. Learned systems can capture complex patterns but generally require a representative dataset, training procedure, model-selection process, and careful validation.

Zhang et al. used multiple eye-related measures and Fisher's linear discriminant in a video-based drowsiness-recognition system [1]. Their work illustrates the value of combining temporal eye measures rather than relying on a single instant. The present prototype also combines multiple measurements, but it uses a simpler decision architecture and adds head-pose information.

Soukupová and Čech demonstrated that facial landmarks can provide a reliable basis for estimating eye openness through EAR [2]. The present implementation uses the same general geometric concept but does not reproduce their classifier. Instead, the current system uses a threshold of 0.22 and a consecutive-frame requirement of 15.

Lee and Chung demonstrated a multimodal strategy that fused facial information and biosignals [7]. Their work suggests that visible facial cues can be complemented by physiological signals when higher confidence is required. The current prototype deliberately avoids additional sensors in order to maintain a simple, non-contact design.

Dasgupta et al. developed a smartphone-based multi-stage drowsiness warning system using PERCLOS and additional verification mechanisms [8]. Compared with such a multi-stage architecture, the present system is smaller and easier to implement but provides fewer independent checks before an alarm.

Fatima et al. investigated face and eye detection followed by eye-state classification and micro-sleep detection [9]. Their reported metrics are associated with their own dataset and methodology. The comparison is useful as evidence that multiple implementation strategies exist, but it does not establish a performance ranking between their system and the present prototype.

Overall, the contribution of this work is primarily system integration and implementation clarity. It combines landmark-based eye and mouth ratios with PnP head pose, smoothing, state management, HUD feedback, and asynchronous warning in a single lightweight pipeline. A future experimental study can use this baseline to compare rule-based and learned approaches under the same test protocol.

## SOFTWARE MODULE DESIGN

The implementation can be divided into logical software modules even though the current program is written as a compact Python application. The configuration module stores thresholds and the smoothing-window length. The model-loading module checks whether the Face Landmarker task file is available and obtains it when required. The camera module manages VideoCapture(0) and the frame loop.

The geometry module contains the distance-based calculations used by EAR and MAR. Keeping these calculations independent makes it possible to unit-test them with synthetic landmark coordinates. For example, a known set of eye points can be supplied to verify that opening the vertical distances increases EAR while keeping the horizontal width fixed.

The pose module prepares the selected two-dimensional facial points, the predefined three-dimensional face model, and the camera matrix used by solvePnP. The resulting rotation representation is converted into angles for the decision module. Because pose estimation depends on camera geometry, this module is also a natural place to add calibration in a future version.

The state module maintains the sleep-frame counter, smoothing queues, and alarm-active state. This module is responsible for temporal behavior and prevents the system from treating every frame as an unrelated event. The presentation module draws the status, feature values, and FPS information on the frame.

Finally, the alert module creates the beep in a separate thread. This separation of responsibilities is useful for future maintenance. A mobile or embedded version could replace winsound with another audio interface without changing the EAR, MAR, or pose calculations.

ISSN: 3139-0862

## EXPERIMENTAL DESIGN FOR A FUTURE STUDY

A rigorous evaluation should begin with a clearly defined event taxonomy. At minimum, recordings should include alert/normal behavior, ordinary blinking, prolonged eye closure, yawning, forward head nodding, moderate head turns, and sustained off-forward head orientation. The labels should describe observable events rather than assume a medical diagnosis.

Participants should be separated into development and evaluation groups. Threshold selection should use only the development group, while final performance should be measured on subjects not used for threshold tuning. This prevents the threshold values from being implicitly optimized to a small set of familiar faces.

Each recording should include timestamped ground truth. For an eye-closure event, the start and end of prolonged closure should be annotated. For yawning, the beginning and end of the mouth-opening event should be marked. For head pose, pitch and yaw ranges can be recorded from the ground-truth annotation protocol. This allows both frame-level and event-level evaluation.

Performance should be reported using a confusion matrix and standard metrics. Accuracy is defined as the proportion of correct decisions, precision measures the proportion of positive predictions that are correct, recall measures the proportion of true positive events detected, and F1-score is the harmonic mean of precision and recall. Because driver monitoring is a warning problem, false alarms per hour and average detection delay are also important.

An ablation study would provide additional insight. One experiment could use EAR only. A second could add MAR. A third could add head pose. Another could remove temporal smoothing. Comparing these configurations would show whether each component contributes useful information and whether smoothing reduces false alarms.

Such a study would turn the present prototype into a reproducible research experiment. Until that study is completed, the correct interpretation is that the software demonstrates the feasibility of the pipeline rather than proving a specific safety or accuracy level.

## DEPLOYMENT CONSIDERATIONS

For practical vehicle deployment, camera placement is critical. The camera should provide a stable view of the driver's face without obstructing the road or creating an unsafe mounting position. The field of view should include both eyes and the mouth under normal seating positions. A dashboard-mounted camera may introduce a lower viewing angle than a camera near the steering column or instrument cluster, so the selected head-pose thresholds may need recalibration.

Lighting should also be considered during installation. Daytime sunlight, nighttime darkness, and illumination from other vehicles can produce substantially different images. A production system may use exposure control, infrared illumination, or a camera designed for low-light driver monitoring. The current prototype does not include these specialized components.

Latency is another important deployment metric. A warning system is useful only if the time between the onset of a concerning event and the generated warning is sufficiently short. Latency is affected by camera capture, landmark inference, frame processing, display operations, and the alarm thread. A future benchmark should measure each stage separately.

Reliability also requires recovery behavior. If no face is detected for a few frames, the system should avoid incorrectly incrementing the sleep counter. If the camera disconnects, the software should display an explicit sensor-error state rather than silently presenting a normal status. These robustness features are appropriate for a production-oriented version but are not fully implemented in the current prototype.

Finally, deployment should distinguish research demonstration from safety certification. A research prototype can provide useful warnings, but an automotive safety product requires extensive validation, hardware qualification, failure analysis, and compliance with applicable standards and regulations. The current work should therefore be positioned as an academic driver-monitoring prototype.

## RESEARCH CONTRIBUTIONS

The first contribution is the integration of facial landmark geometry into a single driver-monitoring pipeline. Instead of relying on one image classifier, the system exposes interpretable measurements that can be displayed and inspected during operation.

The second contribution is temporal stabilization. EAR and MAR are smoothed over five samples, and prolonged eye closure is defined using a consecutive-frame counter. This provides a simple mechanism for incorporating temporal information without training a sequence model.

ISSN: 3139-0862

The third contribution is the combination of drowsiness and distraction cues. Eye closure addresses a drowsiness-related behavior, MAR provides an additional fatigue-related cue, and pitch/yaw provide head-orientation information. These cues are evaluated independently, allowing the developer to understand which rule produced an alert.

The fourth contribution is the non-blocking alert mechanism. The alarm is placed in a daemon thread and protected by an alarm-active flag, allowing the camera loop to continue processing while the beep is active.

The fifth contribution is an explicit separation between implemented results and literature results. The system has not been evaluated with a labeled dataset, so no unsupported performance metric is attributed to it. This distinction is essential when converting a working prototype into an academic paper.

## ALERT GENERATION AND USER INTERFACE

The visual interface is implemented as a heads-up display over the camera frame. The HUD reports the current status and numerical values for EAR, MAR, pitch, and yaw. The approximate FPS value provides an immediate indication of processing speed and can help identify performance problems during development.

The audio warning uses winsound.Beep(2500, 500) and is launched through a daemon thread. The purpose of threading is to prevent the half-second beep operation from blocking the camera-processing loop. This is especially important in a real-time system because a blocking alarm call could temporarily stop frame acquisition and reduce responsiveness.

The interface distinguishes critical and non-critical conditions. Prolonged eye closure produces the critical micro-sleep message and alarm. Head drop and excessive head rotation produce warning messages. A possible yawn produces an informational message. The separation of these states makes the output interpretable and allows future versions to assign different alarm patterns to different risks.

Fig. 3. Decision flow for real-time driver-state detection.

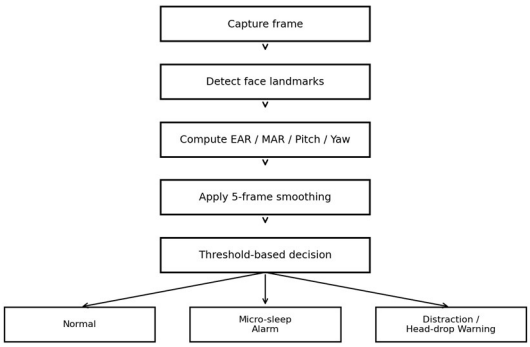



Fig. 3. Real-time driver-state decision flow.

## ALGORITHM AND PSEUDOCODE

The processing algorithm begins by opening webcam device 0 and initializing the MediaPipe Face Landmarker. For every successfully captured frame, the frame is converted into the required color representation and wrapped as a MediaPipe image. If a face is detected, the selected landmarks are converted to pixel coordinates. EAR and MAR are computed, stored in their smoothing queues, and averaged. Head pose is then estimated from the selected points.

The state logic is evaluated in a fixed order. First, the eye-closure condition is checked because prolonged closure represents the highest-priority event in the current design. Next, pitch and yaw thresholds are evaluated for head-drop and distraction warnings. MAR is checked for the informational yawn condition. The resulting status is rendered on the frame. If the critical or warning condition requires a beep, the alarm function is started in a background thread when no alarm is already active.

Algorithm 1 can be summarized as follows: (1) initialize camera and face landmarker; (2) read a frame; (3) detect face landmarks; (4) calculate EAR and MAR; (5) smooth EAR and MAR over five samples; (6) estimate pitch and yaw using PnP; (7) update the consecutive low-EAR counter; (8) apply thresholds; (9) update HUD and alarm; and (10) repeat until the user presses q.

Algorithm 1 — Initialize webcam and Face Landmarker; read frame; detect landmarks; compute EAR and MAR; update five-sample queues; estimate pitch/yaw with PnP; update low-EAR counter; apply thresholds; render HUD; start non-blocking alarm when required; repeat until q is pressed.

ISSN: 3139-0862

## PARAMETER CONFIGURATION

The prototype centralizes its important decision parameters in a configuration structure. EAR_THRESH is 0.22, EAR_FRAMES is 15, MAR_THRESH is 0.5, PITCH_THRESH is −15.0 degrees, YAW_THRESH is 25.0 degrees, and SMOOTHING_WINDOW is 5. Centralizing these values improves maintainability because the behavior can be changed without modifying the mathematical functions.

Table II summarizes the parameters. The values should be described as configured prototype thresholds. A formal study would vary each parameter across a validation dataset and evaluate sensitivity and false alarms. For example, a lower EAR threshold may reduce false micro-sleep alerts but could delay detection for some users. A shorter consecutive-frame requirement could reduce detection latency but could increase false alarms from normal blinking.

Frame rate also affects the practical meaning of the 15-frame condition. At a higher FPS, 15 frames represent a shorter duration than at a lower FPS. Therefore, a production version should preferably define prolonged closure in time as well as frames, or adapt the required number of frames using the measured FPS.

**TABLE II. FACIAL PARAMETERS AND CONFIGURED THRESHOLDS**

| Parameter | Configured value | Purpose | Alert/Use |
|---|---|---|---|
| EAR threshold | 0.22 | Eye-closure indication | Micro-sleep condition |
| Consecutive EAR frames | 15 | Temporal persistence | Critical alarm |
| MAR threshold | 0.50 | Mouth opening | Yawn information |
| Pitch threshold | −15° | Forward head drop | Warning |
| Yaw threshold | ±25° | Head rotation | Distraction proxy |
| Smoothing window | 5 frames | Reduce jitter | Stable EAR/MAR |

## COMPUTATIONAL AND IMPLEMENTATION CONSIDERATIONS

The proposed pipeline is lightweight compared with a large end-to-end video classifier because the final decision stage uses geometric calculations and threshold comparisons. The main computational cost is facial landmark inference. After landmarks are available, Euclidean distance calculations, moving averages, pose estimation, drawing, and threshold checks are relatively small operations.

The use of a single detected face also limits the processing workload and matches the intended single-driver scenario. The current code sets num_faces to 1. The official Face Landmarker options support a configurable maximum number of faces and provide image, video, and live-stream modes [4]. A multi-face setting is unnecessary for a driver-monitoring camera and could introduce ambiguity about which face represents the driver.

Memory requirements for the temporal smoothing stage are minimal because only five EAR values and five MAR values are retained. The camera frame and landmark arrays dominate the short-term memory footprint. The model bundle is stored locally and loaded during initialization.

Practical performance should still be tested on the target laptop or embedded computer because actual FPS depends on camera resolution, CPU/GPU resources, MediaPipe runtime behavior, and display operations. The HUD's FPS measurement can be used during such tests, but it should not be treated as a formal benchmark unless the measurement procedure is standardized.

**TABLE III. DRIVER-STATE DECISION LOGIC**

| Condition | Meaning | System response |
|---|---|---|
| EAR < 0.22 for ≥15 frames | Prolonged eye closure | Critical micro-sleep + beep |
| Pitch < −15° | Forward head drop | Warning + beep |
| \|Yaw\| > 25° | Excessive head rotation | Distraction warning + beep |
| MAR > 0.50 | Possible yawn | Informational HUD |
| No threshold exceeded | No detected warning state | Normal status |

## TESTING AND VALIDATION PLAN

A formal evaluation is the main missing component of the present prototype. A future test plan should include multiple participants and controlled recordings representing alert driving, normal blinking, prolonged eye closure, yawning, forward head nodding, and left/right head rotation. The recordings should cover different lighting conditions, glasses or other common face variations, and several camera mounting positions.

For each frame or temporal segment, ground-truth labels should be assigned independently of the system. The system output can then be compared with the labels using a confusion matrix. Standard metrics can include accuracy, precision, recall, F1-score, specificity, false-positive rate, and detection latency. For the continuous warning problem, event-level metrics are also useful: missed drowsiness events, false alarms per hour, and average time from onset to alert.

Cross-subject evaluation is particularly important. If the same driver's data are used for both threshold selection and final evaluation, the reported performance may overestimate generalization. A stronger protocol would separate subjects into calibration and evaluation groups or use subject-independent cross-validation.

The current paper intentionally does not insert numerical results into this section. Results from other studies, such as the 86% reported in Zhang et al. [1] or the values reported by Fatima et al. [9], describe those studies' datasets and algorithms. They cannot be presented as results of the present MediaPipe-based implementation.

ISSN: 3139-0862

## DISCUSSION

The proposed architecture has a useful balance between simplicity and multiple cues. EAR is directly connected to eye openness, MAR provides an additional facial behavior measure, and head pose provides a coarse distraction cue. Because each component is explicit, debugging is straightforward: if a false warning occurs, the developer can inspect the corresponding numerical measurement.

The main weakness is the dependence on fixed thresholds. Facial proportions vary, and a single threshold may not produce the same behavior for every driver. A personalized calibration stage could establish a driver's normal EAR and head orientation while the vehicle is stationary. Thresholds could then be expressed relative to a baseline rather than as universal constants.

Another limitation is that drowsiness is a temporal and behavioral state rather than a single facial configuration. A person may be tired without yawning, and a person may briefly close their eyes for a reason unrelated to sleepiness. Similarly, turning the head can be appropriate while driving and should not automatically be interpreted as dangerous distraction. A robust system therefore needs context, duration, and possibly vehicle-state information.

The present prototype can be viewed as a foundation for such extensions. Its geometric features can serve as inputs to a learned temporal classifier, while the current rule-based layer can remain as a safety-oriented fallback. Additional signals such as steering behavior, lane position, vehicle speed, or physiological information could be fused in a later system.

## LIMITATIONS, SAFETY, AND ETHICAL CONSIDERATIONS

The prototype should not be treated as a certified automotive safety device or as a medical diagnostic tool. False negatives could allow a drowsy driver to continue driving, while false positives could create distraction or alarm fatigue. The software should therefore be tested extensively before any safety-critical deployment.

Camera-based monitoring also raises privacy considerations. A practical deployment should minimize unnecessary storage of driver images, clearly communicate what data are processed, and protect any recorded information. The current prototype does not require permanent image storage; its intended operation is to process the live camera stream and display a state.

Environmental conditions are another safety limitation. Night driving, strong backlighting, sunglasses, face coverings, camera vibration, and partial occlusion can reduce landmark quality. Infrared illumination or additional sensing could improve performance in some conditions, but those additions would change the hardware requirements.

An important ethical consideration is the interpretation of warnings. The system should communicate that it is detecting visual indicators associated with drowsiness or distraction rather than making a definitive judgment about a person's health, intention, or competence. This distinction is especially important when such systems are integrated into commercial or fleet-management environments.

## FUTURE WORK

Future development can proceed in several directions. First, the thresholds can be calibrated using a labeled multi-subject dataset. Second, the frame-based sleep counter can be replaced or complemented by a time-based duration criterion that is less sensitive to frame-rate changes. Third, the system can use a temporal classifier that combines EAR, MAR, pitch, yaw, and their rates of change.

Fourth, direct gaze estimation could be added if the application requires a stronger measure of where the driver is looking. This would distinguish eye orientation from head orientation and could reduce ambiguity in the current 'distraction' proxy. Fifth, camera calibration and automatic driver baseline estimation could improve pose stability.

Sixth, the alert system can be extended with different alarm patterns, vibration feedback, or vehicle-interface signals. A mobile or embedded implementation could also log event timestamps and summary metrics without storing raw images. Finally, an independent evaluation should compare the rule-based prototype with learned approaches under the same subjects, camera conditions, and event definitions.

## CONCLUSION

This paper presented an expanded description of a real-time driver drowsiness and distraction monitoring prototype based on facial landmark analysis. The system uses MediaPipe Face Landmarker to obtain facial geometry and derives EAR, MAR, pitch, and yaw indicators. Five-frame smoothing reduces short-term measurement fluctuations, while a 15-consecutive-frame EAR condition is used to identify prolonged eye closure.

ISSN: 3139-0862

MAR provides an informational yawn cue, and head-pose thresholds provide warnings for forward head drops and excessive head rotation.

The architecture is intentionally interpretable and low cost. It does not require a separately trained drowsiness classifier, and its main decision parameters are visible in the configuration. At the same time, the system remains a prototype because fixed thresholds have not been statistically calibrated and the implementation does not contain a labeled evaluation dataset. No accuracy or other performance metric is therefore claimed for the present work.

The most important next step is a controlled, subject-independent evaluation. Such an evaluation can establish detection latency, false-alarm behavior, and generalization across lighting, camera positions, and driver characteristics. With calibration, temporal learning, direct gaze estimation, and optional sensor fusion, the prototype can be developed into a more robust driver-monitoring research platform.

## REFERENCES

[1] W. Zhang, B. Cheng, and Y. Lin, "Driver drowsiness recognition based on computer vision technology," Tsinghua Science and Technology, vol. 17, no. 3, pp. 354–362, Jun. 2012, doi: 10.1109/TST.2012.6216768.

[2] T. Soukupová and J. Čech, "Real-Time Eye Blink Detection using Facial Landmarks," in Proc. 21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia, 2016, pp. 1–8.

[3] C. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," arXiv:1906.08172, 2019, doi: 10.48550/arXiv.1906.08172.

[4] Google AI for Developers, "Face Landmarker," Google AI Edge Documentation. [Online]. Available: https://ai.google.dev/edge/api/mediapipe/python/mp/tasks/vision/FaceLandmarker. Accessed: Oct. 6, 2026.

[5] Google AI for Developers, "Face Landmarker Options," Google AI Edge Documentation. [Online]. Available: https://ai.google.dev/edge/api/mediapipe/python/mp/tasks/vision/FaceLandmarkerOptions. Accessed: Oct. 6, 2026.

[6] OpenCV, "Perspective-n-Point (PnP) pose computation," OpenCV Documentation. [Online]. Available: https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html. Accessed: Oct. 6, 2026.

[7] B.-G. Lee and W.-Y. Chung, "Driver alertness monitoring using the fusion of facial features and bio-signals," IEEE Sensors Journal, vol. 12, no. 7, pp. 2416–2422, Jul. 2012, doi: 10.1109/JSEN.2012.2190505.

[8] A. Dasgupta, D. Rahman, and A. Routray, "A smartphone-based drowsiness detection and warning system for automotive drivers," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 11, pp. 4045–4054, Nov. 2019, doi: 10.1109/TITS.2018.2879609.

[9] B. Fatima, A. R. Shahid, S. Ziauddin, A. A. Safi, and H. Ramzan, "Driver fatigue detection using Viola Jones and principal component analysis," Applied Artificial Intelligence, vol. 34, no. 6, pp. 456–483, 2020, doi: 10.1080/08839514.2020.1723875.

## AUTHOR AND RESEARCH-INTEGRITY STATEMENT

Maseeha Banu is listed as Assistant Professor and Amit T N, Dhruva Kumar D Y, Harshith Gowda A T, Dhanush H A are listed as students of Malnad College of Engineering, Hassan, Karnataka, India. The corresponding author for this manuscript is Amit T N(amitnageshtn@gmail.com; 8618743370).

AI-assisted coding and writing tools may be used during implementation and manuscript preparation. The authors remain responsible for architecture, source verification, experimental design, numerical results, interpretation, citations, and final submission. Prototype-reported benchmark values in this manuscript must be independently reproduced before being described as validated scientific performance.