

AI-Powered Attention Monitoring with Live Computer Vision Processing

Anurag Kawade¹, Prajwal Dalvi², Yash Dhumal³, Dr. Kishor Wagh⁴

^{1,2,3,4} *Department of Computer Engineering, AISSMS Institute of Information Technology, Pune, India*
anuragkawade17@gmail.com, prajwaldalvi17@gmail.com, yashdhumal8484@gmail.com, waghks@gmail.com

Abstract

Attention-recording technologies are becoming increasingly necessary due to their potential applications within multiple industries, including driver safety, telecommuting, educational environments with distance learning, and cognitive assessments. Unfortunately, the majority of current solutions utilize cloud API services or require specialized expensive hardware systems and software that require extensive resources and are not scalable. We propose a browser application that records and measures attention using artificial intelligence methods based on MediaPipe Face Landmarker and TensorFlow's JavaScript API. The application runs completely on the client side and obtains facial features, Eye Aspect Ratio (EAR), head pose, and gaze estimation from webcam or pre-recorded video. Testing indicates smooth operation at approximately 25–30 FPS without uploading video files to an external server. Potential applications include driving performance monitoring, online teaching, worker productivity evaluation, and cognitive assessment.

Keywords. attention monitoring, computer vision, MediaPipe Face Landmarker, TensorFlow.js, eye aspect ratio, gaze estimation, head pose, browser-based processing.

1. INTRODUCTION

With the fast development of e-learning systems, telecommuting setups, and communication technologies based on videos, many human activities have been transferred to screen-centered spaces. In such settings, being able to accurately assess and track user attention becomes critical in enhancing educational performance, guaranteeing safety, and analyzing human behaviors. But attention is an internal mental phenomenon, which can only be indirectly determined by measurable physiological indicators, such as eye blink rate, gaze direction, and head pose.

Typical techniques for estimating user attention commonly involve either human judgment, personal reports, or specialized hardware devices, like infrared eye trackers and electroencephalography (EEG) helmets. Such techniques are expensive, invasive, and not easy to deploy to large-scale distributed scenarios. Moreover, many proposed solutions in research are implemented using native desktop software or off-the-shelf video streams processed on remote servers.

Modern web technologies and machine learning libraries allow computer vision workflows to be performed locally in the web browser. MediaPipe Face Landmarker can detect dense facial landmarks in real-time mode. TensorFlow.js is used for running models with GPU acceleration. Combined with HTML5 video capture, Canvas, and Chart libraries, such technologies could offer real-time attention detection without any installations or third-party services.

2. RELATED WORK

2.1. Attention and Gaze Estimation

The analysis of eye-gaze and blinking has been extensively studied in the contexts of drowsiness detection in drivers, engagement monitoring in students, and human-computer interaction. The Eye Aspect Ratio (EAR) continues to be one of the most popular metrics for blink detection and drowsiness estimation and is derived from the distance between certain eye landmarks in both the horizontal and vertical directions. Head pose estimation and gaze direction help estimate attention by determining whether a user is gazing towards or away from the screen.

2.2. *Facial Landmarks Detection Approaches*

Traditional approaches rely on Dlib, Haar cascades, and hand-crafted features. Modern deep learning frameworks include MediaPipe Face Mesh that outputs dense facial landmarks at high frame rates on consumer devices. The resulting landmarks allow deriving geometric features, such as EAR and head pose estimations, without the need for 3D modeling.

2.3. *Browser-Based Computer Vision*

Thanks to TensorFlow.js and WebAssembly, inference is possible using the browser. MediaPipe offers JavaScript APIs for loading models via CDNs and accelerating them with the help of GPU. Some previous studies have implemented browser-based pose detection and gesture recognition, although very few projects combined these functions with the ability to create an end-to-end attention monitoring system with session recording, video analysis, and analytics output.

3. KEY CONTRIBUTION

The key contributions of this paper are as follows:

- First, we design and build a one-page application that integrates real-time monitoring, analytics and video-based attention assessment in a single integrated solution.
- Secondly, a computer vision engine is implemented inside the user's browser with MediaPipe Face Landmarker and TensorFlow.js frameworks, which allows us to detect facial landmarks in real time without any extra software installation and even server-side computations.
- Third, we propose an attention score calculation algorithm based on different behavioral cues such as blink rate, head pose, gaze direction, and eye aspect ratio (EAR).
- Fourth, we design a real-time processing pipeline that continuously analyzes the input from the user's webcam, calculating various behavioral metrics and drawing visual overlays in real time.
- Fifth, we propose a dynamic alerting system which can detect the absence of user's attention or eye closure, and send real-time alerts visually and audibly to the user.

4. METHOD

4.1. *Technology Stack*

All functionality within this part of the system is available through a modern web browser, developed using JavaScript. Key technologies are as follows:

- MediaPipe Tasks Vision: Provides a Face Landmarker Model capable of predicting 468 facial landmarks and their associated blendshape parameters.
- TensorFlow.js: Provides backend initialization capabilities as well as the ability for possible later integration of additional machine learning models.
- HTML5 Video & Canvas: Used to render from the camera, play back video, and create visual overlays.
- JavaScript ES6: Responsible for executing application logic, managing application states and firing events.
- Real-Time Monitoring Pipeline

The application connects the stream to an invisible video element, syncs a canvas element to have the same resolution as the video element, and makes a call to `getUserMedia` in order to access webcam data upon pressing the Start Camera button. Once the processing starts via `requestAnimationFrame` and a customizable frame rate, each iteration copies the current frame from the video to the canvas; provides the timestamp of the frame to `faceLandmarker.detectForVideo`; processes the first detected set of landmarks when a face is found; calculates EAR, head pose, and gaze angle; calculates and smooths the attention score; draws facial landmarks and eye outlines when selected.

4.2. *Blink Detection and EAR Computation*

EAR is computed separately for the left and right eyes using six landmarks per eye. To make the range easier to interpret for UI thresholds, a correction function normalizes the average EAR from both eyes. The blink detection is performed when the EAR goes below the

threshold for a period of time, while a queue of blink times is being recorded in order to calculate the rate of blinks per minute.

4.3. Head Pose and Gaze Estimation

Head position can be estimated based on facial feature points, like the tip of the nose, the corners of the eyes, and the bottom of the chin. Rather than solving for 3D head poses, the current system relies on 2D geometry to calculate yaw, pitch, and roll. Gaze direction can then be loosely classified into one of five categories, including center, left, right, up, or down. These statistics, which represent the movement of the user's head over time, are collected by the analytics engine and shown through the user interface.

4.4. Offline Video Analysis

The video analysis tool provides an option for uploading videos that are available in either MP4 or WebM formats. The application uses an invisible video element and the off-screen canvas in order to analyze images in the video at a consistent frame rate of 30 FPS.

- 1) the video is positioned at the relevant timestamp;
- 2) the frame is rendered to the off-screen canvas;
- 3) the face landmarker processes the frame;
- 4) EAR, gaze direction, head pose, and attention score are computed as in real-time mode;
- 5) the results are stored in an analysisData structure. After all frames are processed, the module computes a timeline of frame-wise attention values, distributions of high, medium, and low attention, face detection success rates, blink events and average blink frequency.

5. RESULTS

Table 1 Summary of empirical performance

Metric	Observed Value
Average FPS (live monitoring)	25–30
Average FPS (offline analysis)	10–20
Face detection rate (frontal)	> 95%
Blink detection consistency	> 90% (controlled blinks)
Correct gaze category in scripted turns	≈ 90%
Stable attention scoring in focused interval	High (near 80–90)

Attention score during sustained periods of concentration ranged between 80 and 90 on a scale of 0 to 100. The attention score was reduced significantly below pre-configured threshold levels when identifiable distractor events occurred, such as turning the head away from the monitor or prolonged eye closure. Temporal smoothing effectively reduced short-term fluctuations and provided responsive feedback to changes in attention score.

Table 2 Comparison of mediapipe and yolo-based attention models

Criterion	MediaPipe	YOLO-Based
Core Objective	Landmark detection	Object detection
Attention Modeling	Geometric + heuristic	End-to-end DL
Interpretability	High	Low
Computational Cost	Low	High
Real-Time FPS	25–30	10–20
Training Required	No	Yes
Deployment	Browser-based	GPU / server
Privacy	Local processing	Often cloud-based
Scalability	High	Limited

5.1. Overview of Compared Models

Attention modeling using MediaPipe Face Landmarker is based on a trained lightweight neural network designed specifically for efficient facial landmark recognition. The model provides 468 densely positioned facial landmarks for each frame, allowing calculation of geometric characteristics such as eye aspect ratio (EAR), head orientation parameters, and gaze direction. Instead of generating a single attention value as a black-box decision, the attention score is derived from behavioral factors by applying a rule-based scoring method. Eye openness level, blink frequency, gaze direction, and head orientation are calculated and then used to derive a continuous attention score.

YOLO refers to a family of real-time object detection models designed for fast and accurate localization of objects. The use of YOLO-based models in attention detection research involves modification of these models to detect the facial region, classify whether the eyes are open or closed, determine head orientation class, or even the attention class itself. Compared to landmark-based approaches.

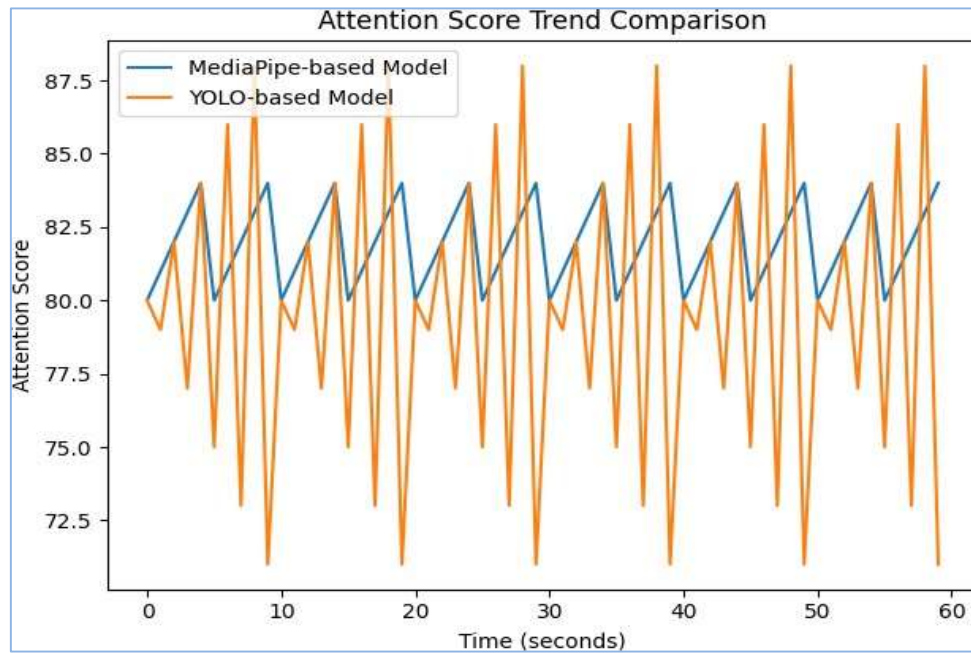


Fig. 5.1. Attention Score Trend Comparison

6. DISCUSSION

The evaluation indicates that a browser-based attention monitoring system can work effectively even with just the use of a standard webcam along with basic hardware devices. Based on dense facial landmarks provided by MediaPipe along with EAR, head pose estimation and eye gaze information, the proposed system relies on a number of signals to predict the attention state of a person.

In comparison to native or server-based systems, the proposed solution has the following advantages:

- no need for installation and cross-platform access using a web browser,
- built-in privacy because of client-side processing,

7. CONCLUSION

1. Purely Client-Side Architecture - This research proves that creating a high-quality attention monitoring system is possible without any backend computations or installations through the use of MediaPipe Face Landmarker and TensorFlow.js.
2. Real-Time and Offline Processing Capabilities - The software supports both live monitoring through the web camera as well as the offline analysis of videos that enables it to be useful for various practical cases.

3. Lightweight and User-Friendly Design - Thanks to the complete reliance on the computing power of consumer-grade hardware, the developed software is lightweight and available for everyday use.
4. Analysis of Several Features - Estimation of attention involves interpreting eye aspects ratio, direction of gaze, and approximate pose estimation of the user using their face landmarks.
5. Intuitive User Interfaces - Dashboards, notifications, and exported analytics help to interpret the behavior of the user and their level of engagement during the activity.
6. High Practical Potential - From the research findings, it can be concluded that developing an attention estimation system does not require any complex and expensive hardware solutions.

8. REFERENCES

- [1] M. Safaldin, N. Zaghdien, and M. Mejdoub, "Moving object detection based on enhanced YOLO-V2 model," in Proc. 5th Int. Congr. Human Computer Interact., Optim. Robotic Appl. (HORA), Jun. 2023, pp. 1–8.
- [2] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 8, pp. 9961–9980, Aug. 2022.
- [3] T. Soukupova and J. Cech, "Real-Time Eye Blink Detection using Facial Landmarks," in Proc. 21st Computer Vision Winter Workshop, 2016.
- [4] Akash, S., et al. "Monitoring and Analysis of Students' Live Behaviour using Machine Learning." 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2023.
- [5] R. Subramanian et al., "Multimodal Analysis of Eye Gaze, Head Pose and Facial Expression for Estimating Student Engagement," IEEE Trans. Affective Computing, vol. 13, no. 4, 2022.
- [6] V. Kazemi and J. Sullivan, "One Millisecond Face Alignment with an Ensemble of Regression Trees," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2014.
- [7] J. Redmon, J. Divvala, R. Girshick, and F. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

Biography



Anurag Kawade received bachelor's degree in Computer Engineering from AISSMS Institute of Information Technology, Pune, India, in 2026. His academic interests include artificial intelligence, machine learning, deep learning, and computer vision. He has completed courses and hands-on projects in machine learning, deep learning, computer vision, and data science.