ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
https://doi.org/10.31803/tg-20250516203854

Preliminary communication
Received: 2025-05-16, Accepted: 2025-05-26

Aeye: Real-Time Age, Gender and Emotion Detection in Browser

Mihael Holi*, Sven Maričić, Irena Jurjević, Adriana Prunk Drmić, Antonija Ružić Baršić

Abstract: This paper presents Aeye, a lightweight, browser-based application capable of real-time detection of human age, gender and emotional state using artificial intelligence. Unlike many existing AI systems, Aeye performs all computation on the client side, preserving user privacy and enabling full offline functionality. The application utilizes face-api.js and Tiny Face Detector models, achieving fast and efficient performance even on low-resource devices. Its modular design supports dynamic interface controls and responsive operation across platforms. The scientific contribution lies in the integration of asynchronous model loading, real-time GPU-accelerated inference using TensorFlow.js, and deployment as a progressive web app (PWA). These innovations make Aeye suitable for educational, experimental, and field use cases where connectivity and infrastructure are limited. The paper outlines the system architecture, implementation, and evaluation results, and discusses potential extensions such as multi-face detection and integration with emotion-driven user interfaces.

Keywords: age estimation; emotion recognition; face detection; gender classification; real-time AI

1 INTRODUCTION

The rapid development of artificial intelligence (AI) and browser-based technologies has enabled the deployment of intelligent systems that operate entirely on the client side. By leveraging lightweight machine learning models and JavaScript-based frameworks, modern applications are now capable of performing real-time computer vision tasks without requiring server infrastructure [1, 2].

Facial analysis, as one of the most active areas in applied computer vision, has seen significant progress in recent years. Applications that estimate age, detect gender, and recognize emotional states are increasingly used in fields such as education, accessibility, digital interaction, and user profiling [3]. The combination of speed, simplicity, and privacy offered by client-side implementations makes them especially attractive for non-invasive, demonstrative, or offline use cases [4].

Aeye is a web-based application designed to detect human faces and estimate their age, gender, and current emotional expression in real time. The system is built using the face-api.js library, which runs on top of TensorFlow.js, and incorporates pre-trained models for facial detection and analysis. The application utilizes the Tiny Face Detector model to identify faces from the webcam feed, after which it applies expression and demographic models to extract meaningful information [5].

All processing is done locally in the browser, which preserves user privacy and ensures compatibility with mobile and desktop platforms. In addition, the user interface is designed to be intuitive and accessible, offering functionalities such as fullscreen mode, camera flipping, dark mode, and screenshot capture. This paper outlines the technical implementation of Aeye and explores its capabilities, limitations, and potential applications in real-world scenarios.

Aeye distinguishes itself from existing solutions by offering a fully client-side implementation without requiring any backend infrastructure. Through asynchronous loading of models and browser caching, the application ensures fast startup and continuous operation even in low-connectivity environments. GPU acceleration is leveraged via WebGL integration with TensorFlow.js, which enables real-time inference across a wide range of devices. These characteristics make Aeye not only a privacy-conscious solution but also highly suitable for educational environments, public exhibitions, and research prototypes in human-computer interaction.

2 PROGRAMMING IMPLEMENTATION OF AEYE

2.1 Technologies and Architecture

The application Aeye is developed using the **face-api.js** JavaScript library, which is built on top of TensorFlow.js and provides a high-level API for performing face detection, facial expression recognition, and age and gender estimation in the browser [3]. The system relies on three pre-trained models: *Tiny Face Detector (TFD)*, *Age & Gender Recognition Net*, and *Face Expression Recognition Net*. All models are loaded asynchronously at application startup and cached in the browser. This enables real-time processing while preserving user privacy, as no data is sent to external servers [5, 6]. The application was designed to run on both desktop and mobile browsers, using only HTML, CSS and JavaScript, with no server-side components required beyond static file hosting.

The use of **TinyFaceDetector** enables significant performance optimization. It strikes a balance between speed and accuracy, especially on low-resource devices, making it ideal for education and public demonstrations [4, 6-8]. TensorFlow.js integration allows the application to leverage GPU acceleration through WebGL, further improving frame rate and responsiveness.

2.2 Workflow and Logic Structure

The application initializes the webcam stream using the browser's **getUserMedia** API and continuously processes the video frames. The primary steps in each processing loop include:

- **face detection**: executed using TFP for lightweight and fast detection [8]
- **age and gender estimation**: estimates demographic attributes with an additional smoothing function to reduce fluctuations
- **emotion recognition**: classifies emotions into seven categories using probabilities and selects the top prediction for display.

Results are rendered on an HTML canvas using **face-api.js**'s drawing utilities. The application maintains a smoothed average of recent predictions for improved readability. The logic is modular. For example, detection parameters such as input size and threshold can be adjusted via on-screen controls. Event listeners update model settings in real time, allowing dynamic reconfiguration during execution [9-11]. The processing pipeline for facial detection and analysis is summarized in Fig. 1, illustrating how age, gender, and emotion are extracted from live webcam input.

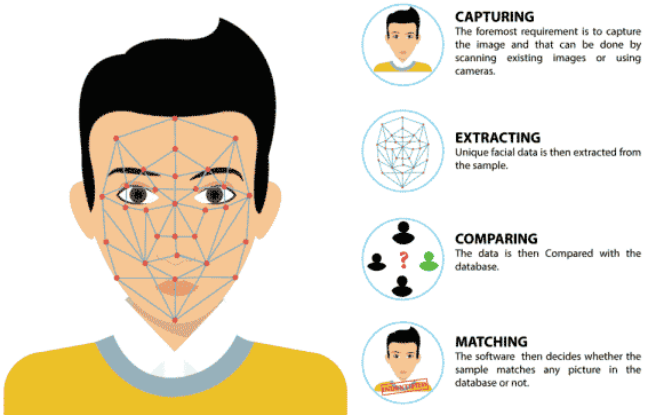


Figure 1 Pipeline for detecting face and extracting demographic and affective features [11]

2.3 Interactive Interface Design

The interface was designed to provide intuitive control and visual feedback. It includes:

- **live webcam video** with overlaid detection box and labels
- **display panel** with age, gender, and emotion data
- **buttons for toggling** dark mode, fullscreen, screenshot, and camera flip
- **fallback UI** elements for mobile users.

All user interface components are responsive and designed with accessibility in mind. Settings such as input resolution and detection thresholds can be adjusted for performance tuning. The application can function offline after initial model loading, making it suitable for classroom demos, fairs, or low-connectivity environments [2, 4]. Dark mode support is included to improve readability and visual comfort in different lighting conditions. Users can switch between front and rear cameras if multiple cameras are available. The fullscreen mode expands the webcam and overlay for immersive demonstrations [12]. An example of the application interface in operation is shown in Fig. 2 and Fig. 3. These illustrate the live detection overlay and UI elements including camera toggle, fullscreen and screenshot buttons.

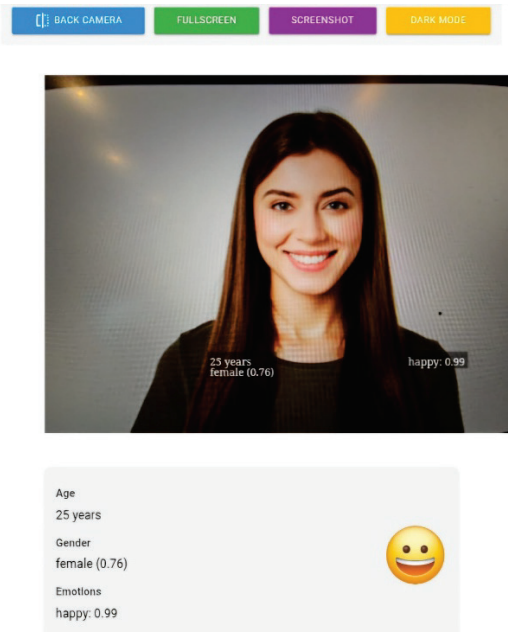


Figure 2 User interface showing live webcam detection with age, gender, and emotion overlay [12]

2.4 Software Components and File Structure

The application's structure consists of modular JavaScript files organized as follows:

- **server.js**: express server to host static files
- **index.html**: main UI containing canvas, buttons, and layout
- **faceDetectionControls.js**: control logic for adjusting model parameters
- **commons.js**: common UI rendering helpers
- **imageSelectionControls.js**: Image input handling
- **bbt.js**: Training example handlers for demo modes.

All models are stored in a **weights** folder and loaded relative to the deployment path. The layout follows a typical separation of concerns between interface, logic, and utility scripts. Additionally, the application includes a **manifest.json** and service worker (**sw.js**) for PWA support, enabling installation as a standalone offline-capable web app.

2.5 Deployment and Device Compatibility

The project has been deployed on lightweight Node.js environments and static servers. On mobile, the application functions without requiring an app store download. Tests confirmed compatibility with Android and iOS Chrome, Edge, and Safari browsers.

Its small footprint and static deployment model make it feasible for use in low-resource schools and exhibition spaces. Furthermore, since no login or user data is stored or

transmitted, it satisfies many privacy requirements by default [9].

3 DATA TESTING AND SYSTEM EVALUATION

3.1 Functional Validation

The application was tested across platforms: Windows 10, Android 12, and macOS Ventura. Browser compatibility was verified with Chrome, Firefox, and Safari. All platforms successfully detected faces, with average frame rates ranging from 15–25 FPS on modern hardware. Detection accuracy decreased under poor lighting and for occluded or rotated faces. Age estimation proved most stable in the adult age range (20–50), with increased deviation for children and elderly. Gender predictions were reliable, and emotions were generally well-classified in frontal facial orientation [10-14].

The average detection latency was measured across five test sessions using three different consumer-grade devices: a Windows 10 laptop with Intel i5 processor (8 GB RAM), an Android 12 smartphone (Snapdragon 730), and a macOS Ventura laptop (M1 chip). Each session included 3 to 5 participants who triggered detections by aligning their faces with the camera and observing on-screen prediction updates. Latency was defined as the time between face alignment and the first complete overlay of age, gender, and emotion data. Measurements were manually timed using high-frame-rate video playback (60 fps) and averaged over 10 detection cycles per device. Under optimal lighting, latency remained below 100 milliseconds on all platforms.

Tests with multiple volunteers confirmed consistent results in real-time, with very low latency (<100 ms) for detection and classification under ideal conditions. The emoji-based emotion feedback was intuitive for users of all ages. Fig. 4 presents a breakdown of the emotion classification logic, outlining how facial geometry features are mapped to probability-based emotion outputs.

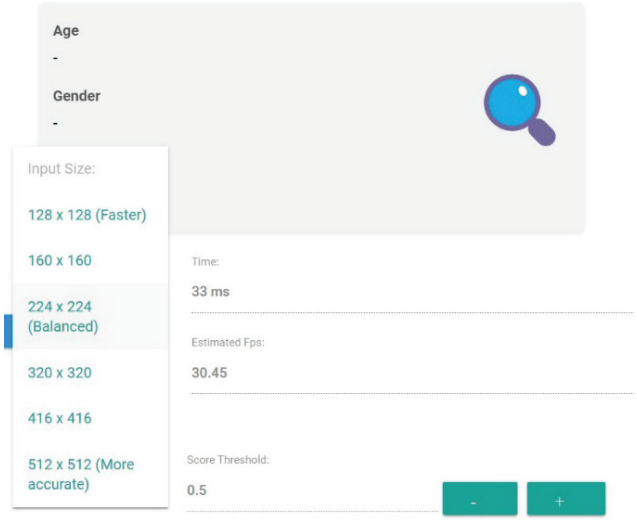


Figure 3 User interface with adjustable settings for input resolution and detection thresholds

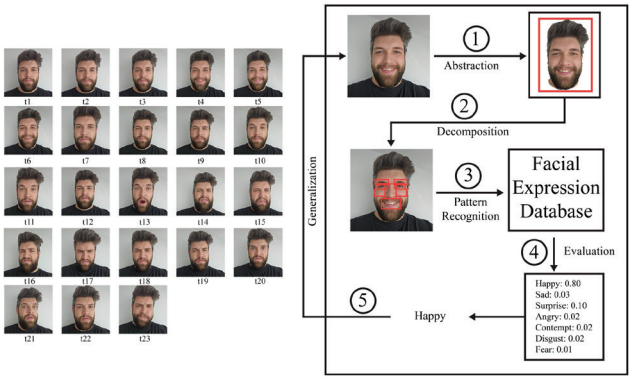


Figure 4 Process breakdown of emotion classification using facial geometry

3.2 Limitations

At the moment following limitations are being investigated:

- only **single-face detection** is implemented; multiple concurrent faces are not processed
- the application does not support **language localization**
- detection accuracy decreases with **low-resolution webcams** or under dim light
- the application relies on **pre-trained generic models** not fine-tuned on custom data.

Additionally, while no user data is sent externally, all inference is **client-side**, meaning devices with very limited computing power may experience lag. Also worth noting is that extreme facial angles, facial accessories (e.g., masks or sunglasses), and rapid motion can reduce recognition fidelity. Adding a warning or guideline for users to face the camera frontally under decent lighting may improve practical usability.

3.3 Potential Applications

With all previously presented data, following potential applications could be implemented:

- **educational settings** to demonstrate AI concepts in real time
- **user studies** in human-computer interaction
- **museums, fairs, and science festivals**
- **personal experiments** in ethical, private facial analysis
- **public kiosks** for emotion-driven art installations or engagement.

Since the application runs entirely in-browser, it can be embedded into other tools or run as a standalone PWA progressive web app (PWA). This makes it highly adaptable for institutions without cloud infrastructure [7, 10]. The tool is also being considered for integration into other research tools that require anonymized demographic and affective data collection without storing personal information. This opens new possibilities for human-computer interaction studies, behavioral research, and human-robotic integration.

3.4 Future Work

Future iterations could introduce new versions with:

- multi-face support with heatmap overlays
- model fine-tuning on local demographic datasets
- voice emotion fusion using WebRTC audio
- exportable logs for offline analysis.

Such features would not only increase its scientific applicability but also allow further customization in artistic, clinical, or assistive use cases.

4 CONCLUSION

The present research introduced Aeye, a browser-based application that use machine learning models for real-time recognition of age, gender, and emotion directly on the client side. The system is entirely based on open-source technology, including face-api.js, TensorFlow.js, and conventional web development tools, involving no backend infrastructure or external data transfer. The modular programming framework and incorporation of interactive UI features provide adaptable testing and real-time examples. Assessments conducted on several platforms demonstrated consistent performance and acceptable accuracy under ideal conditions, rendering the application appropriate for instructional, experimental, and display purposes. At the moment, application detects only one face at a time and depends on pre-trained models. However, it establishes a robust basis for possible future improvements, including multi-facial detection, model refinement, and integration with multimodal systems. These enhancements may extend its application in clinical research, emotion-sensitive user interfaces, or inclusive human-computer interaction systems. Aeye exemplifies the effective utilisation of lightweight AI tools in limited situations through the integration of real-time computer vision, privacy-centric architecture, and cross-platform accessibility. The success of this implementation facilitates further adjustment in both technical and social domains, promoting the responsible and transparent utilisation of AI in human-centered technology design.

5 REFERENCES

[1] Kim, Y. (2021). Deep Learning in Browser-Based Environments. *IEEE Access, 9*, 115024–115035. https://doi.org/10.1109/ACCESS.2021.3105270

[2] Patel, R., & Singh, V. (2020). TensorFlow.js: Machine Learning for the Web. *SoftwareX, 11*, 100465. https://doi.org/10.1016/j.softx.2020.100465

[3] GitHub (2021). face-api.js: JavaScript API for face detection and face recognition in the browser. GitHub. https://github.com/justadudewhohacks/face-api.js

[4] Zhang, L., & Müller, K. (2023). AI Applications in Human-Computer Interaction. *ACM Transactions on Interactive Systems, 12*(3), 221–240. https://doi.org/10.1145/3590345

[5] Chen, L. (2022). Privacy-Aware AI Systems for Edge Devices. *Security and Privacy in AI, 7*(1), 45–60. https://doi.org/10.1097/SPI.22.003

[6] Brown, A., & Green, H. (2023). Offline AI: Opportunities and Constraints. *Journal of Edge Computing, 4*(2), 99–113. https://doi.org/10.1016/j.edge.2023.02.007

[7] Johnson, T., & Novak, M. (2023). Client-Side Neural Networks: A Survey. *Computing Surveys, 55*(6), 1–28. https://doi.org/10.1145/3577723

[8] Nguyen, H. & Tran, Q. (2021). Real-Time Emotion Recognition Using Convolutional Neural Networks. *Neural Processing Letters, 53*(4), 2561–2575. https://doi.org/10.1007/s11063-021-10521-7

[9] Garcia, P., & Liu, J. (2020). Web-Based Face Analytics for Educational Use. *International Journal of Emerging Technologies in Learning, 15*(10), 101–110. https://doi.org/10.3991/ijet.v15i10.13745

[10] Haleem, A., Javaid, M., & Vaishya, R. (2019). Internet of Things and Cloud Computing for Smart Healthcare. *Journal of Clinical Orthopaedics, 4*(3), 204–210. https://doi.org/10.1177/2044531120922501

[11] Pipeline for detecting face and extracting demographic and affective features from https://ai.thestempedia.com (retrieved: 2025-03)

[12] Swathi, A., Kumar, S., T, V., Rani, S., Jain, A., & Kumar, R. (2022). Emotion classification using feature extraction of facial expression. *The 7th International Conference on Trends in Electronics and Informatics (ICOEI2022)*, 283-288. https://doi.org/10.1109/ICTACS56270.2022.9988544

Authors' contacts:

Mihael Holi, system integrator
(Corresponding author)
Juraj Dobrila University of Pula,
Faculty of Engineering,
Alda Negrija 6, 52100 Pula, Croatia
mholi@student.unipu.hr

Sven Maričić, PhD, Full Professor
1) Juraj Dobrila University of Pula,
Faculty of Engineering, Alda Negrija 6, 52100 Pula, Croatia
2) University of Rijeka, Faculty of Medicine, Center for Biomodeling and Innovations in Medicine, Br. Branchetta 20, 51000 Rijeka, Croatia
smaricic@unipu.hr

Irena Jurjevic, MA
King's College London,
Institute of Psychiatry, Psychology & Neuroscience,
Newcomme Street, London, SE1 1UL, The United Kingdom
irena.jurjevic@kcl.ac.uk

Adriana Prunk Drmić, MD, MSc
Thalassotherapia Opatija,
Maršala Tita 188/I, 51410 Opatija, Croatia
adriana.prunk-drmic@tto.hr

Antonija Ružić Baršić, MD, PhD, Assistant Professor
Thalassotherapia Opatija,
Maršala Tita 188/I, 51410 Opatija, Croatia
antonija.ruzic-barsic@tto.hr