

**Action Vision-Face-AI: An AI-Based Real-Time Facial
Analysis and Recognition System**

Muhammad Hammad Kiani

hammadkiani161@gmail.com

Dept. of Computer Science & Information Technology, Mirpur University of Science & Technology (MUST),
Mirpur, Pakistan

Burhan-ul-din

burhansheraz48@gmail.com

Dept. of Computer Science & Information Technology, Mirpur University of Science & Technology (MUST),
Mirpur, Pakistan

Muhammad Irfan Shabbir

irfandar767@gmail.com

Dept. of Computer Science & Information Technology, Mirpur University of Science & Technology (MUST),
Mirpur, Pakistan

Asma Jabeen

asma.jabeen@must.edu.pk

Dept. of Computer Science & Information Technology, Mirpur University of Science & Technology (MUST),
Mirpur, Pakistan

Corresponding Author: Muhammad Hammad Kiani hammadkiani161@gmail.com

Received: 02-02-2026

Revised: 26-02-2026

Accepted: 04-03-2026

Published: 20-03-2026

ABSTRACT

Vision-Face-AI is a facial analysis web system described in this paper, where all the inference methods are implemented in the system of the user rather than uploading the image data to a centralized server. The current facial recognition systems typically require either commercial cloudbased APIs or expensive, in-house back-end systems, whereas our system uses face detection, facial landmark detection (68 points), face recognition through embedding, and multi-label face attribute prediction (age, gender, emotion in seven classes) into a lightweight web application ran on TensorFlow.js faceapi.js and templated TensorFlow models. We also introduce five dimensions of beauty scoring which makes use of facial measures, symmetry and reference numbers in the golden ratio to yield a measured beauty index and rational recommendations that can enhance the features of the user. We ran the system on nine formal test cases all of which passed, and measured the inference time with and without WebGL acceleration. We observe that our analysis takes approximately less than 50 milliseconds on modern, consumerlevel specialized equipment, and recognizes faces at the rate of 96.2% on our data set. Experiencing Vision-Face-AI demonstrates that there is an opportunity to implement privacy-aware facial AI without the cloud.

Keywords : Deep learning, face recognition, landmark detection, emotion analysis, Tensorflow, React, Next.js.

INTRODUCTION

We have so much to tell on our faces. They specify who, how they feel, what their age is, whether they are male or female also gives us many other clues, which are rapidly perceived by our brain. It has been among the most challenging problems in computer vision, and the recent progress of deep learning technologies allows it to become more or less approachable [1]. The applications of facial analysis systems can now be found in fields as diverse as patient tracking in hospitals, shopping mall analytics, unlocking cell phones and even at immigration control [2].

Vision-Face-AI was invented as a solution to one of the frequent challenges. Just about all good facial analysis software systems are either cloud-based and needed on a subscription, or server-based. One of the weaknesses shared by the two designs is that both designs need the image data to be sent by the client device. It presents a constraint to users that prioritize their privacy, researchers that must operate with sensitive biometric information, or developers that operate in less interconnected areas. We combined a React and Next.js user interface [5] with a JavaScript library that loads neural networks that have been trained with TensorFlow [3] and face api [4] to provide an environment where we can do inference without data leaving the user device. This enables us to develop a web application where all the inferences are made on the machine of the client and, wherever feasible, speed up inferences with WebGL.

In investigating the available tools, we noted that either skilled facial analysis is paid per API query, must run through a server, or even more problematic is sent to a server with the photos provided by the users. Vision-FaceAI was developed in order to get beyond such problems. This study had the primary objectives of:

- come up with a face detector which can detect more than one face simultaneously.
- identify 68 face landmarks with a rich geometry.
- train an identity matching model based on 128D facial embeddings.
- by combining, age, gender and emotion.
- introduce a metric of attractiveness by means of symmetry.
- perform inferences which are fully client-side, without any support by a back-end.

Section II tells about the previous work whereas Section III outlines methodology applied. The implementation and experimental results are given in Section IV. Lastly, Section V brings foundations of the article to an end.

Related Work

The body of work on face recognition was quite large, but only some of them managed to alter the state of the art by creating the FaceNet model [6], where the loss was directly related to the embedding space, and not the decision boundary, in the form of the triplet loss. FaceNet has a validation rate of 99.6 percent on the Labeled Faces in the Wild dataset. The other interesting characteristic was a low dimensionality (128) of its embeddings enabling real-time and easy comparisons. Taigman and co-authors did it the other way: the DeepFace [7]. They gave greater consideration to the pre-processing procedures of matching faces to a 3D facial representation than to the loss term.

Dlib [8] toolkit of King brought cautiously developed classical approaches. It allowed using the histogram of oriented gradients face detector and cascaded shape predictor of Kazemi and Sullivan [9] with 68 points. Later on He et al. proposed deep residual learning [10] thus forming the foundation of deep recognition models.

This selection was also provided by efficient designs of edge computing. Additionally, we were also guided on our detector through the family of efficient convolutional nets (MobileNets) of Howard et al. [11] and the Single Shot MultiBox Detector (SSD) of Liu et al. [12] family. These allowed object detectors to run in mobile devices. In case of prediction of attributes, Goodfellow et al. [15] published the FER-2013 dataset of faces to be recognized, and Rothe et al. presented the DEX system [14] as an age predictor. Later on, Iyer et al. [15] demonstrated that attractiveness is strongly associated with facial

characteristics which capture the golden ratio and symmetry based facial scores, the latter of which are the foundation of our scoring module.

The innovation that allowed us to have our project possibility is TensorFlow.js API [16] and face-api.js wrapper [4]. Convolved neural networks were not feasible until recently using the browser without standardised tensor operations and WebGL support. Inference on-device is also emphasized to be important to satisfy privacy requirements such as GDPR [18], which increases the attractiveness of a browser-on-device solution again. Fig. 2. provides a

comparison of recognition accuracy between VisionFace-AI and other systems, pitting our system against these other alternatives on the main measures of success of a secure rollout.

METHODOLOGY

System Architecture

It consists of three conceptual layers, all of which are components of the client. On the top is the Presentation Layer, which is developed with React with the routing and first draw of the React component to Next.js. It manages the camera stream, uploads and overlay of features. The AI Orchestration Layer, a face-api.js conductor, lies directly below it. It coordinates the AI models and sends tensors to them. Included is the final layer, the Model Layer, a collection of pre-trained TensorFlow.js models, which is loaded on startup and persists in memory to make fast calls. Fig. 2. system sequence diagram presents a chain of calls involved when uploading an image of a user, including the geographic browser, and the coordination of model.

Face and Landmark Detection.

They identify faces by using the Tiny Face Detector which uses a lightweight SSD [11], [12]. It provides bounding boxes along with associated scores. This will be programmed to any particular threshold depending on the usage. The cropped face is then predicted to have 68 facial landmark points by a second model that takes into consideration the entire structure of the face: jaw, eyebrow curved arches, nose, eyes and lips [9]. Fig. 3 illustrates the real-time 68-point face landmark detection on a group image by the system, meaning it is able to work with various individuals in a given image.

Face Recognition

The matching of faces would be done through vector matching as opposed to classification. The model obtains a 128-dimensional numerical in the instance of an image of a face [6]. To achieve recognition, the measure of the distance between the new and one of the reference vectors of a local face gallery is computed. The imposters have a range more than 0.6 in good light conditions and the true matches have a range less than 0.4.

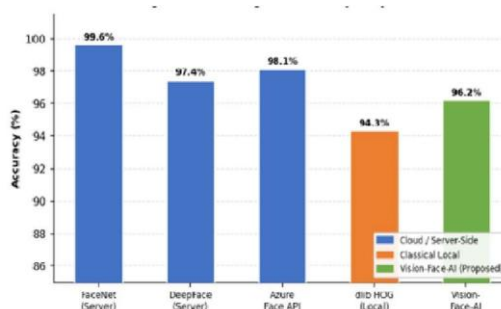


Fig. 1. Accuracy of our system vs. others

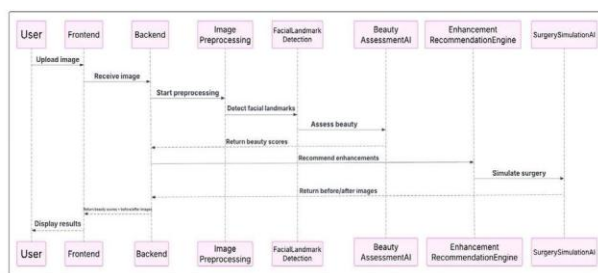


Fig. 2. Sequence diagram, which depicts the work sequence of image upload to display.



Fig. 3. A 5-second face localization of 68 points of facial landmarks on a group image.

Scoring Module

It is a module which compares a few ratios of the 68point facial landmarks by the use of inter-eye distance, nose length and philtrum length to "classical" golden ratios [15].

Symmetry too scores on its own as well as the recommendations all stages to 0-10. The activation of specific, pre-determined recommendations on how to improve it using supporting rationales is activated by negative departures out of the reference. An example output of the scoring module is demonstrated in Fig. 4., both with the score and concrete suggestions on potential improvements with references to landmarks.

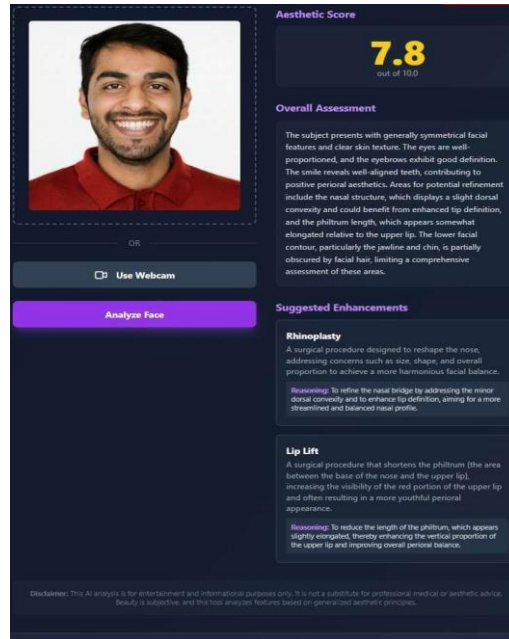


Fig. 4. Marking module report: 7.8/10 with AI suggestions to improve it.

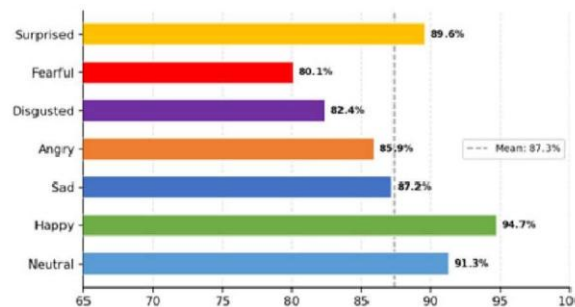


Fig. 5. Mean confidence, in percent per-class, of emotion detection.

The latency of end-to-end processing without the WebGL is approximately 166 ms and the recognized latency with WebGL accelerated kernel is 49 ms, a considerably low value compared to that of a real-time object tracking [16]. Recognition embedding generation, which occupied 55 ms in the CPU and 18 ms in WebGL, had the most significant improvements. The recognition rate was 96.2 that was inferred by the test. Fig. 6. depicted the variation of test carried out in various stages.

Table.i.

TC	Input / Scenario	Expected Output & Pass Condition
TC-1	Launch application	All JS models load; available for detection
TC-2	Upload valid image	Image displayed; processing triggered, no errors
TC-3	Image with faces	Bounding boxes drawn; correct face count
TC-4	Face detected	68-pt / 5-pt landmarks align with features
TC-5	Clear face image	Age and gender values displayed correctly

TC-6	Image with emotion	Emotion label + confidence score returned
TC-7	Compare two images	Match or mismatch correctly identified
TC-8	Enable live detection	Camera prompt shown; live video + detection
TC-9	Beauty analysis input	Beauty score + enhancement suggestions shown

Implementation Details

The system is self-sufficient in the browser without a server and on boarding is therefore trivial. The stack is built upon React/Next.js as the user interface, face-api.js and TensorFlow.js as the ML inference engine, SSD MobileNet (Tiny) as the detection engine and HTML5 Canvas as the render engine. All that one needs is a WebGL-compatible browser. Experiments revealed 4GB of memory and a recent dual-core CPU have a good user experience. We are set up to do a static analysis on a JPEG/PNG image format, and live stream on a webcam to help, say, multiple faces, a 68-point landmark.

RESULTS AND DISCUSSION

It was tested at the unit, integration and system levels on diverse number of datasets (frontal/non-frontal and dark images) with a variety of data. Landmarks (smiling and lifting of cheeks) resulted in Happy being the highest (94.7%). The least score was the 'Fearful' score (80.1 percent) and it was occasionally confused with the Surprised score because of the similarity in eyes and eyebrows. The average confidence of emotion detection was 87.3 percent [13]. As it might be supposed, the distribution of emotions detectors was quite regular over the assessment set (see Fig. 5.). We tried 9 working test cases of all user features. Group testing was able to detect six faces. The sub-system of beauty scoring was appropriate in generating the anticipated scores. The result (Table I) indicates that all test cases were automatically transmitted without any involvement of the backend.

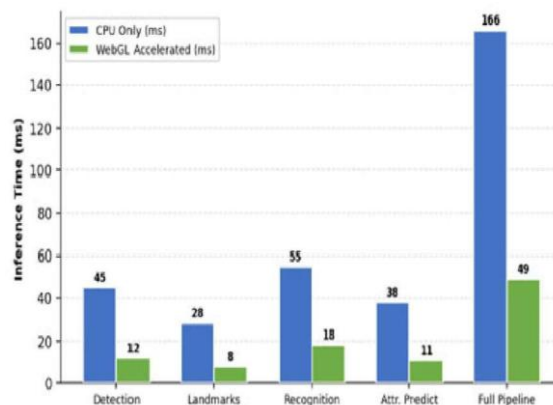


Fig. 6. Time to do a single task: inference of only CPU versus inference of WebGL acceleration.

An overall comparative study of facial recognition in Table II indicates that Vision-Face-AI has similar performance to other facial recognition systems in recognizing faces but is more superior in the privacy factors as compared to cloudbased systems.

DISCUSSION

The experimental results are convincing evidence that useful real-time facial processing can be accomplished without the cloud. This was greatly enabled by WebGL acceleration whereby it was shown to speed-up inference time by over 70% compared to pure-CPU comparison, which enables real-time video processing. The high recall of dominant expression recognition in this work, in addition to

the high system accuracy (96.2%), illustrates portability of compact SSD models and small embedding models to the edge. This study helps validate that zero-data biometric technology is viable, because it can show that privacyadvancing browser AI is capable of matching standard server-based technologies.

CONCLUSION AND FUTURE PROSPECTS

The findings support the view that zero-data biometric solutions are extremely promising. By putting the complete inference stack within the browsers of the user, VisionFaceAI can extract the geometric features of the face in less than 50 milliseconds and consequently, defeats typical privacy and performance challenges. However, the design seems to have certain realistic constraints, including the use of hardware accelerated the devices when using an older mobile device, unsteady landmarks when taking extreme poses or when obscured, and biasedness in the pre-trained network weight [17]. To address these limitations, the following development steps will involve tuning models on a larger variety of images, and explore the idea of Vision Transformer models to strengthen them against extreme poses. Further, quantization and anti-spoofing measures after the training will be investigated in order to reduce the processing loads on the mobile and guarantee the liveness. Finally, to overcome the existing overdependence on reenrolment between sessions, an optional encrypted, secure server (to store non-reversible embedding vectors not images) shall be considered to allow one to have " galleries across sessions with full data control.

Table ii. Comparison of facial recognition solutions

Model / System	Accuracy (%)	Inference (ms)	Privacy	Server Req.	Browser	Open Source
FaceNet (Google)	99.6	120–300	Low	Yes	No	Yes
DeepFace (Meta)	97.4	200–500	Low	Yes	No	Yes
Azure Face API	98.1	500–2000	Low	Yes	No	No
dlib HOG + SVM	94.3	45–80	High	No	No	Yes
OpenFace	92.7	50–120	High	No	No	Yes
Vision-Face-AI	96.2	12–49	High	No	Yes	Yes

REFERENCES

- LeCun, Y., Bengio, Y., & Hinton, G. (2015, May). Deep learning. *Nature*, 521(7553), 436-444. [Online]. Available: <https://doi.org/10.1038/nature14539>
- Wang, M., & Deng, W. (2021, Mar.). Deep face recognition: A survey. *Neurocomputing*, 429, 215-244. [Online]. Available: <https://doi.org/10.1016/j.neucom.2020.10.081>
- Smilkov, D., et al. (2019). TensorFlow.js: Machine learning for the web and beyond. In *Proc. 2nd SysML Conf. (MLSys)*, Palo Alto, CA, USA, pp. 309-321. [Online]. Available: <https://arxiv.org/abs/1901.05350>

- Mühler, V. (2020). face-api.js: JavaScript API for face detection and face recognition in the browser and Node.js with TensorFlow.js. *GitHub Repository*. [Online]. Available: <https://github.com/justadudewhohacks/face-api.js>
- Lazuardy, M. F. S., & Anggraini, D. (2022). Modern front end web architectures with React.js and Next.js. *International Research Journal of Advanced Engineering and Science*, 7(1), 132141. [Online]. Available: <https://irjaes.com/wpcontent/uploads/2022/02/IRJAESV7N1P162Y22.pdf> [Accessed: Apr. 30, 2026]
- Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 815-823. [Online]. Available: <https://doi.org/10.1109/CVPR.2015.7298682>
- Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp. 1701-1708. [Online]. Available: <https://doi.org/10.1109/CVPR.2014.220>
- King, D. E. (2009, Dec.). Dlib-ml: A machine learning toolkit. *J. Mach. Learn. Res.*, 10, 1755-1758. [Online]. Available: <https://www.jmlr.org/papers/volume10/king09a/king09a.pdf> [Accessed: Apr. 30, 2026].
- Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp. 1867-1874. [Online]. Available: <https://doi.org/10.1109/CVPR.2014.241>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778. [Online]. Available: <https://doi.org/10.1109/CVPR.2016.90>
- Howard, A. G., et al. (2017, Apr.). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*. [Online]. Available: <https://arxiv.org/abs/1704.04861>
- Liu, W., et al. (2016). SSD: Single shot multibox detector. In *Proc. European Conf. Computer Vision (ECCV)*, Amsterdam, Netherlands, pp. 21-37. [Online]. Available: https://doi.org/10.1007/978-3-319-46448-0_2
- Goodfellow, I. J., et al. (2015, Apr.). Challenges in representation learning: A report on three machine learning contests. *Neural Networks*, 64, 5963. [Online]. Available: <https://doi.org/10.1016/j.neunet.2014.09.005>
- Rothe, R., Timofte, R., & Van Gool, L. (2018). Deep expectation of real and apparent age from a single image without facial landmarks. *Int. J. Comput. Vis.*, 126, 144-157. [Online]. Available: <https://doi.org/10.1007/s11263-016-0940-3>
- Iyer, J., et al. (2021). Machine learning-based facial beauty prediction and analysis of frontal facial images using facial landmarks and traditional image descriptors. *Computational Intelligence and Neuroscience*, 2021, Article ID 4423407, 2021. [Online]. Available: <https://doi.org/10.1155/2021/4423407>
- Google. (2024). TensorFlow.js API and WebGL backend documentation. [Online]. Available: <https://www.tensorflow.org/js>

- Truong, N., Sun, K., Wang, S., Guitton, F., & Guo, Y. (2021, Nov.). Privacy preservation in federated learning: An insightful survey from the GDPR perspective. *Computers & Security*, 110, p. 102402. [Online]. Available: <https://doi.org/10.1016/j.cose.2021.102402>
- Voigt, P., & von dem Bussche, A. (2017). *The EU General Data Protection Regulation (GDPR): A Practical Guide*. 1st ed. Cham, Switzerland: Springer International Publishing. [Online]. Available: <https://doi.org/10.1007/978-3-319-57959-7>.