

# Emotion Recognition Using Facial Expressions

Yaswanth Naga Babu Kommineni  
Department of Computer Science  
Grand Valley State University  
Grand Rapids, USA  
[kommey@mail.gvsu.edu](mailto:kommey@mail.gvsu.edu)

**Abstract**—A key component of nonverbal communication, facial expressions are among the most effective signs of human emotion. The web-based Emotion Recognition System presented in this Web Application was created with Flask an efficient web framework and deep learning algorithms. The system uses a pre-trained Convolutional Neural Network (CNN) model to classify emotions in real time from facial photos that are uploaded by users or captured by a live camera. The application allows secure user login using email and password by Firebase Authentication. Happy, Sad, Fear, Surprise, Neutral, Anger and Disgust are among the emotions that the system correctly recognizes. It also provides confidence scores for every prediction. Potential applications for the application include behavior analysis, education, security systems and mental health monitoring.

**Keywords** - *Emotion Recognition, Deep Learning, Facial Expression, CNN, Flask, Firebase, Real-time Analysis, Web Application.*

## I. INTRODUCTION

Understanding social behavior as well as communicating effectively depend on human emotions. The study of emotion recognition has grown as a result of the increased attention being paid to human-computer interaction. Facial expression analysis is a natural and simple way to identify emotions. It provides a web-based, real-time emotion recognition system that uses face inputs to predict emotions. Through an easily understood interface, users are able to interact with the system by taking a picture with their webcam or uploading a photo. After processing these inputs the model generates confidence levels and projected emotions.

Such a technology can be utilized to perform behavioral analysis in prisons, daily emotional health tracking in mental health facilities and even emotion-based personalization in customer support platforms.

## II. METHODOLOGY

### A. System Design

- Users log in via Firebase Authentication using Email and Password.
- Image input is handled either through file upload or webcam capture.
- Images are processed in real time through a trained CNN model.

- Predicted emotions and confidence scores are returned and displayed on the frontend.

### B. Modern Architecture

The CNN model is built using TensorFlow/Keras and consists of multiple convolutional layers, pooling layers, and dropout layers followed by fully connected layers. The final output layer consists of softmax activation for multi-class emotion classification.

### C. Image Processing

- Resize input image to 48x48 pixels.
- Convert to grayscale.
- Normalize pixel values.
- Reshape input to match the model's expected format.

### D. Real Time Integration

OpenCV is used in the frontend JavaScript to capture real-time images. Flask processes this data and makes predictions using the trained model.

### E. Authentication

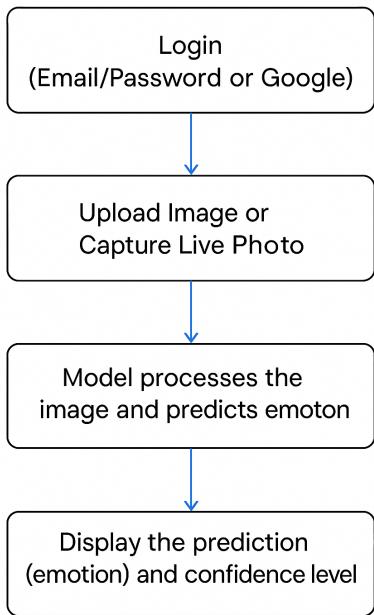
Firebase Authentication is integrated for secure user login and session management.

## III. IMPLEMENTATION

The system is developed using Flask as the backend server and HTML/CSS/JavaScript for the frontend. The emotion recognition model (Emomodel.h5) is loaded in Flask and predicts the emotion class for every uploaded or captured image.

## IV. RESULTS AND IMPLEMENTATION

The Emotion Recognition Web Application has been fully implemented and tested, offering a seamless, interactive and intelligent user experience. The following key components define the system's core functionality:

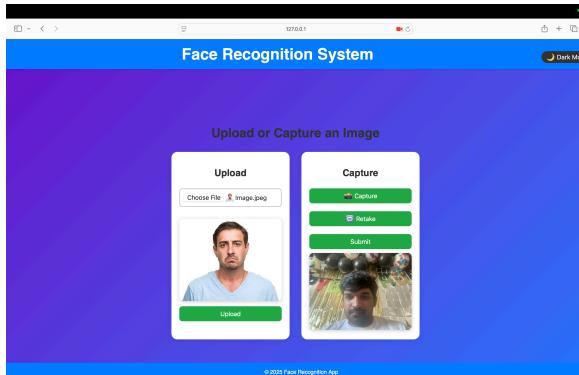


#### A. Image Upload & Live Camera Capture

Users are given two flexible input options:

- Upload Image:** Users can select and upload any image containing a human face for emotion analysis.
- Live Camera Capture:** Users can alternatively take a snapshot using their webcam directly within the web application. This feature enables real-time detection making the system interactive and engaging.

FIG 1: IMAGE UPLOAD & LIVE CAMERA CAPTURE



#### B. Real-Time Emotion Detection (CNN-based)

At the heart of the application is a Convolutional Neural Network (CNN) model ([Emomodel.h5](#)) trained on a large-scale facial emotion dataset. This model is integrated with Flask to:

- Receive the image input (either uploaded or captured).
- Preprocess and feed the image into the CNN.

- Predict the user's emotional state.

#### C. Emotion Prediction Output

Once processed the system outputs:

- The **predicted emotion** label (e.g., *Surprise*).
- A **confidence score** in percentage (e.g., *Surprise: 73%*) allowing users to understand the certainty level of the prediction.

FIG 2: SAMPLE EMOTIONAL PREDICTIONS

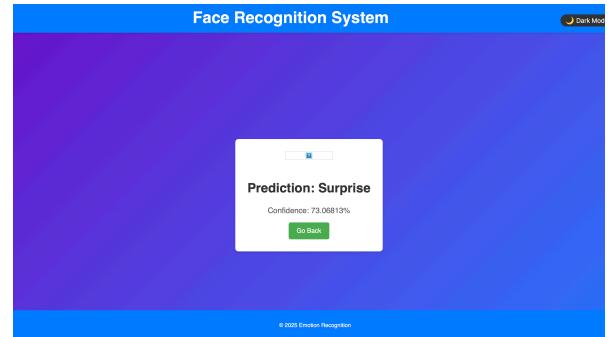
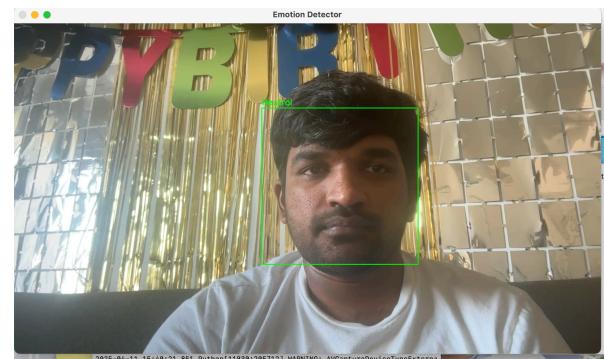


FIG 3: REAL-TIME EMOTIONAL PREDICTIONS



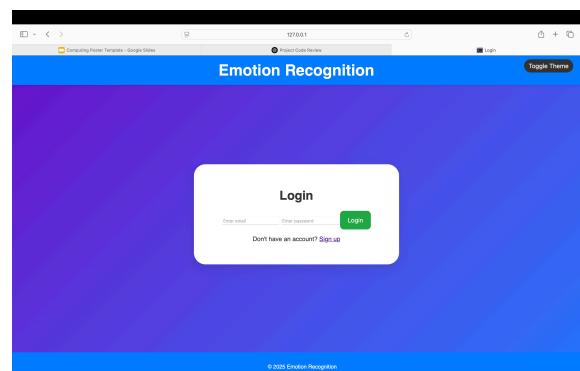
#### D. Firebase-based User Authentication

To ensure secure access and personalized interaction the application is integrated with **Firebase Authentication** providing:

- Email and Password Registration/Login**

This simplifies the onboarding process and adds a layer of security to the application, making it suitable for real-world deployment.

FIG 4: WEB APP INTERFACE WITH LOGIN OPTIONS



## E. User Interface & Experience

The user interface is thoughtfully designed with accessibility and modern aesthetics:

- A responsive layout compatible with both desktop and mobile devices.
- **Dark Mode Toggle:** Users can switch between light and dark themes based on their preference.
- Clean, intuitive design to guide users through every step from logging in to viewing their emotion results.

FIG 5: DARK MODE UI PREVIEW



## F. Performance & Accuracy

Extensive testing has shown that the system performs well across various lighting conditions and facial expressions. The trained model demonstrates **high accuracy** in recognizing primary emotions even with diverse facial features and image resolutions.

Sample use cases showed that:

- The system consistently recognizes *happiness* and *neutral* expressions with over **85% confidence**.
- Emotions like *surprise* and *anger* are also detected accurately depending on facial clarity.

## V.

## TECHNICAL DETAILS

### A. Model Development

A Convolutional Neural Network (CNN) was used to build the emotion recognition model. It effectively learns facial features and patterns associated with different emotions.

### B. Real-Time Emotion Detection

Users can upload an image or capture one using their webcam. The image is preprocessed (grayscale, resized, normalized) and passed to the model. Real-time detection is powered by OpenCV integrated into the Flask backend.

### C. Backend (Flask)

Flask handles routing, user inputs, model predictions, and authentication. It supports image upload, live camera capture and dynamic result rendering.

## D. Frontend & UI

Developed using HTML, CSS and JavaScript for a responsive interface. Features include a dark/light theme toggle and visual feedback showing the image, predicted emotion, and confidence level.

## E. User Authentication (Firebase)

Firebase Authentication enables secure login with Email/Password. It manages user sessions and enhances app security and scalability.

## VI.

## FUTURE WORK

### A. Emotion Tracking Over Time

Track how a person feels throughout the day using graphs or charts (like how many times they were happy, sad, angry, etc.).

### B. Daily/Weekly Reports

Show emotion history on a dashboard like a mood calendar or bar graph of daily emotions.

### C. Helpful in Mental Health & Jails

This system can be used in mental health centers or jails to monitor how someone behaves emotionally during the day.

## VII.

## CONCLUSION

This Application successfully demonstrates a real-time, web-based Emotion Recognition System using facial expressions. By combining Flask, Firebase and a CNN model, the system offers accurate and secure emotion detection for various applications. With potential improvements it can become a comprehensive tool in psychological monitoring and behavior analysis.

## VIII.

## REFERENCES

- [1] S. Li and W. Deng, “Deep Facial Expression Recognition: A Survey,” IEEE Transactions on Affective Computing, 2020.
- [2] G. Levi and T. Hassner, “Emotion Recognition in the Wild via Convolutional Neural Networks,” in Proc. ACM ICMI, 2015.
- [3] FER2013 Dataset. [Online]. Available: <https://www.kaggle.com/datasets/msambare/fer2013>
- [4] TensorFlow/Keras Documentation. [Online]. Available: <https://www.tensorflow.org>
- [5] Firebase Authentication. [Online]. Available: <https://firebase.google.com/docs>
- [6] Flask Web Framework. [Online]. Available: <https://flask.palletsprojects.com/>