Emotion Recognition Using Facial Expressions

Yaswanth Naga Babu Kommineni

A Project Submitted to

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Applied Computer Science

School of Computing

April 2025

The signatures of the individuals below indicate that they have read and approved the project of Yaswanth Naga Babu Kommineni in partial fulfillment of the requirements for the degree of Master of Science in Applied Computer Science.

_____

Jagadeesh Nandigam, Project Advisor Date

_____

Robert Adams, Graduate Program Director Date

_____

<name of unit head>, Unit head Date

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.

Introduction

Understanding human emotions has always been a fundamental part of communication, connection, and decision-making. In today's world, where technology plays a central role in our daily interactions, the ability to recognize and interpret emotions automatically has gained increasing importance. With this in mind, the goal of this project was to design and develop a web-based **Emotion Recognition System** that can detect a user's emotional state simply by analyzing their facial expressions.

The deliverable for this project is a complete web application that allows users to either upload an image or capture a live photo using their webcam. Once an image is submitted, it is processed by a pre-trained deep learning model to predict the user's emotion in real time, providing both an emotion label and a confidence score. The system is built using Flask for backend processing and integrates Firebase Authentication for secure user login, ensuring that the platform is both functional and safe.

The motivation for undertaking this project stems from the increasing demand for technologies that can better understand human behavior. Emotion recognition has valuable applications across a wide range of fields — from mental health monitoring and education to customer service and security. Being able to track emotional patterns could provide support for early interventions in healthcare, improve learning environments, and create more responsive customer experiences.

This project not only applies key concepts learned during the Master's program — such as deep learning, machine learning, web development, and cloud services — but also builds new skills,

particularly in real-time computer vision integration and cloud-based system design. Developing a project that brings together artificial intelligence, user interaction, and secure authentication challenged me to think critically about both the technical and ethical considerations of emotion analysis.

The primary objectives of this project were to:

- Build a real-time, secure and user-friendly platform for emotion recognition.
- Train and deploy a Convolutional Neural Network capable of recognizing multiple emotions such as Happy, Sad, Fear, Surprise, Neutral, Anger and Disgust.
- Create an user interface that supports image uploads and live webcam capture.
- Apply cloud-based authentication to manage user security.

By meeting these goals the project demonstrates both the practical application of academic learning and the importance of designing technology that understands and responds to human emotions. It highlights the exciting intersection between artificial intelligence and real-world needs an area that will only continue to grow in relevance.

Project Management

This project was managed through weekly planning, focusing on small, achievable goals each week. The work was divided into different phases: setting up the environment, collecting data, preprocessing, training the model, developing the backend and frontend, integrating webcam functionality and final testing.

At the start, a **project backlog** was created listing all major tasks such as environment setup, model building, backend development and frontend design. Every week tasks were selected from the backlog and completed step-by-step. If new needs or problems appeared during the project the backlog was updated to keep the project organized.

Here is a brief overview of how the project moved week-by-week:

- **Week 1–2:** Environment setup, library installation and data collection.
- **Week 3:** Preprocessing and feature extraction.
- **Weeks 4–5:** Model training and initial testing.
- **Week 6:** Model saving and prediction testing.
- **Week 7:** Backend development with Flask.
- **Weeks 8–9:** Frontend design and improvement.
- **Week 10:** Webcam integration and final optimization.

Organization

The Emotion Recognition System is designed as a standalone web-based application, consisting of three major components: the frontend interface the backend server and the deep learning model for emotion detection. These components work together to deliver real-time emotion predictions based on user-submitted facial images.

System Architecture

- **Frontend:**

  The user interface is built using HTML, CSS and JavaScript. It allows users to either upload an image or capture one through their webcam. The interface also supports dark mode for improved accessibility. Once an image is selected or captured it is sent to the backend server through HTTP POST requests.

- **Backend:**

  The backend is developed using Flask a lightweight Python web framework. It receives images from the frontend, preprocesses them (grayscale conversion, resizing, normalization) and passes the processed images to the pre-trained Convolutional Neural Network (CNN) model for emotion prediction. After processing the backend sends the prediction result (emotion label and confidence score) back to the frontend for display.

- **Model:**

  The deep learning model (Emomodel.h5) is a Convolutional Neural Network trained to

classify facial expressions into seven emotion categories: Happy, Sad, Neutral, Fear, Angry, Surprise and Disgust. The model is loaded into memory when the Flask server starts and is used for real-time predictions.

- **Authentication:**

  Firebase Authentication is integrated to provide secure login functionality. Users must sign up or log in with an email and password to access the system. Authentication tokens are managed securely to maintain user sessions.

Interfaces

- **Frontend to Backend Interface:**

  The frontend communicates with the Flask backend through HTTP POST requests. Images are encoded and sent as part of the form data. The backend processes the image and returns a JSON response containing the predicted emotion and the associated confidence score.

- **Backend to Model Interface:**

  The backend handles image preprocessing and interacts directly with the loaded CNN model. Predictions are made internally without external API calls ensuring quick response times.

- **Backend to Authentication (Firebase) Interface:**

  User credentials and login sessions are managed through Firebase's secure authentication system which integrates with the Flask server using secure tokens.

Component Location

Since this is a standalone application all major components frontend, backend and model are hosted and executed on the same local server. No distributed architecture is used in the current version. However the system architecture allows future expansion to cloud hosting and distributed processing if necessary.

Reflection

This project set out to develop a real-time web-based Emotion Recognition System that could classify human emotions from facial expressions. Reflecting on the initial objectives, I am pleased to report that the project achieved most of the planned goals successfully. The application allows users to upload or capture images processes them using a trained deep learning model and returns accurate emotion predictions with confidence scores. It also includes user authentication through Firebase and provides a clean user-friendly interface with light and dark mode options.

Strengths and Accomplishments:

- The integration of the frontend and backend worked smoothly creating a seamless experience for users.
- The CNN model demonstrated good accuracy on unseen test images showing the effectiveness of the training process.
- Real-time webcam integration was achieved allowing live emotion detection which significantly enhanced the practical utility of the system.
- Firebase Authentication was successfully incorporated providing secure login and registration functionality.
- The user interface was designed to be simple, accessible and mobile-friendly, improving usability across devices.

New Learning:

Throughout this project I deepened my knowledge of several important areas including real-time image processing, model deployment with Flask, frontend-backend integration and cloud authentication systems. Additionally, handling real-time video data and optimizing model performance for responsiveness taught me valuable lessons in system design and optimization. Managing the entire application lifecycle from data preparation to deployment also improved my practical project management skills.

Challenges and Weaknesses:

- Initially, setting up the webcam integration was challenging due to performance issues and slow frame rates under different lighting conditions. Several rounds of optimization were needed to make it reliable.

- Early versions of the model showed lower accuracy for certain emotions particularly “fear” and “disgust.” This required additional preprocessing and data augmentation to improve performance.

- Time management was sometimes difficult, especially when unexpected bugs appeared in the integration phase. Some features like tracking emotion changes over time or logging user emotions were considered but had to be postponed for future work due to time constraints.

Adequacy, Efficiency and Effectiveness:

Overall, the system is adequate for its intended purpose of recognizing basic human emotions

from static images or live webcam feeds. The system is efficient enough to provide predictions within a second making it suitable for real-time applications. It is effective in detecting and classifying common emotions and offers a good user experience through a responsive and intuitive interface.

Conclusions

The Emotion Recognition System project successfully achieved its main objectives. A fully functional web-based application was developed that allows users to upload or capture images through a webcam, predicts emotions in real time using a trained Convolutional Neural Network (CNN) model, and displays results with confidence scores. Secure user authentication was implemented through Firebase, and the user interface was designed to be responsive, user-friendly and accessible, including a dark mode feature for enhanced usability.

Through this project I applied and expanded knowledge gained during the program including deep learning, computer vision, web development, and cloud-based authentication. The integration of these technologies into a seamless and interactive application demonstrated the practical application of academic concepts to a real-world problem.

Future Work

- **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.).

- **Daily/Weekly Reports**

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

- **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.

Appendices

Appendix A: Simple User Manual

1. Logging In:

- Navigate to the login page.
- Enter your registered email address and password.
- Click “Login” to access the main system.
- If you do not have an account click “Sign Up” to register.

2. Uploading an Image:

- After logging in, click on the “Upload” section.
- Choose an image file (.jpg, .jpeg, .png) containing a clear face.
- Click “Upload” to submit the image.
- The system will display the predicted emotion with a confidence score.

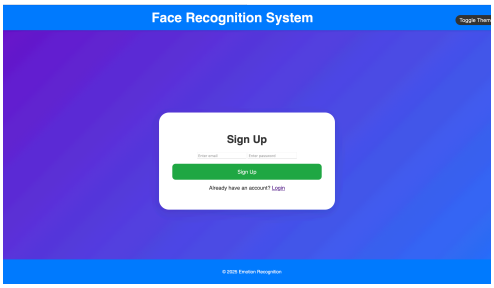
3. Capturing an Image Using Webcam:

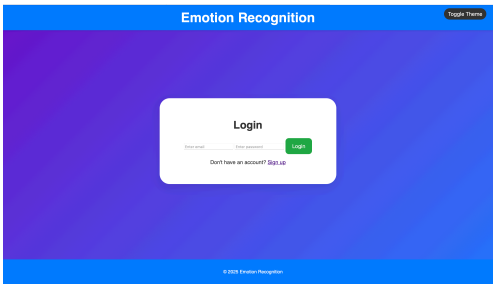
- Allow browser access to your webcam when prompted.
- Click the “Capture” button to take a live photo.

- Click “Submit” to analyze the captured image.
- The predicted emotion and confidence score will appear.

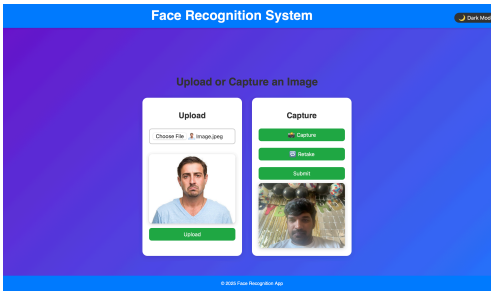
Appendix B: Screenshots

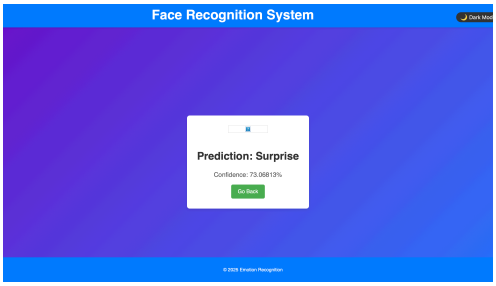
- **Fig. A1:** Home page (Login and Signup interface)



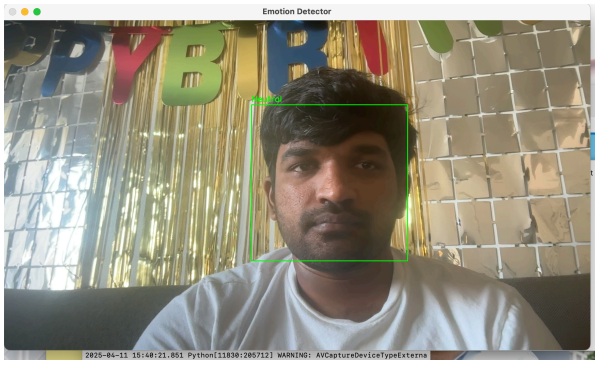


- **Fig. A2:** Image Upload and Prediction Result





- **Fig. A3:** Webcam Live Detection Preview



- **Fig. A4:** Dark Mode View of the Application