International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

Mood Map

Avs Radhika[1], S Ashrutha[2], M Jyothi[3], G Mani Kumari[4]
[1]Associate Professor; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India.
[2,3,4]B.Tech Students; ; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India.
Mail Id; jyothimudavath2005@gmail.com[3],salvojuashrutha@gmail.com[2], gmanikumari789@gmail.com[4]

Abstract
With the rapid advancement of artificial intelligence, facial emotion recognition has emerged as a key component in human–computer interaction. Studies suggest that non-verbal communication, particularly facial expressions, contributes significantly more to interpersonal communication than verbal interaction. Facial Emotion Recognition (FER) systems are designed to automatically identify emotional states by analyzing facial features, thereby providing insights into human behavior and psychological conditions. This paper presents MoodMap, a real-time facial emotion recognition system capable of detecting seven fundamental emotions, namely happy, sad, angry, fear, surprise, neutral, and disgust. The proposed approach integrates YOLOv2 for efficient face detection and SqueezeNet for emotion, age, and gender classification. YOLOv2 enables accurate localization of faces in real time using anchor-based detection, while SqueezeNet extracts meaningful high-level features with reduced computational complexity. Cross-validation techniques are incorporated to enhance model robustness and reliability. The proposed system achieves faster processing speed and improved accuracy, making it suitable for real-time applications such as smart classrooms, healthcare monitoring, customer behavior analysis, and intelligent interactive systems.

Keywords
Facial Emotion Recognition, YOLOv2, SqueezeNet, Deep Learning, Real-Time Detection, MoodMap, Human-Computer Interaction

Introduction

Human emotions play an essential role in communication and decision-making processes. The ability to automatically detect emotions from facial expressions enables intelligent systems to interact more effectively with users. The MoodMap system is designed to recognize facial emotions in real time using deep learning techniques. It detects seven emotional states, including happy, sad, angry, fear, surprise, neutral, and disgust, while also estimating age and gender. The system combines YOLOv2 for accurate face detection with SqueezeNet for classification tasks. A visual dashboard is incorporated to display emotional trends, ensuring user privacy and efficient data handling.The scope of the MoodMap system extends to various application domains. It can be used in smart classrooms to monitor student engagement, in healthcare to analyze mental well-being, in retail environments for customer behavior analysis, and in human-computer interaction systems to enhance user experience. The lightweight architecture ensures fast processing and suitability for real-time deployment. Future enhancements may include mobile integration, improved classification accuracy, and expanded emotion categories.Existing facial emotion recognition systems often rely on shallow convolutional neural networks and limited datasets. These approaches face challenges in handling complex facial variations and real-world conditions. Many implementations also use outdated deep learning frameworks, reducing compatibility with modern libraries. Such systems typically offer limited scalability, reduced accuracy, and inefficient real-time performance.The proposed MoodMap system addresses these limitations by integrating advanced deep learning architectures. YOLOv2 is used for real-time face detection, while SqueezeNet performs emotion, age, and gender classification. The system extracts facial features and processes them efficiently to produce reliable predictions. This approach improves performance, reduces computational overhead, and enhances accuracy. Additionally, the system provides a user-friendly graphical interface, automated workflow for dataset preprocessing and model training, reusable trained models, and performance visualization through training graphs. These features ensure efficient real-time emotion recognition with improved usability.

Requirement Analysis

The MoodMap system consists of functional and non-functional requirements necessary for efficient operation. The functional requirements include face detection, feature extraction, emotion recognition, dataset preprocessing, model training, prediction, and performance visualization. The user module allows login authentication, dataset upload, preprocessing control, visualization of results, and logout functionality. These components collectively ensure smooth interaction between the user and the system.

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Non-functional requirements focus on performance, scalability, reliability, security, usability, compatibility, maintainability, and accessibility. These requirements ensure that the system operates efficiently under different conditions while maintaining stability and ease of use.

The computational resources required for the system include hardware and software components. The hardware requirements consist of an Intel Core i7 or equivalent processor, a minimum of 8 GB RAM, 120 GB storage space, a standard display monitor, and a high-speed internet connection. The software requirements include Windows or Linux operating systems, Python programming language, and development tools such as Tkinter, Jupyter Notebook, and Visual Studio Code. Additional libraries such as OpenCV, NumPy, Matplotlib, and TensorFlow/Keras are used for image processing, model development, and visualization.

The development process follows the Waterfall model, which is a sequential approach to software development. The first phase involves requirement analysis, where all functional and non-functional requirements are identified. The second phase focuses on system design, including architecture planning and module definition. The implementation phase involves developing the facial emotion recognition model using YOLOv2 and SqueezeNet, integrating OpenCV for real-time image capture, and building the graphical interface using Tkinter. The integration and testing phase ensures proper functionality of all components. The deployment phase involves installing the application in a local environment, and the maintenance phase focuses on updates and performance optimization.

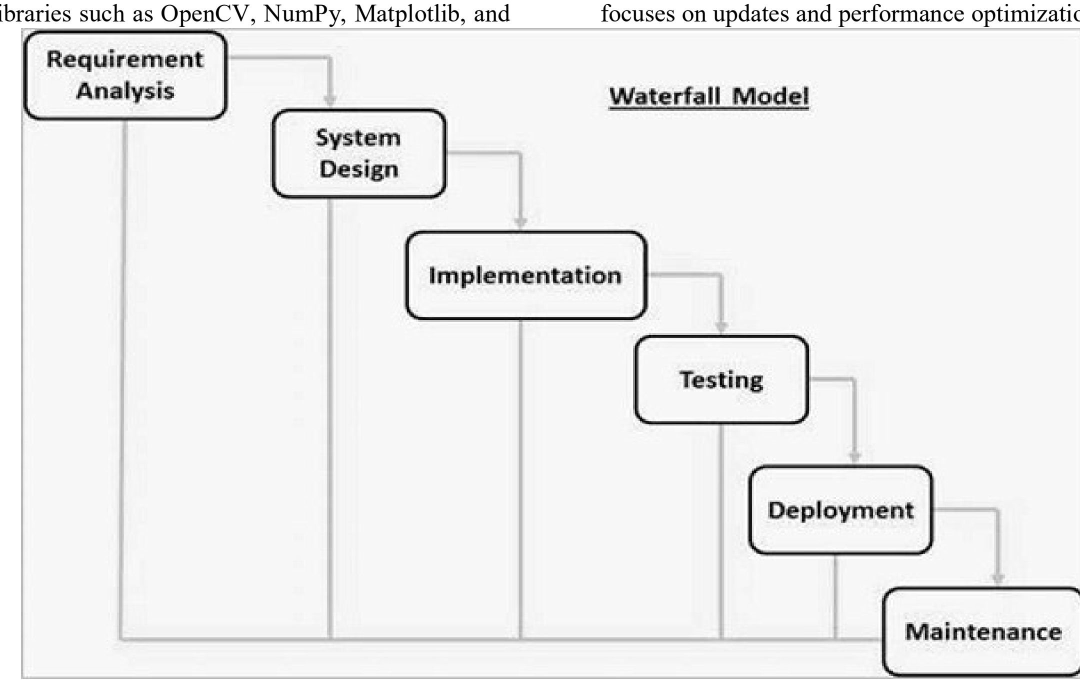

Fig 1 Waterfall Model

System Design

System design defines the structure and interaction of components required for real-time emotion recognition. In the MoodMap system, users can upload images or capture live video through a camera. The system detects faces using YOLOv2 and classifies emotion, age, and gender using a SqueezeNet-based convolutional neural network. The detected emotions are analyzed and displayed through an interactive interface that provides real-time suggestions. The system uses Tkinter for graphical interface development, OpenCV for image processing, and Matplotlib for visualization.

Software Architecture

The software architecture of the system is based on modular components to ensure scalability and maintainability. The system module handles core functionalities such as detection, classification, preprocessing, training, prediction, and visualization. The user module provides interaction capabilities for uploading datasets and viewing results. A central controller manages data flow and model execution. This modular design ensures efficient processing and easy system expansion.

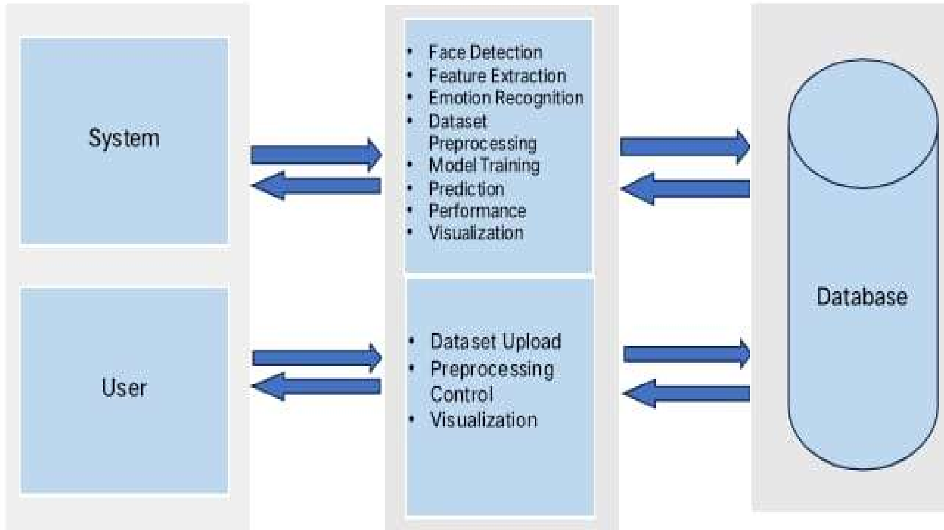

Fig 2 Software Architecture

Technical Architecture

The technical architecture consists of a user interface module, prediction engine, data processing module, and visualization module. The user interface allows users to upload images or use a live camera feed. The prediction engine performs face detection using YOLOv2 and classification using SqueezeNet. The data processing module handles dataset loading, preprocessing, and training operations. The visualization module displays prediction results and performance graphs. This architecture ensures accurate real-time emotion recognition with a seamless user experience.

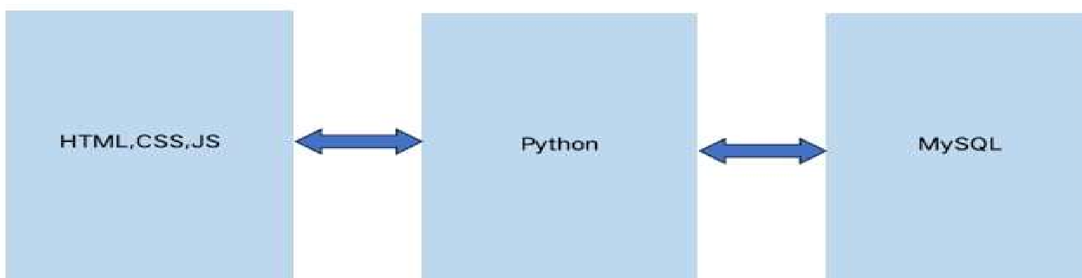

Fig 3 Technical Architecture

Implementation

Technologies

The MoodMap system is implemented using a combination of artificial intelligence, deep learning, and computer vision technologies to achieve accurate and real-time facial emotion recognition. The architecture integrates a graphical user interface, backend processing modules, and lightweight deep learning models to detect emotions, gender, and age efficiently. These technologies work together to provide reliable predictions while maintaining fast processing speed and reduced computational requirements.

Front-End Technologies

The graphical user interface of the MoodMap system is developed using Tkinter, a standard Python library for building desktop-based applications. Tkinter enables the creation of an interactive and user-friendly interface that allows users to upload datasets, initiate preprocessing, train models, and perform predictions. The interface includes components such as buttons, labels, and dialog boxes that facilitate seamless interaction between the user and the system. Additionally, Tkinter is used to display prediction outputs and performance graphs, making the application easy to operate without requiring advanced web technologies.

Back-End Technologies

Python serves as the primary programming language for implementing the MoodMap system. It provides flexibility and extensive library support for artificial intelligence and machine learning tasks. Python is used for dataset handling, preprocessing, model training, prediction, and system integration.

OpenCV is utilized for image processing and face detection. It supports real-time image capture from cameras, preprocessing operations such as resizing and normalization, and conversion of images into a suitable format for model input. OpenCV improves detection accuracy and enhances system responsiveness during real-time execution.

TensorFlow and Keras are employed for developing and training deep learning models. Keras provides a high-level interface that simplifies model creation, while TensorFlow ensures efficient computation and scalability. These frameworks are used to implement convolutional neural networks for emotion classification, enabling accurate prediction and faster training.

AI and Model Technologies

YOLOv2 is used for real-time face detection in the MoodMap system. The algorithm divides the input image into grids and predicts bounding boxes along with confidence scores. It employs anchor boxes and non-max suppression techniques to detect faces accurately and efficiently. Due to its fast processing capability, YOLOv2 is suitable for real-time emotion recognition tasks.

SqueezeNet is implemented as a lightweight deep learning architecture for emotion recognition, gender classification, and age estimation. The model extracts high-level features from facial images while maintaining a small number of parameters. This reduces memory consumption and improves performance without compromising accuracy.

Convolutional Neural Networks are used for feature extraction and classification of facial expressions. The CNN architecture processes input images through convolutional, pooling, and fully connected layers to learn meaningful patterns. The trained model classifies emotions such as happy, sad, angry, fear, surprise, neutral, and disgust. This deep learning approach enhances accuracy and robustness in real-time scenarios.

Visualization and Supporting Technologies

Matplotlib is used for generating graphical representations such as training loss and accuracy curves. These visualizations assist in analyzing model performance and identifying potential improvements during training. NumPy and Pandas are utilized for numerical computations and dataset handling. NumPy supports array operations and normalization, while Pandas organizes and preprocesses datasets. These libraries help prepare data efficiently, improving training quality and prediction accuracy.

Pseudo Code Description

The system module begins by reading an image using OpenCV and resizing it to a fixed dimension suitable for model input. The extracted images are stored as feature vectors and normalized before training. The CNN model is constructed using convolutional layers, pooling layers, and fully connected layers, followed by softmax activation for emotion classification. The model is compiled using the Adam optimizer and categorical cross-entropy loss function. After training, the system predicts emotions by reshaping input images and passing them through the trained classifier. Performance graphs are generated using Matplotlib to visualize loss and accuracy.

The user module includes authentication logic to validate login credentials and manage user sessions. It allows dataset upload using file dialog boxes and initiates preprocessing operations. The module also provides functionality to display training graphs and logout options. These modules collectively ensure smooth interaction and efficient workflow management.

Testing

Testing of the MoodMap system was conducted in multiple stages to ensure accuracy, reliability, and real-time performance. Since the system performs facial emotion recognition along with age and gender estimation, special attention was given to validating face detection, classification accuracy, and system responsiveness. The testing process also focused on usability and integration of various components.

Stages of Testing

Unit testing was performed on individual modules such as image upload, real-time camera capture, face detection, emotion classification, and suggestion generation to verify their functionality independently. Integration testing was conducted to ensure that components such as the Tkinter interface, OpenCV processing module, and CNN-based models worked together seamlessly. System testing evaluated the complete application workflow, including real-time detection, prediction, and result visualization. Acceptance testing involved end users who validated whether the system met project requirements and produced meaningful outputs.

Phases of Testing

The testing process followed structured phases including requirement validation, test planning, execution, bug fixing, and final validation. Each phase ensured that the system performed correctly under various conditions and maintained stability.

Types of Testing

Functional testing verified core features such as image upload, face detection, emotion recognition, and age and gender estimation. Usability testing ensured that the interface was simple and intuitive for users. Performance testing evaluated system speed and prediction accuracy in real-time scenarios. Compatibility testing confirmed that the application functioned properly across different operating systems and display configurations.

International Journal of
Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

Test Cases

Several test cases were executed to validate system performance. Image upload functionality was tested using valid facial images, and the system successfully uploaded and processed the input. Invalid file uploads were handled by displaying appropriate error messages. Face detection tests confirmed that faces were correctly identified in input images. Emotion recognition tests verified that predicted emotions matched expected outputs. Model training tests confirmed successful training using the dataset. Accuracy graphs were generated correctly after training. Exit functionality was also tested to ensure proper application closure. All test cases passed successfully, indicating that the system performed reliably under different conditions.

Screenshots

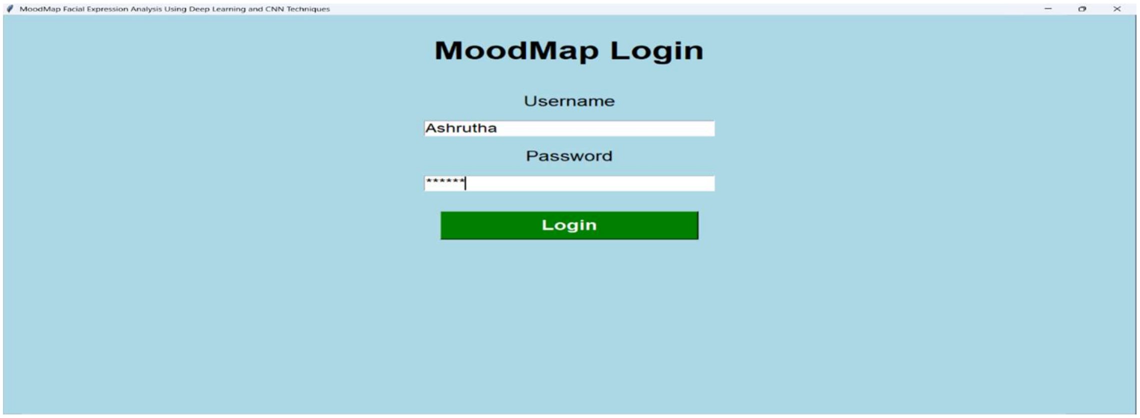

Fig 1 Login Page

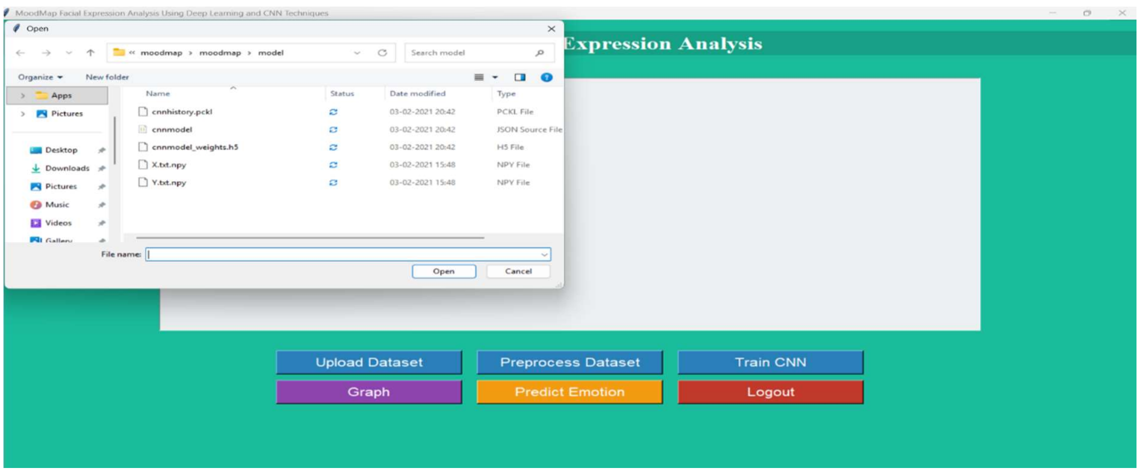

Fig 2 Upload Dataset

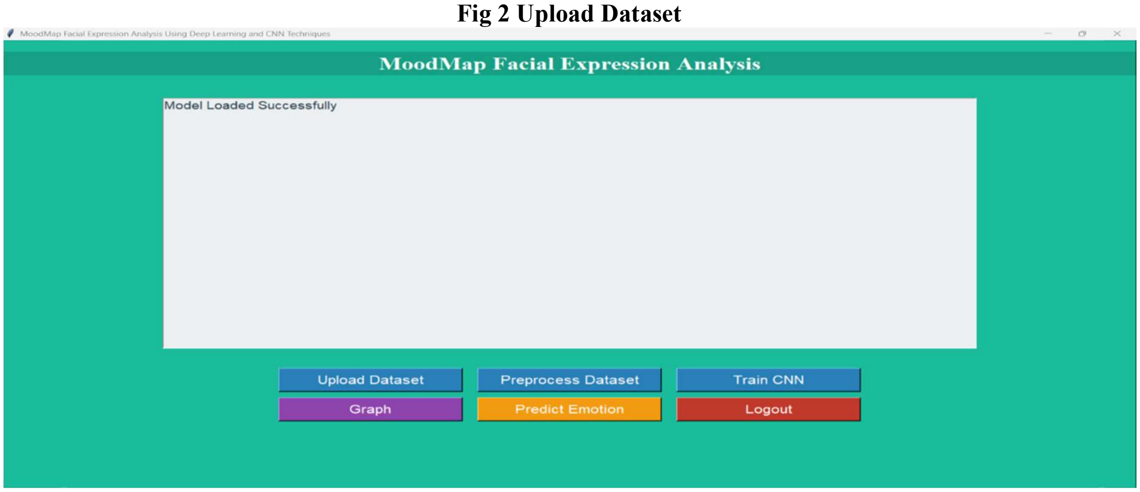

Fig 3 Train CNN

International Journal of
Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

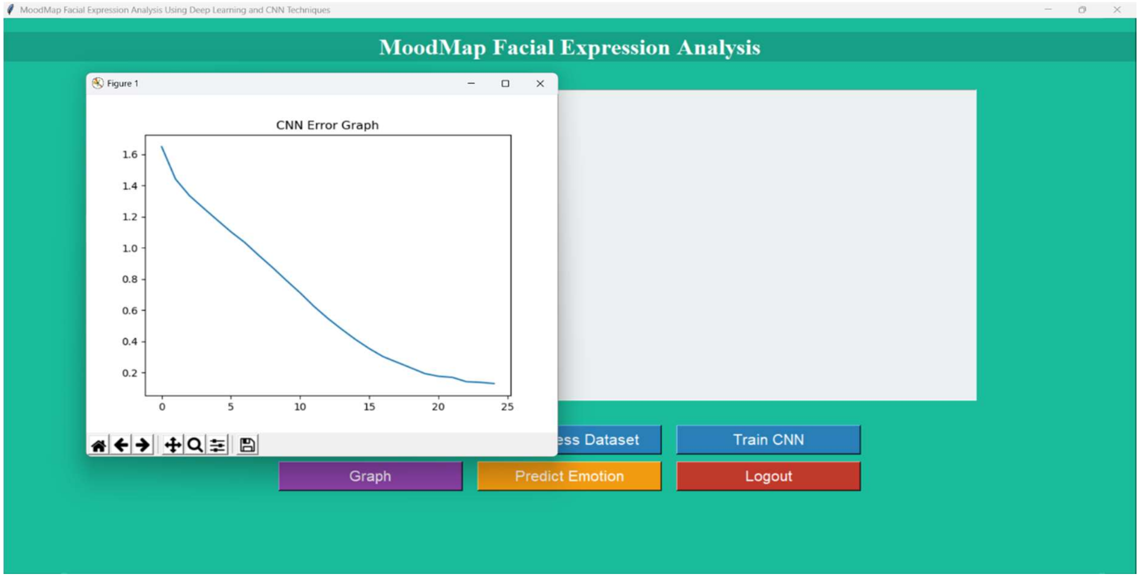

Fig 4 Graph

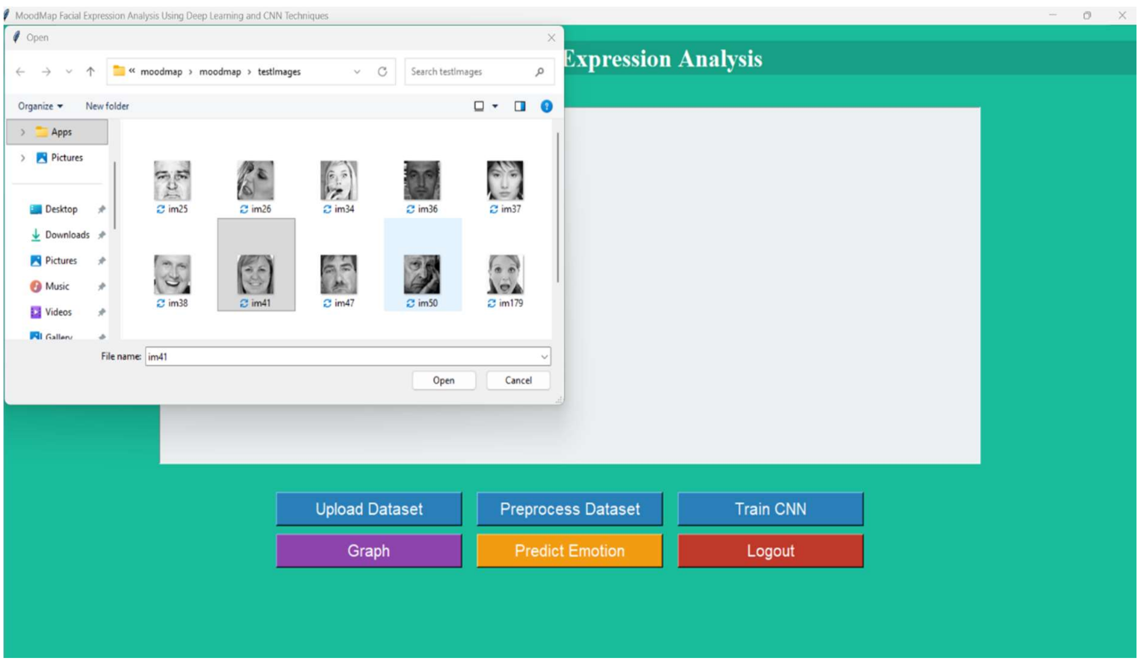

Fig 5 Emotion Prediction

Conclusion

The MoodMap system presents an efficient approach for real-time human emotion recognition using artificial intelligence and facial analysis techniques. The proposed framework integrates YOLOv2 for accurate face detection and SqueezeNet for emotion, age, and gender classification. This combination ensures fast processing, reduced computational complexity, and reliable prediction performance. The system operates effectively under varying conditions and provides clear visualization of detected emotional states. Additionally, the lightweight architecture and user-friendly interface make the application accessible for non-technical users. By enabling automatic interpretation of facial expressions, the proposed solution enhances human–computer interaction and supports intelligent decision-making in interactive systems. The system also prioritizes user privacy by minimizing unnecessary data storage, thereby improving security and trustworthiness.

Future Scope

The MoodMap system can be further improved by extending its capabilities to mobile platforms, enabling real-time emotion detection on smartphones and portable devices. Incorporating advanced deep learning architectures and multimodal analysis, including voice and physiological signals from wearable devices, can enhance prediction accuracy. Cloud-based deployment can support large-scale data processing and improve system scalability. Personalized recommendation modules may also be integrated to provide adaptive responses based on detected emotional patterns. Furthermore, the system can be expanded for applications in healthcare monitoring, smart education environments, customer behavior analysis, and intelligent smart-home systems. These enhancements will increase usability, accuracy, and applicability across diverse real-world scenarios.

References

[1] A. Author et al., "A Survey on Facial Emotion Recognition of Static and Dynamic Methods," arXiv, 2024.

[2] B. Author et al., "Facial Emotion Deep Learning Algorithms in the Detection of Human Emotions," PMC/NCBI, 2024.

[3] C. Author et al., "Detection of Human Emotions Through Facial Expressions Using Hybrid CNN-RNN Model," ScienceDirect, Elsevier, 2024.

[4] N. Dakhil and A. M. Abdulazeez, "Face Recognition Based on Deep Learning: A Review," 2024.

[5] H. L. Gururaj, B. C. Soundarya, P. Saravanan, and F. Flammini, "A Comprehensive Review of Face Recognition Techniques, Trends, and Challenges," *IEEE Access*, vol. 12, pp. 107903–107926, 2024.

[6] D. Author et al., "A Survey on Automatic Face Recognition Using Side-View Face Images," *IET Biometrics*, Wiley, 2024.

[7] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in *Proc. IEEE CVPR*, 2001.

[8] J. Redmon and A. Farhadi, "YOLOv2: Better, Faster, Stronger," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017.

[9] F. N. Iandola et al., "SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size," arXiv preprint arXiv:1602.07360, 2016.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.