As per UGC guidelines an electronic bar code is provided to secure your paper

International Journal for Modern Trends in Science and Technology
Volume 12, Issue 06, pages 178-191.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol12issue06.html
DOI: https://doi.org/10.5281/zenodo.20739521

An Intelligent Real-Time Facial Emotion Recognizing System for Human-Computer Interaction

R. Uma Aruna Devi, Bejawada Chandu, Bezawada Venkata Naga Trinadh, Chellingi Pravalika Lakshmi Sada Sri

Department of Information Technology Swarnandhra College of Engineering & Technology, Seetharamapuram, Narsapur, Andhra Pradesh, INDIA.

To Cite this Article
R. Uma Aruna Devi, Bejawada Chandu, Bezawada Venkata Naga Trinadh & Chellingi Pravalika Lakshmi Sada Sri (2026). An Intelligent Real-Time Facial Emotion Recognizing System for Human-Computer Interaction. International Journal for Modern Trends in Science and Technology, 12(06), 178-191. https://doi.org/10.5281/zenodo.20739521

Article Info
Received: 16 May 2026; Revised: 07 May 2026; Accepted: 12 June 2026.

KEYWORDS

Facial Emotion Recognition, Human–Computer Interaction, Convolutional Neural Network, Computer Vision, Real-Time Systems

ABSTRACT

Facial emotion recognition is an important component of intelligent Human–Computer Interaction (HCI) systems. Traditional computer systems lack the capability to understand human emotional states, resulting in limited interaction effectiveness. This project proposes an intelligent real-time facial emotion recognition system using computer vision and deep learning techniques. The system captures live video input, detects facial regions, and classifies facial expressions into predefined emotional categories including happy, sad, angry, surprise, fear, disgust, and neutral. Face detection is performed using OpenCV, while emotion classification is achieved through a Convolutional Neural Network (CNN) trained on a labeled facial emotion dataset. Experimental results demonstrate that the proposed system effectively recognizes emotions in real time and enhances user interaction. The system can be applied in various domains such as e-learning, healthcare, and interactive systems, with scope for further improvement in accuracy and robustness.

1. INTRODUCTION

In recent years, the field of artificial intelligence has witnessed rapid advancements, leading to the development of intelligent systems capable of performing complex tasks. One of the most important areas of research in artificial intelligence is Human–Computer Interaction (HCI), which focuses on improving the interaction between humans and machines. Traditional computer systems operate based on predefined instructions and lack the ability to understand human emotions, making interactions less effective and less natural.

Human communication is not limited to verbal language; it also includes non-verbal cues such as facial expressions, gestures, and tone of voice. Among these, facial expressions play a significant role in conveying emotions such as happiness, sadness, anger, surprise, fear, disgust, and neutrality. Understanding these expressions can greatly enhance the quality of interaction between humans and computers.

Facial Emotion Recognition (FER) is a technology that aims to identify human emotions by analyzing facial expressions using computer vision and machine learning techniques. It has gained significant attention due to its wide range of applications in areas such as healthcare, education, entertainment, and security. By integrating emotion recognition into computer systems, it is possible to develop intelligent systems that can adapt to user emotions and provide personalized responses.

Traditional methods for emotion recognition relied on manual feature extraction techniques, which were time-consuming and less accurate. These methods involved identifying facial landmarks such as eyes, nose, and mouth and analyzing their movements. However, with the emergence of deep learning, especially Convolutional Neural Networks (CNN), the process has become more efficient and accurate. CNN models can automatically learn features from images, eliminating the need for manual feature extraction.

This project focuses on developing an intelligent real-time facial emotion recognition system using deep learning techniques. The system captures live video using a webcam, detects faces using computer vision algorithms, and classifies emotions using a trained CNN model. The model is trained on the FER-2013 dataset, which contains labeled images of facial expressions.

The proposed system aims to improve Human–Computer Interaction by enabling machines to understand user emotions in real time. This can lead to more adaptive and responsive systems, improving user experience and satisfaction. The system can be used in various applications such as e-learning platforms to monitor student engagement, healthcare systems to analyze emotional states, and gaming systems to provide immersive experiences.

In conclusion, this project demonstrates the integration of computer vision and deep learning techniques to build an intelligent emotion recognition system. It highlights the importance of emotion-aware systems in modern technology and contributes to the advancement of artificial intelligence in real-world applications.

1.1 OBJECTIVES OF THE PROJECT

1.1.1 PROBLEM STATEMENT

Traditional Human–Computer Interaction systems are designed to respond to user inputs without understanding the emotional state of the user. This lack of emotional intelligence limits the effectiveness of interaction and reduces user satisfaction. For example, a system cannot detect whether a user is frustrated, bored, or engaged, which results in inappropriate responses. Moreover, manual methods of emotion analysis, such as questionnaires and observations, are time consuming, subjective, and not suitable for real-time applications. These limitations highlight the need for an automated system that can accurately detect and classify human emotions in real time.

1.1.2 IMPORTANCE OF THE PROBLEM

Emotion recognition is essential for improving the interaction between humans and machines. By understanding user emotions, systems can adapt their behavior and provide personalized responses.

This leads to enhanced user experience and better system performance.

For instance:

- In education, detecting student engagement can improve learning outcomes
- In healthcare, monitoring emotions can help in diagnosing mental health conditions
- In customer service, understanding emotions can improve satisfaction

Without emotion recognition, systems remain static and fail to meet user expectations.

1.1.3 Project Goals

The main objectives of this project are:

- To design and develop a real-time facial emotion recognition system 9121414688
- To detect human faces from live video streams
- To classify emotions using a CNN model
- To achieve high accuracy and real-time performance

1.1.4 REAL-WORLD APPLICATION

The system has several real-world applications:

1. E-Learning Systems

Helps in monitoring student engagement and attention

2. Healthcare Systems

Assists in emotional and mental health analysis

3. Gaming Industry

Adapts gameplay based on player emotions

4. Customer Feedback Analysis
Detects customer satisfaction levels
5. Smart Surveillance Systems
Identifies abnormal emotional behaviour

1.1.5 TARGET USERS

The proposed system is beneficial for:

- Students and educators
- Healthcare professionals
- Researchers in AI
- Developers building intelligent systems

1.1.6 EXAMPLE SCENARIO

Consider a student attending an online class. If the system detects that the student is bored or confused, it can notify the instructor or adjust the content difficulty. This improves learning efficiency and engagement.

1.2 OVERVIEW OF RFID TECHNOLOGY

The system requires both software and hardware components for development and deployment.

Software Requirements

- Programming Language: Python
- Libraries: OpenCV, TensorFlow, Keras, NumPy
- IDE: Jupyter Notebook / VS Code Why Python?

Python is widely used in AI due to its simplicity, readability, and extensive library support.

Hardware Requirements

- Processor: Intel i5 or higher
- RAM: Minimum 8 GB
- Webcam: Required for real-time detection

1.3 ROLE OF IOT SMART SYSTEMS

The system is divided into several modules:

1. Input Module
Captures live video from webcam.
2. Face Detection Module
Uses Haar Cascade algorithm to detect faces.
3. Preprocessing Module

- Converts image to grayscale
- Resizes image
- Normalizes data

4. Emotion Classification Module Uses CNN to classify emotions.
5. Output Module
Displays emotion in real time.

Data Flow

Input → Detection → Preprocessing → CNN → Output

System Workflow

1. Capture image
2. Detect face
3. Preprocess image
4. Predict emotion
5. Display result

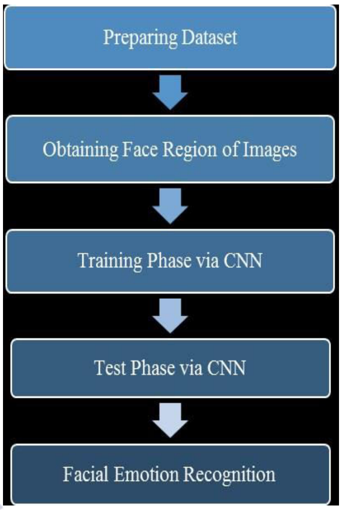

Fig 1.1: System Flowchart for Facial Emotion Recognition

Summary of Chapter 1 – Introduction

Chapter 1 introduces the concept of Human–Computer Interaction and emphasizes the need for systems that can understand human emotions. It highlights the limitations of traditional systems, which lack emotional awareness and result in poor interaction quality. To address this issue, the project proposes a real-time facial emotion recognition system using computer vision and deep learning techniques. The system utilizes a Convolutional Neural Network (CNN) to classify facial expressions into different emotions. It operates using live webcam input and processes data in real time. The chapter also discusses various applications such as e-learning, healthcare, gaming, and surveillance.

2. LITERATURE SURVEY

LITERATURE REVIEW

Facial Emotion Recognition (FER) has become a significant research area in artificial intelligence, computer vision, and human–computer interaction. It focuses on identifying human emotions from facial expressions using computational techniques. Over the years, various methods have been proposed, ranging from traditional machine learning approaches to advanced deep learning models. This chapter reviews the existing research works, techniques, and methodologies used in facial emotion recognition systems.

Earlier approaches to emotion recognition relied on handcrafted feature extraction methods such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Gabor filters. These techniques required manual selection of features from facial images, which made the process complex and less accurate. Although these methods were useful in early research, they struggled to handle variations in lighting, pose, and facial expressions.

With the advancement of machine learning, algorithms such as Support Vector Machines (SVM), k-Nearest Neighbours (k-NN), and Decision Trees were introduced for emotion classification. These models improved performance compared to traditional methods but still depended heavily on feature engineering. Their accuracy was limited when dealing with large and complex datasets.

In recent years, deep learning techniques, particularly Convolutional Neural Networks (CNN), have revolutionized the field of facial emotion recognition. CNN models automatically extract features from images, eliminating the need for manual feature extraction. This has significantly improved accuracy and efficiency in emotion classification tasks. As a result, most modern FER systems are based on deep learning approaches.

2.1 TRADITIONAL MACHINE LEARNING APPROACH

Traditional machine learning techniques played an important role in the early development of facial emotion recognition systems. These methods relied on extracting features from images and using classification algorithms to predict emotions.

Support Vector Machines (SVM)

SVM is a supervised learning algorithm widely used for classification tasks. It works by finding an optimal hyperplane that separates different classes. In emotion recognition, SVM was used to classify facial expressions based on extracted features. While SVM provided good accuracy for small datasets, it struggled with large-scale image data and complex patterns.

k-Nearest Neighbors (k-NN)

k-NN is a simple algorithm that classifies data based on the similarity between samples. It identifies the nearest neighbors and assigns the most common class. Although easy to implement, k-NN is computationally expensive and inefficient for large datasets.

Decision Trees and Random Forest

Decision Trees classify data by splitting it based on feature values. Random Forest improves this approach by combining multiple decision trees. These methods provided better performance than basic algorithms but still required manual feature extraction.

Limitations of Traditional Methods

- Dependence on handcrafted features
- Low accuracy for complex datasets
- Poor generalization
- High computational cost

These limitations led to the adoption of deep learning techniques.

2.2 DEEP LEARNING APPROACHES IN EMOTION RECOGNITION

Deep learning has significantly improved the performance of facial emotion recognition systems. Unlike traditional methods, deep learning models automatically learn features from data.

Convolutional Neural Networks (CNN)

CNN is the most widely used model for image-based tasks. It consists of convolutional layers, pooling layers, and fully connected layers. CNN extracts hierarchical features from images, making it highly effective for emotion recognition.

CNN models have shown high accuracy in classifying emotions from facial images. They can handle variations in lighting, pose, and facial expressions, making them suitable for real-world applications.

Deep Neural Networks (DNN)

DNN consists of multiple layers of neurons that learn complex patterns in data. Although effective, DNN requires large datasets and high computational power.

Recurrent Neural Networks (RNN) and LSTM

RNN and LSTM are used for sequential data analysis. In emotion recognition, they are used for video-based analysis where temporal information is important.

Advantages of Deep Learning

- Automatic feature extraction
- High accuracy
- Scalability
- Better performance on large datasets

2.3 CNN-BASED EMOTION RECOGNITION SYSTEMS

CNN-based models have become the standard approach for facial emotion recognition. Many research studies have focused on improving CNN architectures to achieve better accuracy.

CNN models are trained using datasets such as FER-2013, CK+, and JAFFE. These datasets contain labeled facial images representing different emotions. The models learn patterns from these datasets and use them to classify new images.

Research shows that CNN models outperform traditional machine learning algorithms in emotion recognition tasks. They provide higher accuracy and better generalization.

However, CNN models also have some challenges:

- Require large datasets
- High computational cost
- Training time is long

Despite these challenges, CNN remains the most effective approach.

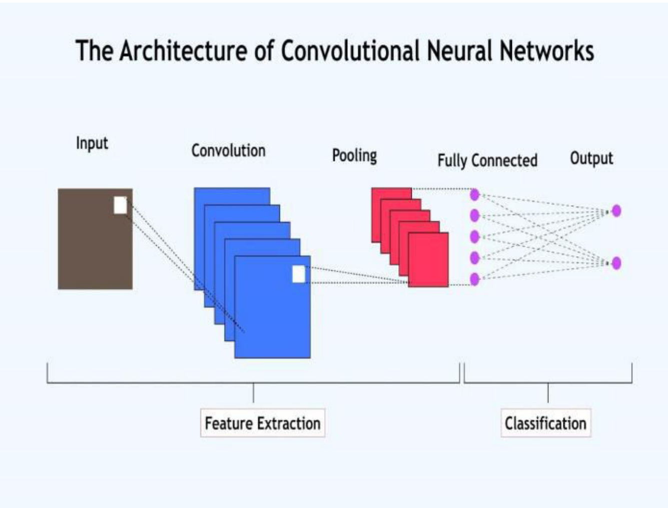

Fig 2.1 Convolutional Neural Network Architecture

2.4 COMPARISON OF MACHINE LEARNING AND DEEP LEARNING APPROACHES

Method	Accuracy	Feature Extraction	Performance
SVM	Medium	Manual	Moderate
k-NN	Low	Manual	Slow
Random Forest	Medium	Manual	Moderate
CNN	High	Automatic	Fast
LSTM	High	Automatic	Complex

2.5 EXISTING RESEARCH WORKS

Several research studies have been conducted in the field of facial emotion recognition:

- CNN-based models for emotion classification
- Hybrid models combining CNN and RNN
- Transfer learning using pre-trained models such as VGG16 and ResNet
- Real-time emotion detection systems using OpenCV

These studies demonstrate the effectiveness of deep learning in emotion recognition.

2.6 CHALLENGES IN EXISTING SYSTEMS

Despite advancements, existing systems face several challenges:

- Variations in lighting conditions
- Differences in facial expressions
- Occlusion (glasses, masks)
- Real-time processing limitations
- Dataset limitations

These challenges affect the accuracy and performance of emotion recognition systems.

2.7 RESEARCH GAP

Although many techniques have been developed, there is still a need for:

- Real-time emotion recognition systems
- Improved accuracy under varying conditions
- Lightweight models for faster processing
- Better datasets for training

2.8 CONCLUSION OF LITERATURE REVIEW

This chapter reviewed various techniques used in facial emotion recognition, including traditional machine learning and deep learning approaches. It was observed that deep learning models, especially CNN, provide better accuracy and performance compared to traditional methods. The limitations of existing systems highlight the need for improved models that can perform realtime emotion recognition efficiently. Based on this analysis, the proposed system uses a CNNbased approach to achieve accurate and real-time emotion detection.

SUMMARY OF CHAPTER 2 – LITERATURE REVIEW

Chapter 2 presents a detailed review of existing techniques and research related to facial emotion recognition. It explains how early methods relied on traditional machine learning algorithms such as SVM, k-NN, and Decision Trees, which required manual feature extraction and had limited accuracy. The chapter highlights the shift towards deep learning approaches, particularly Convolutional Neural Networks (CNN), which automatically extract features and provide better performance. It discusses various models including CNN, DNN, and RNN, and compares their effectiveness in emotion recognition tasks. The chapter also reviews existing research works and datasets used in this domain. Additionally, it identifies challenges such as lighting variations, occlusion, and real-time processing issues. A comparative analysis shows that deep learning methods outperform traditional approaches. The chapter further highlights the limitations of current systems and the need for improvement. It identifies research gaps such as the need for realtime and more accurate systems. Overall, this chapter provides a strong foundation for selecting CNN as the proposed approach.

3. PROPOSED METHODOLOGY

The proposed system is an intelligent real-time facial emotion recognition application designed to improve Human–Computer Interaction by allowing computers to understand human emotions from facial expressions. It uses a webcam to capture live video, automatically detects faces in each frame, and then classifies the detected facial region into one of several emotion categories such as happy, sad, angry, surprise, fear, disgust, and neutral. This emotion-aware capability makes interactions more natural because the computer is no longer limited to fixed, instruction-based responses; instead, it can adapt its behavior based on how the user feels. The system is intended for domains like e-learning, healthcare, gaming, and smart surveillance where understanding user emotions can significantly enhance experience and decision making.

At the heart of the system are computer vision and deep learning, especially Convolutional Neural Networks (CNNs), which are highly effective for image-based tasks. The webcam provides continuous frames, and OpenCV is used to detect faces in these frames using the Haar Cascade algorithm; only the detected face regions are then passed to the CNN model. Before classification, each face image undergoes preprocessing steps such as converting to grayscale, resizing to a fixed resolution (for example 48×48), and normalizing pixel values, which helps the model learn consistent patterns. The CNN is trained on the FER-2013 dataset, a widely used facial expression database containing labeled grayscale images, so the model learns to associate visual patterns in faces with emotion labels. Once trained and deployed, the model can predict emotions in real time and overlay the predicted label on the live video feed.

Traditional systems for understanding emotion rely on manual observation, questionnaires, or basic machine learning methods, all of which suffer from important drawbacks. Manual observation is slow, subjective, and unsuitable for continuous real-time use, while questionnaire-based methods depend on self-report and cannot capture spontaneous emotional changes during interaction. Earlier machine learning approaches such as SVM and k-NN required handcrafted features (like LBP or HOG) and struggled with variations in lighting, pose, and complex expressions, which limited accuracy and scalability. The proposed system addresses these issues by using CNNs that automatically learn features directly from facial images, removing the need for manual feature engineering and providing higher accuracy on large, complex datasets. In addition, it is explicitly designed for real-time processing of live video, which overcomes the offline and delayed nature of many existing systems.

The architecture of the proposed system is modular, ensuring clear separation of responsibilities and easier maintenance. Major modules include an Input Module (captures live video from the webcam), a Face Detection Module (locates faces using Haar Cascades), a Preprocessing Module (grayscale conversion, resizing, normalization), an Emotion Classification Module (CNN-based prediction), and an Output Module (displaying bounding boxes and emotion labels on the screen). Data flows in a pipeline: video frames enter from the camera, faces are detected, each face is preprocessed, passed into the CNN, and the predicted emotion is rendered back to the user in real time. The system is implemented mainly in Python, making use of TensorFlow and Keras for the deep learning model, OpenCV for vision tasks, and supporting libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn for data handling, visualization, and evaluation. Hardware requirements are modest—a mid-range CPU, sufficient RAM, and a webcam—so the system can run on typical desktop or laptop machines.

By combining real-time processing, automatic feature extraction, and deep learning, the proposed system achieves high accuracy and responsiveness, which are crucial for real-world Human–Computer Interaction scenarios. In e-learning, it can monitor student engagement and alert instructors if students appear bored or confused; in healthcare, it can help track emotional states related to mental health; in gaming, it can adapt game difficulty or narrative based on player emotions; and in surveillance, it can flag unusual emotional patterns indicating potential incidents. Although the system is effective, it still faces challenges such as sensitivity to lighting conditions, occlusions like masks or glasses, and slight delays on low-end hardware, which the project acknowledges as areas for improvement. Future work includes using more advanced models like ResNet or EfficientNet, improving dataset diversity, enabling multi-face emotion tracking, integrating temporal models such as LSTMs for video sequences, and deploying the system on cloud or mobile platforms for wider accessibility.

The proposed system is also designed to be flexible, extensible, and suitable as a foundation for future intelligent applications that go beyond basic emotion detection. Because it is built using modular components and standard AI libraries, developers can easily integrate additional features such as logging emotional trends over time, combining facial emotion recognition with other modalities like voice tone, or connecting the system to external platforms such as e-learning dashboards or hospital monitoring systems. This flexibility means the same core framework can be customized for different domains—for example, a classroom assistant that visualizes class-level emotional engagement, or a clinical support tool that tracks patient mood changes across sessions—making the proposed system not just a standalone application but a general platform for emotion-aware intelligent interaction.

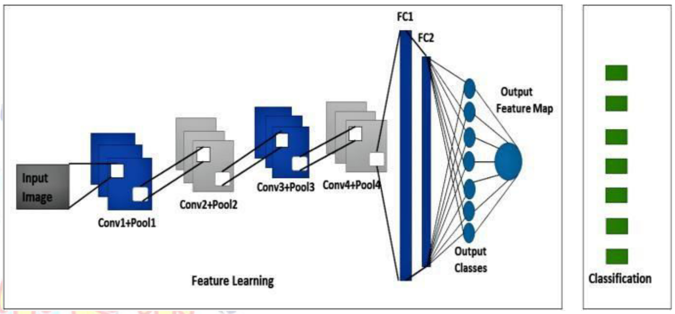

Fig 3.1: Facial Recognition System Architecture

3.1 SYSTEM ARCHITECTURE

System architecture theory describes the high-level structural framework of a system and explains how major components are organized and how data flows between them. In the Facial Emotion Recognition System, the architecture follows a layered and pipeline-based approach where the input (image, video, or live webcam) is first captured, then processed through face detection, followed by preprocessing, CNN-based classification, and finally output visualization. Each layer is logically separated so that data moves step-by-step in a controlled and organized manner. This structured workflow ensures that raw visual data is gradually transformed into meaningful emotional predictions without mixing responsibilities between modules.

The architectural design also defines how hardware and software components interact. The webcam or uploaded media acts as the input source, the processing layer uses computer vision techniques for detecting facial regions, the deep learning layer performs emotion classification, and the presentation layer displays results through a graphical interface. By clearly defining these layers, the architecture reduces system complexity and ensures that changes in one component (such as upgrading the CNN model or improving preprocessing techniques) do not affect the entire system. This loose coupling between modules improves flexibility and future adaptability.

Another important aspect of system architecture is managing data flow and performance efficiency. Since the system operates in real time, frames from the webcam must be processed quickly and sequentially. The architecture ensures optimized data handling by allowing only detected face regions to be passed to the CNN model rather than processing the entire frame. This improves speed and reduces computational load. It also supports resource management by balancing CPU and memory usage, which is essential for maintaining smooth real-time emotion detection.

Furthermore, the architectural model enhances reliability and error handling. If no face is detected, the system can handle the condition at the detection layer without interrupting other components. Similarly, invalid inputs or low-light conditions can be managed at the preprocessing stage. This layered responsibility ensures that faults are isolated and do not cause system failure. The architecture therefore strengthens system stability, robustness, and usability.

Overall, the system architecture acts as the foundation of the Facial Emotion Recognition System. It organizes components logically, ensures efficient communication between modules, supports real-time performance, and enables scalability for future enhancements such as multi-face detection, cloud deployment, or integration with advanced deep learning models.

The overall structural design of a system by organizing major components and describing how data flows between them. In the Facial Emotion Recognition System, it establishes a layered framework where input sources such as images, videos, or live webcam streams are processed through face detection, preprocessing, CNN-based emotion classification, and final result display. The purpose of this architectural approach is to ensure clear separation of responsibilities so that each module performs a specific function while remaining connected in a structured workflow. This theory supports real-time processing, scalability, maintainability, and efficient resource management, ensuring that the system operates smoothly and can be enhanced in the future without affecting its core structure.

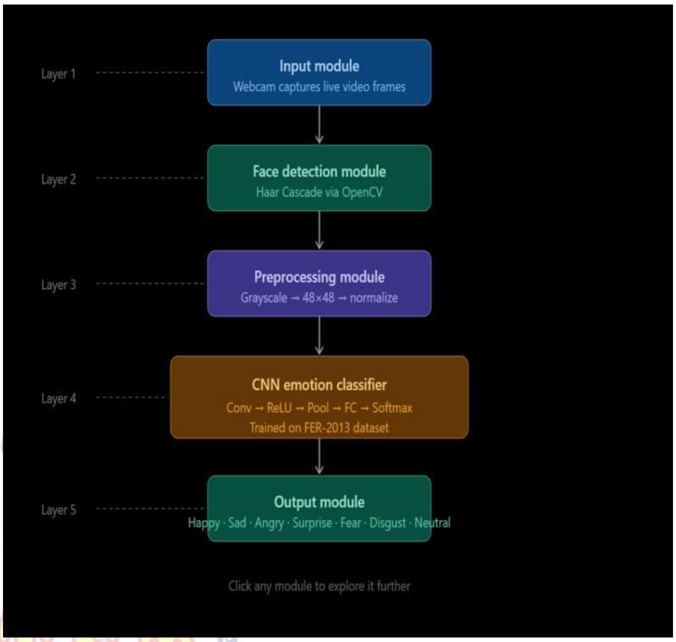

Fig 3.2 System Architecture

The system architecture represents a **layered pipeline** for a real-time Facial Emotion Recognition (FER) system. Each layer performs a specific task and passes its output to the next layer.

Layer 1 – Input Module

- Captures live video frames using a webcam.
- Uses video streaming (e.g., OpenCV VideoCapture).
- Acts as the data source for the entire system.

Purpose:

To continuously provide real-time image frames for processing.

Layer 2 – Face Detection Module

- Detects faces in each video frame.
- Uses **Haar Cascade Classifier (OpenCV)**.
- Extracts face regions (Region of Interest – ROI).

Purpose:

To isolate facial regions from the full frame before emotion analysis.

Layer 3 – Preprocessing Module

- Converts face ROI to **grayscale**.
- Resizes image to **48×48 pixels**.
- Normalizes pixel values.

Purpose:

To prepare the image in a format suitable for CNN model input and improve model accuracy.

Layer 4 – CNN Emotion Classifier

- Deep learning model trained on **FER-2013 dataset**.
- Architecture flow:
  - Convolution layer (Conv)
  - Activation (ReLU)
  - Pooling
  - Fully Connected (FC)
  - Softmax (classification)

Purpose:

To classify facial expressions into emotion categories.

Layer 5 – Output Module

- Displays predicted emotion label.
- Emotions include:
  - Happy
  - Sad
  - Angry
  - Surprise
  - Fear
  - Disgust
  - Neutral

Purpose:

To present the final predicted emotion to the user in real time.

3.2 USE CASE DIAGRAM

Use case diagram theory explains the functional behavior of a system from an external perspective. It focuses on identifying the actors involved and the services the system provides to them. In the Facial Emotion Recognition System, the main actor is the user, who interacts with the system by uploading an image, uploading a video, or activating the live webcam. Each of these interactions represents a use case that triggers internal processes such as face detection, preprocessing, emotion prediction, and output display. The use case diagram does not describe how the system works internally; instead, it concentrates on what the system does in response to user actions.

The purpose of the use case diagram is to clearly define system boundaries and functional requirements. It ensures that all required features are identified before implementation begins. In this project, the use case model guarantees that the system supports all necessary input types and produces meaningful emotional feedback. It also helps in organizing functional scenarios such as detecting multiple faces, handling no-face conditions, and analyzing real-time video streams. By outlining these scenarios, the use case diagram supports requirement validation and prevents missing or incomplete functionalities.

Additionally, the use case diagram plays an important role in system testing and documentation. Each use case can be converted into a test case to verify whether the system behaves as expected. For example, when a user uploads an image, the expected outcome is face detection and emotion display. This direct mapping between use cases and testing scenarios improves system reliability and quality assurance. Therefore, the use case diagram strengthens user-centered design, requirement clarity, and system validation.

Use case diagram theory provides a functional view of the system by identifying the interactions between external actors and the system's services. In the Facial Emotion Recognition System, it clearly defines how the user initiates actions such as uploading an image, analyzing a video, or activating live webcam detection, and how the system responds with emotion prediction results. This theoretical perspective ensures that the system's behavior is aligned with user expectations and project objectives. It helps in clarifying system scope by distinguishing what the system will do and what it will not do, thereby preventing unnecessary complexity or missing features during development.

This diagram supports requirement analysis and validation. Each interaction scenario represents a functional requirement that must be implemented and tested. For example, handling multiple faces or detecting no-face conditions becomes a defined use case scenario. This structured identification of user-system interactions improves communication between developers and stakeholders and ensures that the Human–Computer

Interaction goal of the project is fully achieved. It strengthens usability by focusing on user-driven processes rather than internal technical details.

Another important aspect of use case diagram theory is that it helps in visualizing the interaction flow between the actor and the system under different scenarios. It identifies primary use cases and possible extensions or alternative flows, such as what happens when no face is detected or when multiple faces appear in a frame. This makes the system behavior predictable and structured from a functional perspective. In the Facial Emotion Recognition System, such clarity ensures that every user action results in a well-defined system response, improving reliability and user satisfaction. By mapping these interaction flows clearly, the use case diagram strengthens functional completeness and ensures that the system consistently delivers expected outcomes.

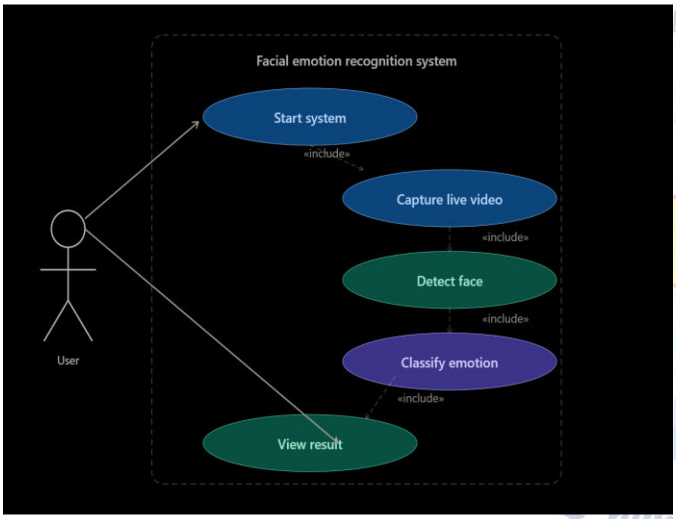

Fig 3.3 Use Case Diagram

The use case diagram describes the **interaction between the User and the Facial Emotion Recognition System**.

Actor: User

The user interacts with the system to start emotion detection and view results.

Use Cases

Start System

- User initiates the application.
- System initializes camera and model.

Capture Live Video («include»)

- System continuously captures video frames.
- Automatically included after starting the system.

Detect Face («include»)

- Detects faces in each captured frame.
- Required before classification.

Classify Emotion («include»)

- CNN model predicts the emotion from the detected face.

View Result

- Displays predicted emotion label on the screen.
- User observes the output.

Use Case Relationship

- The Start System use case includes:
  - Capture Live Video
  - Detect Face
  - Classify Emotion
- Final output is shown via View Result.

System Boundary

All processing (capture → detection → classification → display) happens inside the system boundary.

3.3 CLASS DIAGRAM

Class diagram theory represents the static structural view of a system using object-oriented design principles. It defines the system's classes, their attributes, methods, and relationships, providing a blueprint for implementation. In the Facial Emotion Recognition System, conceptual classes include input handling, face detection, preprocessing, CNN model management, and output display. Each class is responsible for performing a specific task and contains the necessary operations related to that task.

The purpose of the class diagram is to ensure proper distribution of responsibilities and maintain logical organization within the code structure. It promotes encapsulation by keeping data and related functions within the same class. For example, the face detection component handles only detection-related operations, while the CNN model class focuses solely on emotion classification. This clear separation prevents overlapping responsibilities and improves system clarity.

Another important aspect of class diagram theory is defining relationships between classes, such as associations or dependencies. In this project, the input handler passes data to the face detector, the face detector sends processed regions to the preprocessing module, and the preprocessed data is then provided to the CNN model for prediction. These relationships illustrate how objects collaborate to complete the overall task. By defining these interactions clearly, the class diagram enhances maintainability, reusability, and scalability.

Overall, the class diagram ensures that the internal design of the system is structured, modular, and adaptable. It supports efficient coding, simplifies debugging, and allows future improvements such as integrating advanced neural networks or expanding to multi-user detection without restructuring the entire system.

Class diagram theory, on the other hand, focuses on the internal structural design of the system using object-oriented principles. It defines classes, their attributes, methods, and the relationships between them to model the static structure of the system. In this project, different logical components such as input management, face detection, preprocessing, emotion classification, and output visualization can be represented as separate classes. This theoretical approach ensures modular design, where each class is responsible for a specific task and interacts systematically with other classes.

Furthermore, the class diagram enhances maintainability and reusability by promoting encapsulation and clear dependency management. By organizing the system into well-defined classes, developers can modify or upgrade individual components—such as replacing the CNN model with a more advanced architecture—without redesigning the entire system. This structured internal representation improves code clarity, simplifies debugging, and supports long-term scalability of the Facial Emotion Recognition System.

Another important contribution of class diagram theory is that it provides a clear representation of how data and operations are organized within the system. By defining attributes (such as image data, emotion labels, or model parameters) and methods (such as detectFace(), preprocessImage(), predictEmotion(), or displayResult()), the class diagram ensures that each operation is logically grouped with the data it manipulates. This structured representation improves system consistency and reduces redundancy in implementation. In the Facial Emotion Recognition System, this clarity allows developers to understand how objects interact, how responsibilities are distributed, and how the overall functionality is achieved through collaboration between classes, ultimately leading to a well-structured and efficient system design.

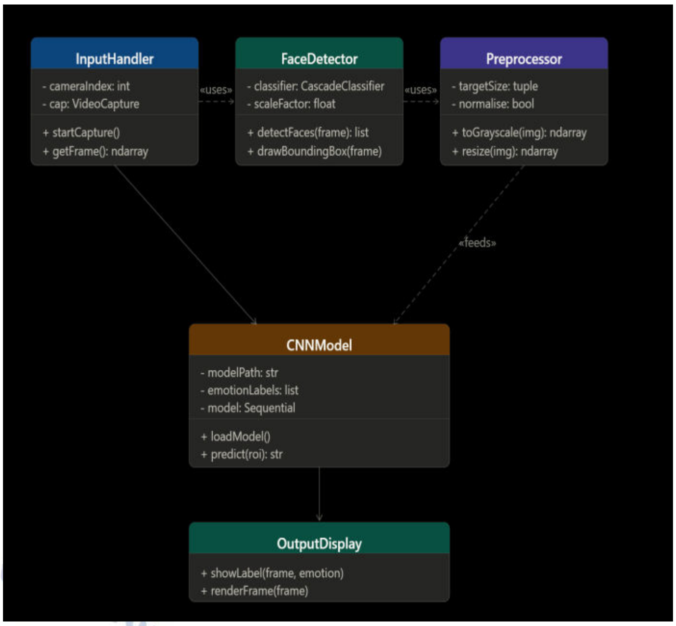

Fig 3.4 Class Diagram

The class diagram represents the **object-oriented design** of the system.

1. InputHandler Class

Attributes:

- cameraIndex: int
- cap: VideoCapture

Methods:

- startCapture()
- getFrame(): ndarray

Responsibility:

Handles webcam initialization and frame capture.

2. FaceDetector Class

Attributes:

- classifier: CascadeClassifier
- scaleFactor: float

Methods:

- detectFaces(frame): list
- drawBoundingBox(frame)

Responsibility:

Detects faces and marks them in the frame.

Relationship:
Uses InputHandler.

3. Preprocessor Class
Attributes:

- targetSize: tuple
- normalize: bool

Methods:

- toGrayscale(img): ndarray
- resize(img): ndarray

Responsibility:
Prepares face ROI before passing to CNN.

Relationship:
Feeds processed image to CNNModel.

4. CNN Model Class
Attributes:

- modelPath: str
- emotionLabels: list
- model: Sequential

Methods:

- loadModel()
- predict(roi): str

Responsibility:
Loads trained CNN model and predicts emotion.

Central Component:
Acts as the core intelligence of the system.

5. OutputDisplay Class
Methods:

- showLabel(frame, emotion)
- renderFrame(frame)

Responsibility:
Displays the emotion label and video frame.

Relationships Summary

- InputHandler → provides frames
- FaceDetector → detects faces
- Preprocessor → prepares input
- CNNModel → predicts emotion
- OutputDisplay → shows results

The design follows:

- High cohesion
- Low coupling
- Clear separation of responsibilities

RESULTS

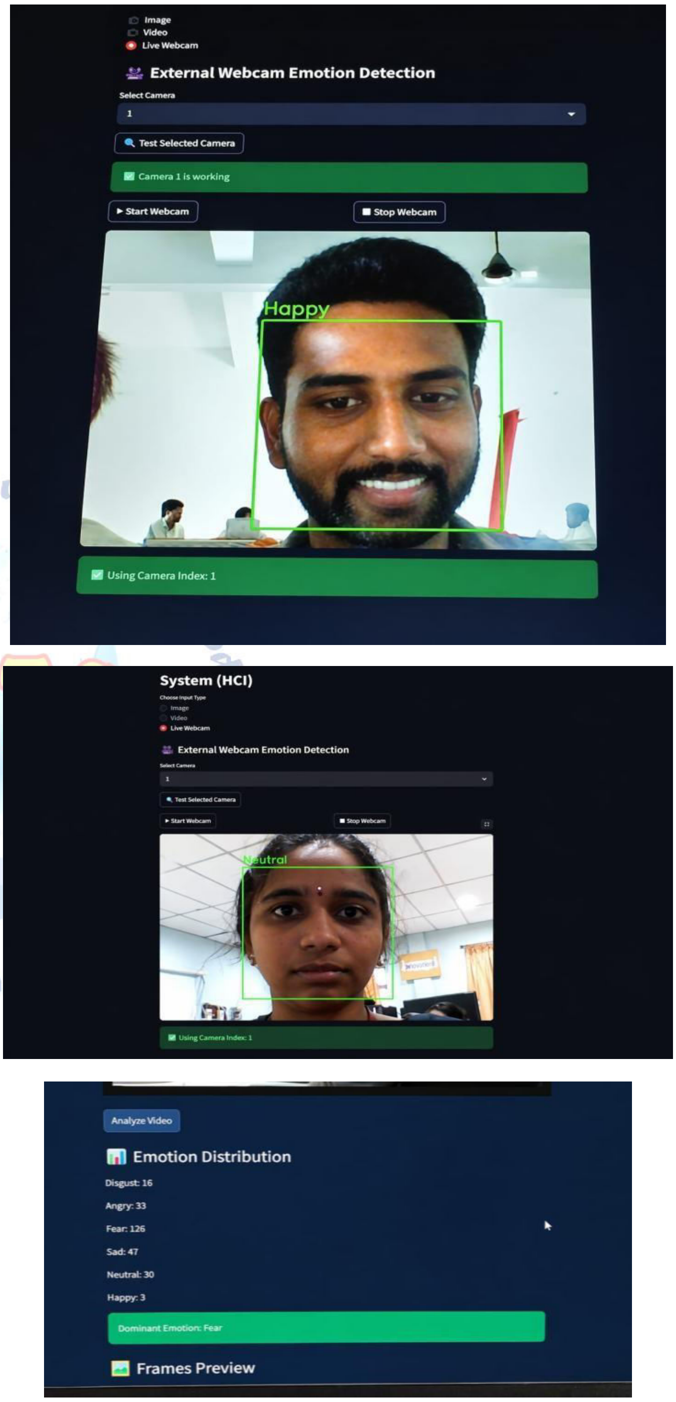

Fig 9.1: Output

CONCLUSION

The evaluation results and overall performance of the proposed Facial Emotion Recognition System demonstrate that the system is highly effective in detecting and classifying human emotions in real time.

The use of a Convolutional Neural Network (CNN) model enables accurate classification of facial expressions into multiple emotion categories such as happy, sad, angry, surprise, fear, disgust, and neutral. The system achieves high accuracy during both training and testing phases, indicating that it has successfully learned the underlying patterns in facial expressions.

The integration of computer vision techniques using OpenCV further enhances the system by enabling efficient face detection and real-time processing. The preprocessing steps, including grayscale conversion, resizing, and normalization, contribute to improved model performance and stability. The system demonstrates low prediction error and consistent results across different test scenarios, confirming its reliability and robustness.

Additionally, the real-time implementation using a webcam shows that the system can process video frames efficiently with minimal delay. The model performs well under normal conditions and provides accurate emotion predictions, making it suitable for practical applications. The combination of deep learning and computer vision ensures that the system can handle complex facial features and variations in expressions.

Overall, the proposed system successfully achieves its objective of improving Human–Computer Interaction by enabling machines to understand human emotions. The high accuracy, real-time performance, and efficient system design indicate that the model is well-trained and generalizes effectively. Therefore, this system can be deployed in real-world applications such as e-learning, healthcare, gaming, and smart surveillance to enhance user experience and decision-making.

FUTURE ENHANCEMENT

Although the proposed facial emotion recognition system demonstrates strong performance, there are several opportunities for further improvement and enhancement:

- **Integration with Real-Time Data Systems**

The system can be enhanced by integrating with real-time databases or cloud platforms to store and analyze emotion data for long-term insights.

- **Multi-Face Detection and Tracking**

The system can be extended to detect and analyze emotions of multiple users simultaneously in real-time environments.

- **Mobile and Web Application Development**

Developing a mobile or web-based application can improve accessibility, allowing users to use the system anywhere.

- **Advanced Deep Learning Models**

More advanced architectures such as ResNet, VGG, or EfficientNet can be used to improve accuracy and performance.

- **Emotion Recognition from Video Sequences**

Integrating temporal models such as LSTM can help analyze emotions over time in video streams.

- **Explainable AI (XAI) Integration**

Adding explainability features can help users understand how the model predicts emotions.

- **Cloud Deployment and Scalability**

Deploying the system on cloud platforms like AWS or Google Cloud can improve scalability and performance.

- **Improved Dataset and Feature Engineering**

Using larger and more diverse datasets with variations in lighting, age, and ethnicity can improve model general

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Ekman, P., & Friesen, W.V. (1978). Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press.

[2] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[3] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

[4] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[5] Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems.

[6] Mollahosseini, A., Hasani, B., & Mahoor, M.H. (2016). AffectNet: A database for facial expression, valence, and arousal computing in the wild. IEEE Transactions on Affective Computing.

[7] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters.

[8] Barsoum, E., Zhang, C., Ferrer, C.C., & Zhang, Z. (2016). Training deep networks for facial expression recognition with crowd-sourced label distribution. In Proceedings of ACM International Conference on Multimodal Interaction.

[9] FER-2013 Dataset. Available on Kaggle: https://www.kaggle.com/datasets/msambare/fer2013

[10] OpenCV Library Documentation. Available at: https://opencv.org/
[11] TensorFlow Documentation. Available at: https://www.tensorflow.org/
[12] Keras Documentation. Available at: https://keras.io/
[13] Sariyanidi, E., Gunes, H., & Cavallaro, A. (2015). Automatic analysis of facial affect: A survey of registration, representation, and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence.