BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

Autonomous Institute under VTU, Belagavi | Approved by AICTE,New Delhi Recognized by Govt. of Karnataka

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belgavi)

"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India)

Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF CSE(DATA SCIENCE)

A Mini Project Report On

"Human Emotion Detection from Facial Images"

A report submitted in partial fulfilment of the requirements for the

NEURAL NETWORK AND DEEP LEARNING

Project Associate:

M N NIHARIKA SHREE **3BR22CD030**

Under the Guidance of

Mr. Azhar Baig M & Ms. Chaithra B M

Dept of CSE (DATA SCIENCE),

BITM, Ballari.

Visvesvaraya Technological University

Belagavi, Karnataka

2025-2026

BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

Autonomous Institute under VTU, Belagavi | Approved by AICTE,New Delhi Recognized by Govt. of Karnataka

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belgavi)

"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballari-583 104 (Karnataka) (India)

Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF CSE (DATA SCIENCE)

CERTIFICATE

This is to certify that the Mini Project of NEURAL NETWORK AND DEEP LEARNING title "Human Emotion Detection from Facial Images" is a Bonafide work carried out by **M N NIHARIKA SHREE** bearing USN **3BR22CD030** in partial fulfillment for the award of degree of Bachelor Degree in CSE(Data Science) in the VISVESVARAYA TECHNOLOGICAL UNIVERSITY, Belagavi during the academic year 2025-2026. It is certified that all corrections and suggestions indicated for internal assessment have been incorporated in the report deposited in the library. The project has been approved as it satisfies the academic requirements in respect of mini project work prescribed for a Bachelor of Engineering Degree.

Signature of Coordinators	Signature of HOD
Mr. Azhar Baig M & Ms. Chaithra B M	**Dr. Aradhana D**

ABSTRACT

Facial emotion recognition is an essential component of human–computer interaction, yet manual interpretation of expressions can be subjective, inconsistent, and difficult to scale. With advances in deep learning, automated facial analysis has become a reliable solution for recognizing emotional states with high accuracy. This project presents an emotion detection system using a Convolutional Neural Network (CNN) model trained to classify facial images into seven categories: Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise. The FER-based facial emotion dataset is used for training, and preprocessing steps such as grayscale conversion, resizing, normalization, and augmentation help improve model generalization. The CNN architecture learns discriminative facial features through a series of convolutional, pooling, and dense layers, ultimately performing multiclass classification. Model performance is evaluated using accuracy and loss metrics across training and validation sets. The results demonstrate consistent predictions, effective feature extraction, and strong classification accuracy, highlighting the capability of CNN-based systems to support automated emotion analysis in real-world applications.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of project work on **"Human Emotion Detection from Facial Images"** would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement, and support crowned our efforts with success. It is our privilege to express our gratitude and respect to all those who inspired us in the completion of this project work.

We are extremely grateful to our Guides **Mr. Azhar Baig M & Ms. Chaithra B M** for their noble gesture, support co-ordination and valuable suggestions given in completing the project work. We also thank **Dr. Aradhana D,** H.O.D. Department of CSE(DS), for his coordination and valuable suggestions given in completing the project work. We also thank Principal, Management and non-teaching staff for their co-ordination and valuable suggestions given to us in completing the project work.

Name	USN
M N NIHARIKA SHREE	3BR22CD030

Table of Contents

CHAPTER 1

INTRODUCTION

Emotion recognition is an increasingly important area of research due to its wide applications in human–computer interaction, security, mental health assessment, and intelligent systems. Human facial expressions serve as a primary indicator of emotional state, yet accurately interpreting these expressions manually is challenging because emotions often vary subtly across individuals, cultures, lighting environments, and facial orientations. Traditional emotion analysis methods rely heavily on handcrafted features and manual observation, making them slow, subjective, and prone to inconsistencies. As digital communication and AI-driven interfaces continue to expand, there is a growing demand for automated systems that can interpret emotional cues quickly, consistently, and with minimal human intervention.

Advancements in deep learning, particularly Convolutional Neural Networks (CNNs), have revolutionized visual recognition tasks by enabling automatic feature extraction and highly accurate image classification. Leveraging this capability, this project develops an automated facial emotion detection system that classifies images into seven categories: Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise. Using an FER-based facial emotion dataset, along with preprocessing steps such as grayscale conversion, resizing, normalization, and augmentation, the system is trained to effectively recognize complex emotional patterns. The proposed CNN model delivers fast, stable, and reliable predictions, demonstrating its potential as a valuable tool for real-time emotion analysis and forming a strong foundation for future advancements in intelligent interactive systems.

1.1. Objectives

- To build an automated facial emotion detection system using CNNs.
- To classify images into seven emotion categories.
- To improve model performance using preprocessing and augmentation.
- To evaluate accuracy and other performance metrics.
- To provide reliable emotion-recognition support for intelligent systems.

1.2. Literature Survey

In paper [1], the research highlights the rapid progress of deep learning in facial expression analysis, emphasizing how CNN-based models outperform traditional manual interpretation methods. The study introduces deep architectures capable of learning subtle emotional features, significantly improving accuracy and consistency in emotion classification tasks. By leveraging large annotated facial datasets, the research demonstrates the potential of deep learning to automate and streamline emotion recognition.

In paper [2], the study examines the Xception model, which uses depthwise separable convolutions to improve feature extraction and computational efficiency. The research shows that Xception achieves superior performance in image classification tasks due to its ability to capture fine-grained facial details. This makes the model highly suitable for emotion detection, where subtle variations in expression must be identified accurately without increasing computational cost.

In paper [3], the research discusses the importance of publicly available emotion datasets such as FER-2013, which provide diverse labeled images across multiple emotion categories. The study emphasizes how such datasets support benchmarking, model comparison, and standardized evaluation. It also highlights how preprocessing and augmentation techniques help handle challenges such as image noise, class imbalance, and variability in facial appearance.

In paper [4], the study explores the use of ensemble deep learning techniques to enhance facial emotion recognition performance. By combining multiple CNN architectures, the ensemble captures richer and more diverse emotional features, leading to improved prediction accuracy. The research also demonstrates how visualization tools like activation maps help interpret influential facial regions, supporting more transparent and reliable emotion analysis.

In paper [5], the research focuses on advanced deep learning architectures such as ResNet and their role in improving facial emotion classification. Residual connections enable training deeper networks without degradation, leading to better feature learning and stability. The study highlights how such models achieve high accuracy across emotion datasets and emphasizes the growing adoption of AI-driven emotion recognition in interactive systems and real-world applications.

1.3. Problem Statement

Manual interpretation of facial expressions is subjective and error-prone due to subtle and inconsistent emotional cues. Therefore, an automated and accurate deep learning–based emotion detection system is required to ensure reliable and efficient classification.

CHAPTER 2

SYSTEM ANALYSIS

2.1. Existing System

- **Emotion recognition relies on manual observation and human judgment:** Currently, identifying emotions from facial expressions is performed manually, making the process subjective and dependent on individual interpretation.

- **High chances of misclassification due to subtle facial variations:** Many emotions share similar facial features, and small differences in expression can be difficult to distinguish accurately, leading to inconsistent results.

- **Process is slow, subjective, and varies by individual expertise:** Manual interpretation differs between observers and is not reliable for real-time applications. In scenarios requiring quick and accurate emotion understanding, human-based analysis becomes inefficient.

2.2. Proposed System

- **A CNN-based model automatically classifies facial expressions into seven emotion categories:** Unlike manual interpretation, the proposed system uses a deep-learning model that learns emotional patterns directly from the dataset, reducing dependency on human judgment.

- **Uses image preprocessing (grayscale conversion, resizing, normalization, augmentation):** Preprocessing ensures uniformity across facial images and enhances feature extraction, improving classification accuracy compared to traditional approaches.

- **Provides fast, consistent, and highly accurate emotion detection:** The system delivers predictions within milliseconds, ensuring reliable and repeatable results suitable for real-time interactive applications.

2.3. Data Collection

- **Dataset: FER-based Facial Emotion Recognition Dataset:**

A standardized and widely used dataset containing labeled images for seven emotions, providing a strong foundation for training automated emotion-detection models.

- **Contains: Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise categories:**

Similar to how radiologists compare infected and normal X-rays, the dataset includes diverse emotional expressions to help the model learn distinguishing features between them.

- **Preprocessing applied: grayscale conversion, resizing (48×48), normalization, augmentation:**

Raw facial images vary in lighting, pose, and clarity. These preprocessing steps standardize inputs and improve robustness so the model can handle real-world variations more effectively.

CHAPTER 3

SYSTEM DESIGN

3.1 Software Requirements

- **Python:** Python is the primary programming language used to develop the emotion detection system. Its simplicity and extensive machine learning ecosystem make it suitable for deep learning applications.

- **TensorFlow / Keras:** These deep learning frameworks are used to build and train the CNN model. Keras offers easy-to-use high-level APIs, while TensorFlow handles efficient backend computations.

- **OpenCV:** OpenCV is used for reading, preprocessing, and resizing facial images. It also supports prediction tasks and real-time emotion detection.

- **NumPy & Matplotlib:** NumPy supports numerical operations, while Matplotlib is used to visualize training accuracy, loss curves, and dataset samples.

- **Google Colab or Jupyter Notebook:** These notebook environments enable interactive coding, GPU usage, and easy visualization, making training and testing of the model more convenient.

3.2 Hardware Requirements

- **CPU with minimum 8 GB RAM:** A system with at least 8 GB RAM is needed to load the dataset, preprocess images, and run the training process without memory issues.

- **GPU recommended for faster training (e.g., Google Colab GPU):** Training CNN models on a CPU is slow. A GPU accelerates the computation and reduces model training time significantly.

- **Storage space for dataset (~1 GB):** Facial emotion datasets contain thousands of images; sufficient storage is required for dataset download, preprocessing, and model saving.

3.3 Functional Requirements

- **Load and preprocess facial images:** The system must read image files, convert them to grayscale, resize them to 48×48, normalize pixels, and prepare them as CNN input.

- **Train CNN model:** The system must train the CNN architecture using the emotion dataset to learn distinguishing facial features for different emotional states.

- **Perform classification:** The trained model should classify a given facial image into one of the seven emotion categories.

- **Provide accuracy results and graphs:** After training, the system must display performance metrics such as training/validation accuracy and loss through plotted graphs.

- **Save the trained model:** The system should store the final trained model so it can be reused later without retraining.

3.4 Non-Functional Requirements

- **Accuracy and reliability:** The system should consistently produce accurate emotion classifications, ensuring dependable results for real-world applications.

- **Scalability for large datasets:** The system should be able to handle additional emotion images or expanded datasets in the future.

- **Real-time prediction capability:** The model should support fast inference so it can be integrated into real-time emotion recognition systems.

- **Usability for end users:** The system should follow a simple workflow that can be easily used in applications such as interactive systems, monitoring tools, or educational platforms.

CHAPTER 4

IMPLEMENTATION

4.1 Architectural Design

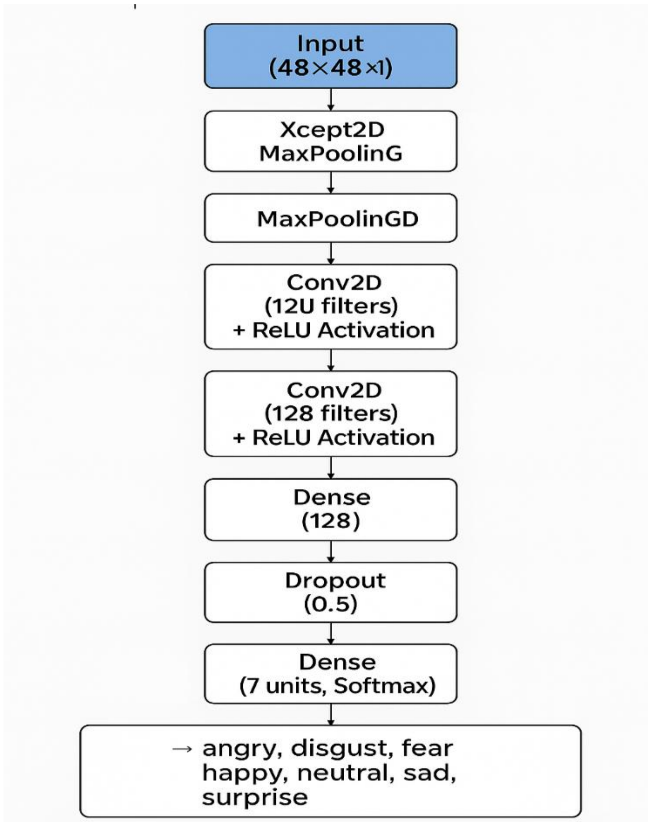

Fig 4.1 - Architectural Design

The architectural design of the proposed emotion detection system is based on a Convolutional Neural Network (CNN) trained to classify facial expressions into seven emotion categories. The process begins with input face images resized to 48×48 pixels and converted to grayscale, ensuring uniformity and reduced computational load. These images pass through multiple convolutional and pooling layers, where the network extracts low- to high-level spatial features such as edges, textures, and facial structure patterns. The resulting feature maps are flattened and processed by a dense layer with 128 neurons to learn deeper representations useful for emotion classification. A dropout layer (0.5) is included to prevent overfitting and enhance model generalization. Finally, a softmax-activated output layer with seven units produces probability scores for each emotion class: Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise. This architecture provides an efficient, scalable, and accurate framework for facial emotion recognition.

4.2 Description of Modules

4.2.1 Data Acquisition and Preprocessing Module

This module manages the loading and preparation of the facial emotion dataset. The images are organized into training and testing sets, followed by preprocessing steps such as converting images to grayscale, resizing them to 48×48 pixels, and normalizing pixel values. Using tools like ImageDataGenerator, the module also performs augmentation to increase data variety. This ensures that all input images are standardized and suitable for training the CNN model.

4.2.2 Model Construction Module

In this module, a Convolutional Neural Network (CNN) is built for multi-class emotion classification. The architecture consists of multiple convolutional and max-pooling layers for feature extraction, followed by Flatten and Dense layers for high-level learning. A dropout layer is included to reduce overfitting, and the final Softmax output layer predicts one of seven emotion categories. The model is compiled using the Adam optimizer and categorical cross-entropy loss, preparing it for effective training.

4.2.3 Model Training and Validation Module

This module handles the training process of the CNN using the pre-processed facial images. The model is trained for a specific number of epochs, and both training and validation accuracy and loss are monitored throughout. The module ensures that the CNN learns expressive patterns corresponding to different emotions while tracking model behaviour to detect issues such as overfitting or insufficient learning.

4.2.4 Model Evaluation and Visualization Module

After training, this module evaluates the model using the test dataset to assess generalization performance. It calculates key metrics such as test accuracy and generates visual plots of training and validation accuracy and loss. These visualizations help interpret how well the model performs and whether the learning process was stable and effective.

4.2.5 Model Saving and Deployment Module

This module saves the trained emotion detection model in .h5 format for future use or deployment. The saved model can later be integrated into applications for real-time emotion recognition, including webcam-based detection or embedded systems. This ensures easy reuse, scalability, and practical implementation of the system.

4.3 Code Implementation

Algorithm: Emotion Detection using CNN Model

Input: FER-based Facial Emotion Dataset (Kaggle)
Output: Predicted class (Angry / Disgust / Fear / Happy / Neutral / Sad / Surprise) and performance metrics
1. Start

2. Load and Download Dataset

2.1 Install and import required libraries (TensorFlow, Keras, Numpy, Matplotlib, OpenCV-Python, OS).
2.2 Download or organize the FER dataset into project folders (train / test).
2.3 Define dataset directories:
- data/train
- data/test

3. Preprocess Images

3.1 Set image size to **(48 × 48)** and batch_size = 32.
3.2 Initialize ImageDataGenerator for training with rescale = 1./255 and augmentation (rotation, zoom, horizontal_flip).
3.3 Initialize ImageDataGenerator for validation/test with rescale = 1./255.
3.4 Create generators using flow_from_directory() for:

• Training images (color_mode = "grayscale", target_size = (48,48))
• Test/Validation images (same settings)
3.5 Generators automatically resize, normalize, augment, and batch the images.

4. Build CNN Model

4.1 Initialize a Sequential CNN model.
4.2 Add convolutional & pooling layers, for example:
• Conv2D(32,(3,3)) + ReLU → MaxPooling2D
• Conv2D(64,(3,3)) + ReLU → MaxPooling2D
• Conv2D(128,(3,3)) + ReLU → MaxPooling2D
4.3 Add Flatten() layer.
4.4 Add Dense(128, activation='relu').
4.5 Add Dropout(0.5) to reduce overfitting.
4.6 Add final Dense(7, activation='softmax') for multiclass output.

5. Compile Model

5.1 Set optimizer: Adam(learning_rate=0.0001).
5.2 Set loss function: categorical_crossentropy.
5.3 Set evaluation metric = Accuracy.

6. Train Model

6.1 Train the model using training and validation generators with:
• Epochs = 25
• Batch size = 32
6.2 Store training history containing accuracy and loss values.

7. Test Model

7.1 Evaluate the trained model using the test generator.
7.2 Obtain test accuracy and test loss values.

8. Evaluate Performance

8.1 Display test accuracy returned by model.evaluate().
8.2 Compare training vs validation accuracy to check model consistency.
8.3 Check for overfitting or underfitting through stored accuracy/loss curves.

9. Visualize Results

9.1 Plot training and validation accuracy across epochs.
9.2 Plot accuracy graph using Matplotlib.
9.3 Display visual trends for model performance.

10. Save Model

10.1 Save trained CNN model as emotion_model.h5 for future inference.

11. Deployment / Prediction

11.1 Load emotion_model.h5 and labels.npy.
11.2 For a new image: detect/crop face → convert to grayscale → resize (48×48) → scale 1./255 → model.predict() → map argmax to emotion label.

12. End

CHAPTER 5

RESULT

The performance of the proposed CNN-based emotion detection model was evaluated using training and validation datasets. The accuracy curve, shown in Figure 1, illustrates the progressive improvement of the model across 25 epochs. The graph demonstrates how the network gradually learns discriminative facial features and stabilizes its performance during training.

```
Found 28709 images belonging to 7 classes.
Found 7178 images belonging to 7 classes.
Emotion labels: {'angry': 0, 'disgust': 1, 'fear': 2, 'happy': 3, 'neutral': 4, 'sad': 5, 'surprise': 6}
2025-12-11 14:35:15.272233: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2025-12-11 14:35:15.273190: W tensorflow/stream_executor/cuda/cuda_driver.cc:263] failed call to cuInit: UNKNOWN ERROR (303)
2025-12-11 14:35:15.312169: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: LAPTOP-D69MFD6J
2025-12-11 14:35:15.312827: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: LAPTOP-D69MFD6J
2025-12-11 14:35:15.326204: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations
:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 46, 46, 32)        320

 max_pooling2d (MaxPooling2D  (None, 23, 23, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 21, 21, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 10, 10, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 8, 8, 128)         73856

 max_pooling2d_2 (MaxPooling  (None, 4, 4, 128)        0
 2D)

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 128)               262272

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 7)                 903

=================================================================
Total params: 355,847
Trainable params: 355,847
Non-trainable params: 0
```

Fig 5.1 – CNN Model Architecture Summary

```
=================================================================
Total params: 355,847
Trainable params: 355,847
Non-trainable params: 0
_________________________________________________________________
Epoch 1/25
898/898 [==============================] - 146s 162ms/step - loss: 1.7428 - accuracy: 0.2958 - val_loss: 1.5666 - val_accuracy: 0.4106
Epoch 2/25
898/898 [==============================] - 109s 121ms/step - loss: 1.5135 - accuracy: 0.4145 - val_loss: 1.3969 - val_accuracy: 0.4631
Epoch 3/25
898/898 [==============================] - 88s 98ms/step - loss: 1.4032 - accuracy: 0.4651 - val_loss: 1.3157 - val_accuracy: 0.4978
Epoch 4/25
898/898 [==============================] - 73s 81ms/step - loss: 1.3344 - accuracy: 0.4917 - val_loss: 1.2946 - val_accuracy: 0.5001
Epoch 5/25
898/898 [==============================] - 66s 73ms/step - loss: 1.2791 - accuracy: 0.5144 - val_loss: 1.2464 - val_accuracy: 0.5254
Epoch 6/25
898/898 [==============================] - 58s 64ms/step - loss: 1.2381 - accuracy: 0.5293 - val_loss: 1.2232 - val_accuracy: 0.5302
Epoch 7/25
898/898 [==============================] - 68s 76ms/step - loss: 1.2047 - accuracy: 0.5419 - val_loss: 1.1983 - val_accuracy: 0.5433
Epoch 8/25
898/898 [==============================] - 73s 82ms/step - loss: 1.1754 - accuracy: 0.5558 - val_loss: 1.1967 - val_accuracy: 0.5474
Epoch 9/25
898/898 [==============================] - 73s 82ms/step - loss: 1.1476 - accuracy: 0.5643 - val_loss: 1.1876 - val_accuracy: 0.5479
Epoch 10/25
898/898 [==============================] - 75s 83ms/step - loss: 1.1166 - accuracy: 0.5739 - val_loss: 1.1883 - val_accuracy: 0.5508
Epoch 11/25
898/898 [==============================] - 66s 74ms/step - loss: 1.0876 - accuracy: 0.5836 - val_loss: 1.1792 - val_accuracy: 0.5522
Epoch 12/25
898/898 [==============================] - 60s 67ms/step - loss: 1.0562 - accuracy: 0.5977 - val_loss: 1.1962 - val_accuracy: 0.5559
Epoch 13/25
898/898 [==============================] - 59s 66ms/step - loss: 1.0354 - accuracy: 0.6003 - val_loss: 1.1819 - val_accuracy: 0.5616
Epoch 14/25
898/898 [==============================] - 60s 66ms/step - loss: 1.0188 - accuracy: 0.6108 - val_loss: 1.2173 - val_accuracy: 0.5483
Epoch 15/25
898/898 [==============================] - 59s 65ms/step - loss: 0.9935 - accuracy: 0.6189 - val_loss: 1.1837 - val_accuracy: 0.5619
Epoch 16/25
898/898 [==============================] - 60s 67ms/step - loss: 0.9671 - accuracy: 0.6267 - val_loss: 1.2164 - val_accuracy: 0.5571
Epoch 17/25
898/898 [==============================] - 59s 66ms/step - loss: 0.9508 - accuracy: 0.6358 - val_loss: 1.2353 - val_accuracy: 0.5665
Epoch 18/25
898/898 [==============================] - 60s 66ms/step - loss: 0.9281 - accuracy: 0.6402 - val_loss: 1.2374 - val_accuracy: 0.5614
Epoch 19/25
898/898 [==============================] - 59s 65ms/step - loss: 0.9134 - accuracy: 0.6463 - val_loss: 1.3100 - val_accuracy: 0.5552
Epoch 20/25
898/898 [==============================] - 62s 69ms/step - loss: 0.8977 - accuracy: 0.6501 - val_loss: 1.2645 - val_accuracy: 0.5620
Epoch 21/25
898/898 [==============================] - 59s 65ms/step - loss: 0.8745 - accuracy: 0.6594 - val_loss: 1.2804 - val_accuracy: 0.5574
Epoch 22/25
898/898 [==============================] - 59s 65ms/step - loss: 0.8593 - accuracy: 0.6654 - val_loss: 1.3281 - val_accuracy: 0.5580
Epoch 23/25
898/898 [==============================] - 74s 83ms/step - loss: 0.8394 - accuracy: 0.6692 - val_loss: 1.2809 - val_accuracy: 0.5646
Epoch 24/25
898/898 [==============================] - 74s 82ms/step - loss: 0.8281 - accuracy: 0.6769 - val_loss: 1.3269 - val_accuracy: 0.5574
Epoch 25/25
898/898 [==============================] - 58s 64ms/step - loss: 0.8104 - accuracy: 0.6822 - val_loss: 1.3595 - val_accuracy: 0.5642
```

Fig 5.2 – Epoch-wise Training and Validation Performance of the CNN Model

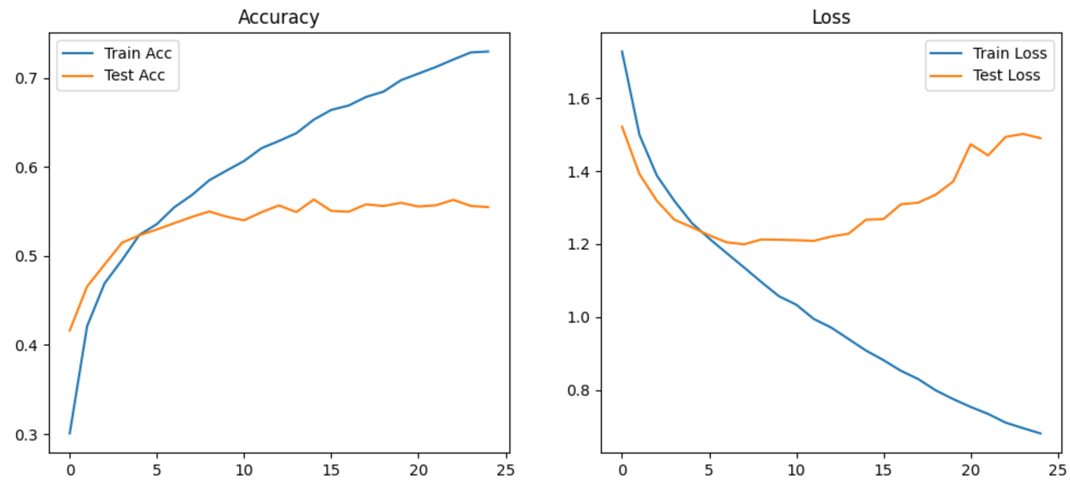

Fig 5.3 – Training and Validation Accuracy and Loss Curves of the CNN Model.

CHAPTER 6

CONCLUSION

The project successfully demonstrates the capability of Convolutional Neural Networks to recognize human emotions from facial images with reliable accuracy. Through preprocessing steps such as grayscale conversion, resizing, normalization, and data augmentation, the model effectively learned distinct facial features associated with different emotional states. The achieved performance validates CNNs as a fast, consistent, and objective tool for automated emotion recognition. While the model performs well, further improvements are possible by experimenting with deeper architectures, expanding the dataset, and integrating real-time detection and explainable AI techniques to enhance interpretability. Overall, this work establishes a strong baseline for developing advanced emotion recognition systems applicable in human–computer interaction, mental health monitoring, surveillance, and intelligent user interfaces.

CHAPTER 7

REFERENCES

1. **Goodfellow, I., Bengio, Y., & Courville, A.** (2016). *Deep Learning*. MIT Press.

2. **Chollet, F.** (2017). Xception: Deep Learning with Depthwise Separable Convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

3. **Simonyan, K., & Zisserman, A.** (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556*.

4. **He, K., Zhang, X., Ren, S., & Sun, J.** (2016). Deep Residual Learning for Image Recognition. *CVPR*.

5. **Lundqvist, D., Flykt, A., & Öhman, A.** (1998). *The Karolinska Directed Emotional Faces (KDEF) Dataset*. Department of Clinical Neuroscience, Psychology Section, Karolinska Institutet.

6. **Goodfellow, I., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., & Bengio, Y.** (2013). Challenges in Representation Learning: A Report on Three Machine Learning Contests. *Neural Information Processing*.

7. **Zhou, B., Khosla, A., Lapedriza, A., Torralba, A., & Oliva, A.** (2016). Learning Deep Features for Discriminative Localization. *CVPR*.

8. **Krizhevsky, A., Sutskever, I., & Hinton, G. E.** (2012). ImageNet Classification with Deep Convolutional Neural Networks. *NIPS*.

9. **FER-2013 Facial Expression Dataset.** (2013). Kaggle. https://www.kaggle.com/datasets/msambare/fer2013