Department of Electrical and Computer Engineering
North South University

CSE445 Project

Facial Recognition with Expression Detection System

Tofayel Ahmed	**2112096042**
Tanzeem Haque	**2131122642**
S M Samiul Haq	**2011103042**
Entiser Hossain	**2021789642**

Faculty Advisor:

Mohammad Shifat-E-Rabbi

Assistant Professor & Undergraduate Coordinator(CSE)

ECE Department

Spring, 2025

Abstract

Facial expression recognition (FER) helps computers understand how people feel by analyzing their facial expressions. This is useful in many areas like online learning, customer service, and healthcare. In this project, we built a system using machine learning to identify four emotions: Angry, Happy, Neutral, and Surprised. We used a deep learning model called ResNet50 and trained it with images collected from local volunteers. Our goal was to create an effective yet resource-friendly solution. The report explains how the system was built, the challenges we faced, and our future plans.

Index Terms Facial Recognition, Expression Detection, Machine Learning, ResNet50.

Introduction

1.1 Background and Motivation

Facial expressions are a natural, non-verbal way humans communicate emotions like happiness, anger, surprise, and sadness. We use them daily to understand how others feel. Similarly, machines can be trained to recognize these expressions, enabling more human-like interactions. This has many real-world applications—in healthcare, education, security, and human-computer interaction.

Automatic facial expression recognition is gaining popularity. It helps online learning platforms track student engagement, assists hospitals in identifying patient discomfort, and allows robots to interact emotionally with users. These benefits show the growing importance of emotion-aware systems.

Deep learning, especially Convolutional Neural Networks (CNNs), has shown great results in facial expression recognition. ResNet50 is a powerful CNN that performs well with fewer training parameters. Inspired by this, we decided to build a facial expression recognition system using ResNet50. Our system focuses on recognizing four basic emotions: Angry, Happy, Neutral, and Surprised.

Project Objectives:

- Local Dataset Creation using the Facial Images of Volunteers
- Apply standard image pre processing.

- To classify the facial expressions use the ResNet50 model and train and test the model.

What Makes Our Project Unique:

- Uses a locally collected dataset from Bangladeshi volunteers.
- Designed to run on basic hardware, suitable for learning environments.
- Made simple for easy understanding and further development by other students.

Literature Review

2.1 Existing Research and Limitations

Facial Expression Recognition (FER) has become popular in recent years, thanks to deep learning, especially Convolutional Neural Networks (CNNs).

- Zhang et al. [1] used ResNet50 for emotion recognition on large datasets like FER2013 and RAF-DB, achieving strong results. They showed that deep models perform well when trained properly.
- Mollahosseini et al. [2] used a different CNN trained on AffectNet, a large dataset with millions of face
- images from the internet. Their model recognized many emotions more effectively than older methods.
- Goodfellow et al. [3] stressed the importance of using diverse data for training to improve accuracy across different faces, angles, and lighting.

Although many studies on facial expression recognition show good results, they have some limitations:

- Foreign Datasets: Most datasets are from Western countries and don't match local facial features.
- High Hardware Needs: Many models require powerful GPUs not available in typical classrooms or small setups.
- Too Many Emotion Classes: Some studies use 6–8 emotions, which are hard to separate with small datasets.

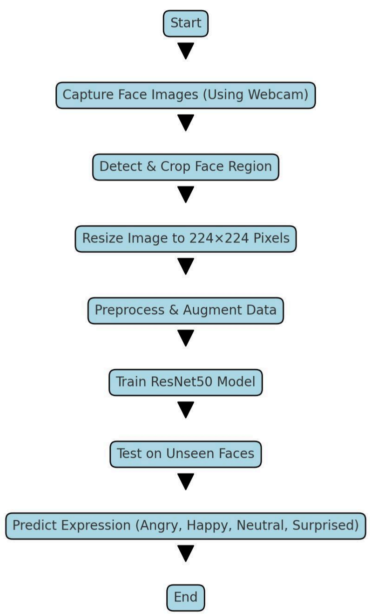
To overcome these issues, we took a simple and local approach:

- We used a local dataset from Bangladeshi volunteers, making it more culturally relevant.
- We focused on four clear emotions: Angry, Happy, Neutral, and Surprised.
- We chose ResNet50, which is accurate and runs on normal laptops.

Methodology

3.1 System Design

Our facial expression recognition system was designed in a simple step-by-step process. First, we collected face images from people. Then, we prepared the images, trained a deep learning model, and finally tested it to see how well it could recognize emotions.



3.2 Hardware and Software Components

This is a software-based project. We did not use any physical hardware like sensors or microcontrollers. Everything was done on a regular laptop using Python and some useful AI libraries.

3.3 Software Implementation

Step 1: Image Collection

We created our own dataset by collecting images from 12 volunteers. Each person showed four expressions: Angry, Happy, Neutral, and Surprised.

To collect the images, we used a webcam and a Python program. The program:

- Detects faces in real time
- Crops the face area
- Resizes the face to 224x224 pixels
- Saves the image when we press ‘s’
- Stops saving when we press ‘f’
- Pressing ESC exits the program

This helped us collect many good quality images for training.

Step 2: Preprocessing

After collecting the images:

- We resized all to 224×224 pixels
- We applied data augmentation such as:
- Horizontal flip
- Small rotation
- Zoom in/out
- Brightness and contrast changes

These steps helped the model learn better and prevented overfitting.

Step 3: Model Training

We used a deep learning model called ResNet50, which is good at understanding image features. Instead of building from scratch, we used a pre-trained model and added our own final layer to classify the four emotions.

- Loss function: Sparse Categorical Cross Entropy
- Optimizer: Adam
- Training epochs: 20
- Validation split: 20% from training data

The model learned to detect emotions from thousands of example images.

Step 4: Model Testing

We tested the model using images from 2 people not included in training. This helped us check if the model can recognize expression from new faces.

Step 5: Results Visualization

We used graphs to show:

- Training accuracy and loss
- Validation performance
- Confusion matrix to see which emotions were easy or hard to classify

TABLE I. A SAMPLE SOFTWARE/HARDWARE TOOLS TABLE

Tool	Function	Other Options	Why We Chose It
Python	Programming language	C++, Java	Easy to learn and popular in AI projects
OpenCV	Face detection and image processing	PIL, Dlib	Works well with cameras and image editing
TensorFlow & Keras	Model training (ResNet50)	PyTorch	Simple syntax and good documentation
Google Colab	Cloud-based training (optional)	Jupyter Notebook	Free GPU support for faster model training
Matplotlib, Seaborn	Creating graphs and visualizations	Plotly	Helps to show accuracy and test results clearly
CVZone	Easy face detection module built on OpenCV	-	Quick and simple for detecting faces from webcam

Result and Discussion

4.1 Experiment Setup and Variables

The base goal of this project was to come up with a facial expression recognition model that could classify accurately between 4 of the leading emotions such as: angry, happy, neutral, and surprised. After going through multiple research papers, we filtered out two model architectures: ResNet-50 and VGG-face. However, due to hardware constraints, and due to VGG-face being resource heavy, we decided to stick to ResNet-50. In the short time we had for the project, we were able to go through three training iterations.

4.2 Data Collection

The dataset comprised of images of 10 individuals, where each individual produced 100 images per class. There were 100 images per individual class (each emotion) for 4,000 images in total for the training dataset, which was split into 0.8 for train and 0.2 for validation. Initially there were five classes, however, the 'sad' class was dropped considering its similarity with the expression differentiation class of 'neutral' during data collection.

4.3 Model Training and Validation

- First Iteration: At first, the model was initially trained with all layers frozen, except for the last layers and a low learning rate. The aim was to take advantage of pre-trained features but to modify the model to the new facial expression data. It reached the plateau at 68% accuracy, but the confusion matrix showed not good generalization among classes..

- Second Iteration: They unfrozen the last 50 layers and further reduced the learning rate. The purpose was to more finely adjust more of the network's parameters to make them better approximate the facial expression data. Finally, we constructed a model that had a better accuracy, equal to 76%, as can be seen in the confusion matrix (Figure 2) that

has a better distribution between the predicted classes.

- Third Iteration: Due to the poor results from the previous iterations, all layers were unfrozen and the learning rate was reduced in order to achieve a deeper and finer layer adaptation. To make the learning thorough enough, we increased the number of epochs to 20. However, during this iteration, the accuracy dropped to 51% (Figure 3), which may indicate an overfitting problem or in spite of a lack of training stability.

4.4 Results Presentation

The results from each iteration were methodically recorded and are presented in Figures 1-3:

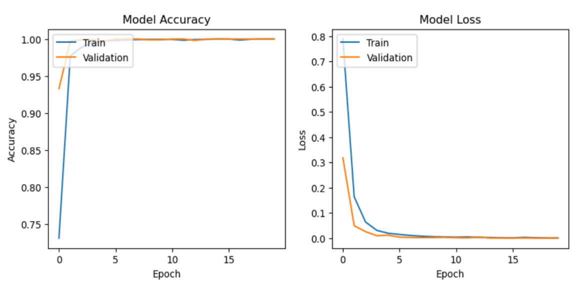


Figure 1: Model accuracy and loss of the model of the second iteration

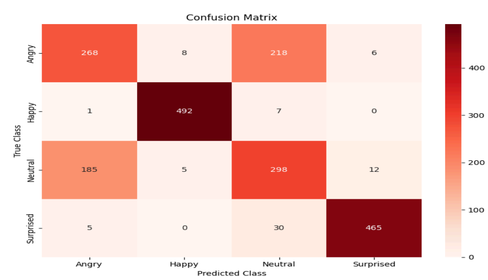


Figure 2: Confusion matrix for the second iteratio

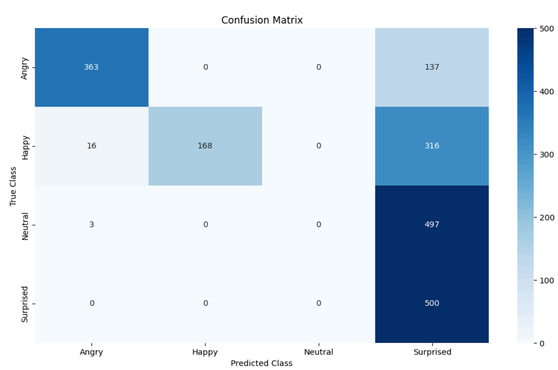


Figure 2: Confusion matrix for the first iteration

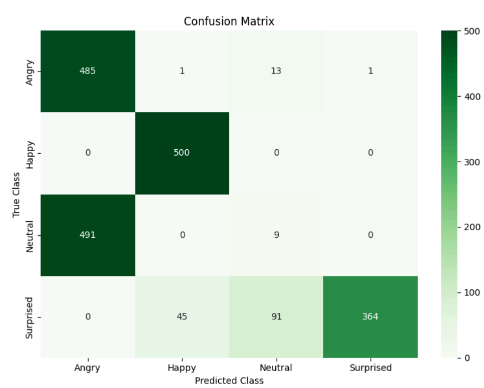


Figure 3: Confusion matrix for the third iterationc, showing signs of overfitting.

Conclusions

5.1 Summary

This project developed a machine learning-based system to detect four emotions Angry, Happy, Neutral, and Surprised from facial images. The goal was to meet the growing demand for emotion-aware systems in fields like healthcare, education, security, and HCI.

Data was collected from 12 Bangladeshi volunteers using a webcam. Images were cropped, resized to 224x224, and augmented (flip, rotate, zoom, brightness) to improve model performance. A total of 4,000 images were used for training.

We used ResNet50 for its balance of accuracy and efficiency. In early training, only final layers were trained. Accuracy reached 76% in the second phase by unfreezing more layers. However, full unfreezing dropped accuracy to 51% due to overfitting.

Tools used included Python, OpenCV, TensorFlow, Keras, and Google Colab. The model performed best on Happy and Neutral, while Angry and Surprised were harder to classify.

5.2 Limitations

- Dataset limited to 12 volunteers, all in controlled indoor conditions
- Only 4 emotions were included (Sad, Fear, Disgust excluded)
- Overfitting occurred when all layers were trained
- No real-time or dynamic testing was performed
- Relied on Google Colab (internet + GPU) for training
  Evaluation used only accuracy and confusion matrix, no precision, recall, or F1-score

5.3 Future Improvements

- Expand dataset with more diversity (age, gender, lighting, accessories)
- Add more emotions like Sad, Fear, and Disgust
- Enable real-time detection using OpenCV and threading
- Test lighter models like EfficientNet or MobileNet

References

1. [1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. CVPR.

2. [2] Goodfellow, I., et al. (2013). Challenges in representation learning. Neural Networks.

3. [3] Mollahosseini, A., et al. (2016). AffectNet: A Database for Facial Expression Computing. IEEE Transactions on Affective Computing.

4. [4] Zhang, K., et al. (2017). Facial expression recognition with deep residual networks. IEEE Trans. Multimedia.

5. [5] Parkhi, O. M., et al. (2015). Deep Face Recognition. BMVC.