**Final Project:**

**Facial Emotion Recognition on FER2013 Dataset Using Deep Learning**

**Submitted By:**

**Surabhi Kharote**

**Email:**

**surabhi.kharote0708@gmail.com**

**Submitted To:  
IOT Academy, IIT Guwahati**

**Date:  
25/03/2025**

**Facial Emotion Recognition on FER2013 Dataset Using Deep Learning**

**Brief on the Project:**

This project focuses on developing a deep learning model for facial emotion recognition using the FER2013 dataset. Facial emotion recognition is a significant task in computer vision with wide-ranging applications, such as human-computer interaction, mental health analysis and customer feedback systems. The primary problem addressed in this project is the accurate classification of human facial emotions into seven categories: angry, disgust, fear, happy, neutral, sad, and surprise.

**Project Type:**

This is a machine learning project that leverages deep learning techniques, specifically Convolutional Neural Networks (CNNs) and transfer learning, to classify emotions from facial images.

**Problem Statement:**

The challenge is to build a model that can accurately classify facial emotions despite the inherent difficulties in the FER2013 dataset, such as class imbalance, label noise, and low-resolution images (48x48 pixels). The goal is to achieve a test accuracy of at least 80% on the FER2013 dataset.

**Motivation:**

Facial emotion recognition has significant real-world applications. For instance, in mental health, it can help detect emotional states for early intervention. In customer service, it can analyze customer reactions to improve services. The problem is interesting because it combines computer vision and deep learning to interpret human emotions, a task that is inherently complex due to the variability in facial expressions and dataset challenges.

**Previous Work:**

Previous studies on the FER2013 dataset have shown varying levels of success. For example, a study by Goodfellow et al. (2013) introduced the FER2013 dataset and achieved a baseline accuracy of around 65% using a simple CNN. More recent works have explored transfer learning with models like VGG16 and ResNet50, but they often struggle with the dataset’s low-resolution images and label noise. This project builds on these efforts by comparing a custom CNN with transfer learning models to identify the most effective approach.

**Tentative Approach:**  
  
The approach involves:

**1. Data Preprocessing:**   
  
Use data augmentation (e.g., horizontal flips, width/height shifts) to increase dataset diversity and address class imbalance.

**2. Model Development:**   
 Built a CNN model.

**3. Training and Evaluation:**Train the models on the FER2013 dataset and evaluate them using accuracy, confusion matrices, and classification reports.

**4. Analysis:**  
Compare the performance of models to identify the best approach and suggest improvements.

**Deliverables of the Project**

The project deliverables include:

**1. Trained Models:**

- A custom CNN model tailored for the FER2013 dataset.

**2. Performance Metrics:**

- Test accuracy for each model.

- Confusion matrices to analyze misclassifications.

- Classification reports detailing precision, recall, and F1-score for each emotion class.

- Visualizations of predictions to show the model’s performance on sample images.

**3. Analysis and Report:**

- A detailed comparison of the three models.

- Insights into the dataset challenges (e.g., class imbalance, label noise).

- Recommendations for future improvements.

**4. Evaluation Evidence:**

- The models will be evaluated on the test set of the FER2013 dataset (7,178 images).

- Evidence will include quantitative metrics (accuracy, F1-scores) and qualitative analysis (visualizations of predictions).

The evaluation will focus on achieving a test accuracy of at least 80%, with detailed analysis to understand why the target was or was not met.

**List of Questions Your Model/Problem Are Designed to Answer**

1. Can a custom CNN achieve a test accuracy of 80% or higher on the FER2013 dataset?

3. What are the main challenges in the FER2013 dataset that affect model performance (e.g., class imbalance, label noise)?

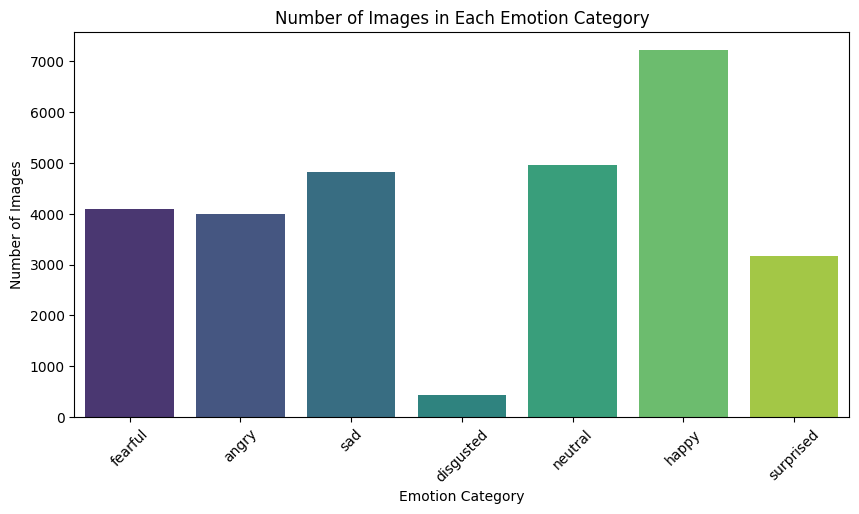
4. Which emotions are most difficult to classify, and why?

5. How can data augmentation and regularization techniques improve model performance on a noisy dataset?

6. What strategies can be used to improve the performance of facial emotion recognition models in the future?

**Explore and Pre-process Data:**

The FER2013 dataset, used for facial emotion recognition, contains 35,887 grayscale images (48x48 pixels) across seven emotions: angry, disgust, fear, happy, neutral, sad, and surprise. It is split into a training set (28,709 images) and a test set (7,178 images), organized into train and test directories.



Above fig no.1 Number of Images in Each Emotion Category

**Exploratory Data Analysis (EDA)**:

* **Class Distribution**: The training set’s class distribution was visualized (Figure 1), showing:
  + Happy: 7,215
  + Neutral: 4,965
  + Sad: 4,830
  + Fear: 4,097
  + Angry: 3,995
  + Surprise: 3,171
  + Disgust: 436 This imbalance, with “disgust” having the fewest images, risks model bias toward majority classes.
* **Sample Images**: One image per emotion was displayed (Figure 2), revealing challenges like low resolution, poor lighting, and potential label noise.



Above fig no.2 Sample Images from Each Emotion Category

**Preprocessing**:

* **Data Augmentation**: Using ImageDataGenerator, the training set was augmented with horizontal flips, 10% width/height shifts, and rescaling (to [0, 1]). A 20% validation split was applied. The test set was only rescaled.
* **Generators**: Training (22,968 images), validation (5,741 images), and test (7,178 images) generators were created with a batch size of 64, grayscale mode, and categorical labels.

**Create Model**

A Custom CNN was designed for the FER2013 dataset’s 48x48 grayscale images:

* **Input Layer**: 48x48x1 grayscale images.
* **Conv2D Layers**:
  + 64 filters (3x3), ReLU, BatchNormalization, MaxPooling2D (2x2), Dropout (0.25).
  + 128 filters (5x5), ReLU, BatchNormalization, MaxPooling2D (2x2), Dropout (0.25).
  + 512 filters (3x3, L2 reg 0.01), ReLU, BatchNormalization, MaxPooling2D (2x2), Dropout (0.25).
  + 512 filters (3x3, L2 reg 0.01), ReLU, BatchNormalization, MaxPooling2D (2x2), Dropout (0.25).
* **Dense Layers**:
  + 256 units, ReLU, BatchNormalization, Dropout (0.25).
  + 512 units, ReLU, BatchNormalization, Dropout (0.25).
* **Output Layer**: 7 units (softmax for 7 emotions).

The model was compiled with:

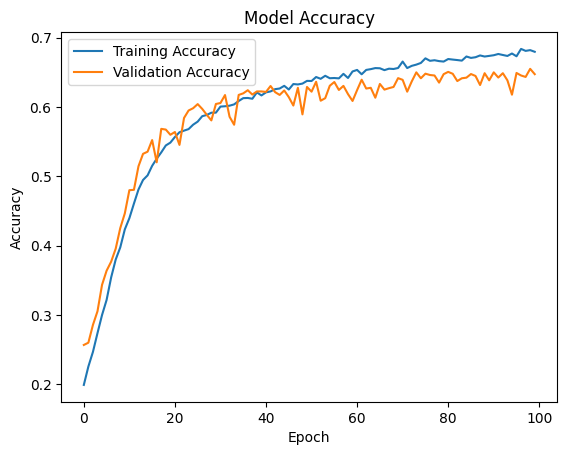
* Optimizer: Adam (learning rate 0.0001).
* Loss: Categorical cross-entropy.
* Metric: Accuracy.

Training was performed for 100 epochs with a batch size of 64. The training history (Figure 3) shows validation accuracy peaking at 65.51% (epoch 99).

**Model Evaluation**

The Custom CNN was evaluated on the test set (7,178 images). The test accuracy was 65.10%, as reported by model.evaluate(test\_generator).

Since the confusion matrix and classification report were not provided in this code, further evaluation metrics (e.g., precision, recall, F1-score) could not be computed. However, the test accuracy indicates the model’s overall performance on unseen data.



Above fig no.3 Training and Validation Accuracy Over Epochs

**Analysis of Results**

The Custom CNN achieved a test accuracy of 65.10%, below the target of 80%. Key observations:

* **Training Dynamics**: The training and validation accuracy plot (Figure 3) shows steady improvement, with validation accuracy peaking at 65.51% (epoch 99). The close alignment of training and validation accuracy indicates minimal overfitting, thanks to BatchNormalization, Dropout (0.25), and L2 regularization (0.01).
* **Class Imbalance**: The dataset’s imbalance (e.g., “disgust” with only 436 images) likely impacted performance on minority classes.
* **Challenges**: Low resolution (48x48 pixels), label noise, and class imbalance limited the model’s ability to generalize effectively.

**Report Writing**

This report summarizes the facial emotion recognition task using the FER2013 dataset, aiming for an 80% test accuracy with applications in human-computer interaction.

**Deliverables**:

* Trained Custom CNN model.
* Performance metric: Test accuracy (65.10%).
* Visualizations: Class distribution (Figure 1), sample images (Figure 2), and training history (Figure 3).

**Evaluation Approach**: The model was evaluated on the test set using accuracy. Additional metrics (e.g., confusion matrix, classification report) were not computed due to missing data.

**Model Details**: The Custom CNN used four Conv2D layers (64 to 512 filters), BatchNormalization, Dropout (0.25), and L2 regularization (0.01) to prevent overfitting. Data augmentation helped address class imbalance.

**Findings**:

* Test accuracy (65.10%) did not meet the 80% target.
* Class imbalance and dataset challenges (low resolution, label noise) impacted performance.

**Resource:**

Dataset Source:

- Dataset: FER2013

- Source: Kaggle

- URL: [https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer](<https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer>)

- Description:   
  
The FER2013 dataset contains 35,887 grayscale images of size 48x48 pixels, categorized into 7 emotions: angry, disgust, fear, happy, neutral, sad, and surprise. It is divided into:

- Training Set: 28,709 images

- Test Set: 7,178 images

- Challenges:

- Class imbalance (e.g., “disgust” has only 436 training images, while “happy” has 7,215).

- Label noise due to mislabeled images.

- Low-resolution images (48x48 pixels).

Software

- Platform: Google Colab (with GPU support for faster training).

- Programming Language: Python.

- Libraries:

- TensorFlow/Keras: For building and training deep learning models.

- NumPy, Pandas: For data manipulation.

- Matplotlib, Seaborn: For visualization (e.g., confusion matrices, training plots).

- Scikit-learn: For evaluation metrics (e.g., confusion matrix, classification report).

- OpenCV (cv2): For image processing and visualization.

- Environment Setup:

- Used Google Colab to leverage free GPU resources.

**References**

1. Goodfellow, I. J., et al. (2013). “Challenges in Representation Learning: A Report on Three Machine Learning Contests.” \*Neural Networks\*, 64, 59–63. [This paper introduced the FER2013 dataset and provided a baseline for facial emotion recognition.]

2. <https://github.com/akmadan/Emotion_Detection_CNN/blob/main/emotion-classification-cnn-using-keras.ipynb>

3. <https://chatgpt.com/>