Project Report

Title: Facial Expression Recognition using Convolutional Neural Networks (CNNs)
Internship: Coding Samurai
Prepared by: Henil Parag Patel
Date: 18-08-2025

1. Abstract

This project focuses on developing a **machine learning model** to classify facial expressions into categories such as *Happy, Sad, Angry, Neutral, and Surprise*. Using **Convolutional Neural Networks (CNNs)**, the system was trained on grayscale images of human faces. The dataset used was the FER-2013 dataset, which contains labeled facial expression images. The project pipeline covers **data preprocessing, model building, training, evaluation, and real-time emotion detection using OpenCV**. Results demonstrate that CNNs are effective for facial expression recognition, achieving promising accuracy in both training and validation, with the ability to generalize to unseen data.

2. Introduction

Facial expressions are a primary mode of non-verbal communication and are essential in human interaction. Automating facial expression recognition has applications in **healthcare, human-computer interaction, security, and entertainment**.
The goal of this project is to design a CNN-based system that:

1. Learns patterns from facial images.
2. Classifies expressions into predefined categories.
3. Detects and predicts expressions in real-time using webcam input.

3. Objectives

- To preprocess and prepare the FER-2013 dataset for training.
- To build a CNN model capable of classifying facial expressions.
- To evaluate the model using accuracy and loss metrics.
- To test the system in real-time with webcam input.
- To analyze model performance and suggest future improvements.

4. Literature Review

Several studies have applied machine learning to facial expression recognition. Traditional approaches (e.g., SVMs with handcrafted features) often fail to capture complex patterns in facial images. Deep learning methods, particularly CNNs, outperform classical methods by automatically learning spatial hierarchies of features. Prior research has also utilized LSTMs for temporal emotion recognition in videos, and ensemble models for higher accuracy. In this project, we implement CNNs for simplicity, interpretability, and effectiveness.

5. Methodology

5.1 Data Collection

- Dataset: **FER-2013 (Facial Expression Recognition dataset)**
- Classes: Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral
- Image size: 48x48 grayscale images

5.2 Data Preprocessing

- Normalized pixel values (0–255 → 0–1).
- Converted images to grayscale.
- Augmented training data using **ImageDataGenerator** (rotation, zoom, flips).
- Split dataset into **80% training, 20% validation**.

5.3 Exploratory Data Analysis (EDA)

- Distribution of samples across emotion classes visualized.
- Sample images displayed per category.
- Observed class imbalance (e.g., fewer disgust images compared to happy/neutral).

5.4 Model Development

- **CNN Architecture**:
  - Conv2D → MaxPooling (multiple layers)
  - Flatten → Dense → Dropout
  - Output Layer with **softmax activation**
- **Loss Function:** Categorical Crossentropy
- **Optimizer:** Adam
- **Batch Size:** 32
- **Epochs:** 10

5.5 Model Evaluation

- Metrics: Training accuracy, Validation accuracy, Loss curves
- Confusion matrix for class-level performance

5.6 Real-Time Prediction

- Integrated **OpenCV Haar Cascade** for face detection.
- Preprocessed face region → model prediction.
- Labeled webcam feed with predicted expression.

6. Results

Model	Training Accuracy	Validation Accuracy
CNN	~80%	~70-72%

- Training and validation loss decreased consistently across epochs.

- Validation accuracy plateaued around ~70%, which is consistent with literature for FER-2013 baseline CNNs.
- Real-time testing correctly identified most common expressions (*happy, sad, angry*) but struggled with subtle classes (*disgust, fear*).

7. Discussion

- **Strengths:**
  - CNN automatically extracts facial features.
  - Works in real-time with webcam input.
- **Limitations:**
  - Misclassifications in subtle emotions.
  - Dataset imbalance affects model performance.
- **Future Improvements:**
  - Use advanced architectures (ResNet, MobileNet, or EfficientNet).
  - Apply transfer learning for better accuracy.
  - Include temporal modeling (CNN+LSTM) for video-based recognition.
  - Balance dataset using oversampling or synthetic data.

8. Conclusion

This project successfully implemented a **CNN-based facial expression recognition system**.
The model achieved **~70% validation accuracy** and demonstrated reliable **real-time recognition** of common facial expressions. While there are limitations in distinguishing subtle emotions, the project shows the potential of deep learning in **emotion recognition applications**. Future work can extend this to more robust models and multimodal emotion recognition systems.

9. References

- FER-2013 Dataset: Kaggle FER-2013 Dataset

- TensorFlow/Keras Documentation: https://www.tensorflow.org/
- OpenCV Documentation: https://docs.opencv.org/
- Goodfellow et al., *Deep Learning*, MIT Press, 2016

10. Appendix

- The full Jupyter Notebook (`facial_expression_recognition.ipynb`) is attached.
- Python script (`facial_expression_recognition.py`) provided for execution.
- Training/validation accuracy and loss plots are included in project files.
- Screenshots of real-time emotion detection results are attached.