Facial Expression Recognition

***Abstract*—This report details the development of a Machine Learning (ML) model for Facial Expression Recognition (FER) employing a Convolutional Neural Network (CNN). Our goal was to craft a model adept at identifying human emotions through CNN techniques. Designed with a four-layer architecture—comprising three inner layers and one outer layer—the model discerns seven core emotions: happy, sad, fear, disgust, surprise, neutral, and angry. The resulting system attains an accuracy of 71.08%, alongside a precision of 71.90%, a recall of 71.08%, and an F1-score of 71.97%. Looking ahead, we aim to refine the model for greater accuracy.**

I. Introduction

Facial Expression Recognition (FER) is part of computer vision (CV) and is useful for things like mental health anal- ysis and customer feedback systems. Human emotions like happy, sad, and angry are common, but detecting them with a machine is tough. To solve this, we built a Convolutional Neural Network (CNN) model with four layers to spot seven emotions: happy, sad, angry, surprise, neutral, disgust, and fear. Our goal was to make this CNN model and check its performance with metrics like accuracy, precision, recall, and F1-score. We aimed for at least 65% accuracy and got 71% by the end. This project focuses only on emotions from face images, but later we want to make a model for real-time detection.

II. Data Understanding and Preparation

At the start of this project, we examined the dataset to understand its contents for training our model to detect seven emotions: happy, sad, fear, disgust, surprise, neutral, and angry.

We focused on three main questions:

(1) Is the number of images the same for all emotions, or is it uneven?

(2) How do we handle it if some emotions have fewer images?

(3) What is the size of the images?

We conducted Exploratory Data Analysis (EDA) on the images. The results showed an uneven distribution: The distribution was as follows:

“happy”	7285
“disgust”	436
“surprise”	3171
“fear”	4107
“angry”	3995
“neutral”	4965
“sad”	4830

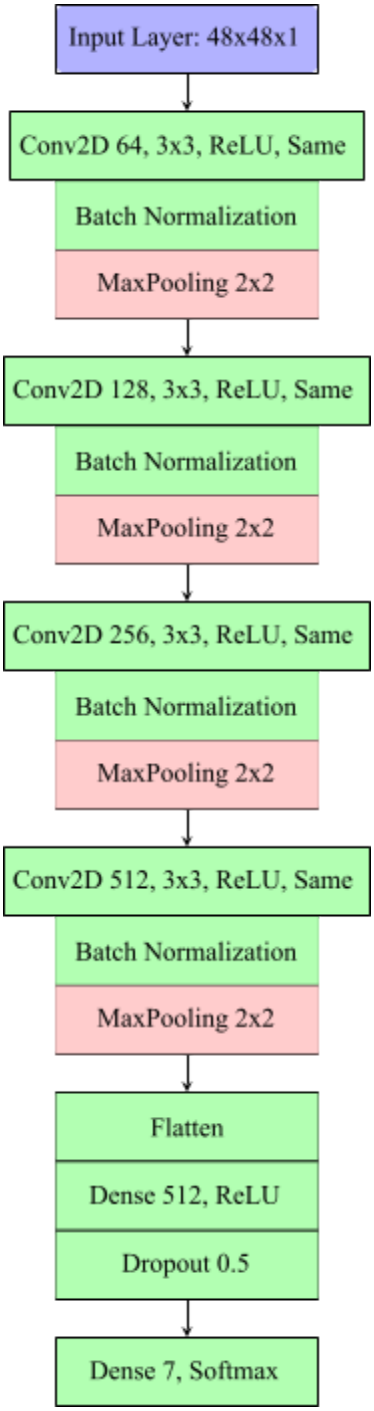
To address this, we used *Augmentor* [1] for data augmentation. We flipped, rotated, and zoomed the images of “disgust” and “surprise” to increase their numbers. After this, we had enough images for “disgust” and “surprise”—not ideal but workable—while “happy” stayed high and the rest were balanced. Next, we checked image size by sampling a few randomly. The images were resized to 48×48 pixels to reduce the computational cost.

In short, we found an uneven dataset, improved it with augmentation using Aug, and set all images to 48×48 pixels for the next steps.

III. Model Architecture

The CNN model architecture is illustrated in the flow chart. It extracts features from images and employs 3 × 3 filters and max-pooling to down-sample the feature maps.

The model was trained using the Adam optimizer with a learning rate scheduler to improve convergence [2]. Sparse categorical cross-entropy was used as the loss function, and accuracy was used as the primary evaluation metric. The training was carried out with a batch size of 64 over 71 epochs.



Initial tests with the architecture on default parame- ters showed promising results, achieving validation accuracy 65.26%. However, the training accuracy exceeded 99%, leading to overfitting of the model.

We used regularization methods like “Dropout” [3] and “Batch Normalization” [4] to tackle overfitting.

Challenges:

- **Overfitting** – The model exhibited high training accuracy but significantly lower test accuracy, indicating poor generalization.
- **Imbalanced Class** – Emotions like ’Disgust’ had fewer training samples, leading to lower classification accuracy for these categories.
- **Complexity** – The slight variations in facial expressions obtained through augmentation made it difficult for a simple CNN to generalize properly.
- **Misclassification Issues** – Expressions like ’fear’ and ’sad’ or ’angry’ and ’fear’ were frequently confused, which affected overall performance.

IV. Evaluation

```
Model Evaluation Metrics:
Accuracy: 0.7108
Precision: 0.7190
Recall:    0.7108
F1 Score: 0.7197
```

V. Future Improvements

1) **Use of Deeper CNN Architectures** – Implementing advanced models can enhance feature extraction and improve classification accuracy.
2) **Real-World Deployment** – The model can be integrated into applications such as Human-Computer Interaction, mental health monitoring, and smart surveillance.
3) **Addressing Overfitting** – Applying regularization techniques and data augmentation can improve generalization and prevent overfitting.
4) **Integrated Emotion Analysis** – Integrating facial expressions with audio cues or physiological signals can improve recognition accuracy and robustness.

By addressing these challenges and implementing these improvements, we can enhance the model’s performance and robustness, making it more suitable for real-world applications.

VI. Conclusion

In this project, we successfully implemented a Facial Expression Recognition (FER) system using Convolutional Neural Networks (CNNs). The model was trained to classify facial expressions into multiple categories (emotions), incorporating data augmentation, batch normalization, and dropout to enhance generalization. The model achieved a training accuracy of over 79%, and the test accuracy reached 71.08%.

References

[1] M. Bloice, P. Roth, and A. Holzinger, ”Augmentor: Image Augmentation Library,” 2017. [Online]. Available: GitHub: Augmentor Library

[2] CodeSignal, ”Learning Rate Scheduling in PyTorch,” PyTorch Techniques for Model Optimization. [Online]. Available: CodeSignal: Learning Rate Scheduler

[3] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, ”Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *JMLR*, vol. 15, no. 1, pp. 1929-1958, 2014. [Online]. Available: JMLR: Dropout Paper

[4] S. Ioffe and C. Szegedy, ”Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *ICML*, pp. 448-456, 2015. [Online]. Available: ICML: Batch Normalization