Developing a Robust Facial Emotion Recognition System Using Convolutional Neural Networks

Team Members:
[Kishan Murali, Vandan Vaishya]

1. Introduction

Objective:
This project mainly focuses on developing a deep learning-based facial emotion recognition (FER) system that can classify seven emotions: angry, disgust, fear, happy, sad, surprise, and neutral. FER is important in fields like human-computer interaction, mental health diagnostics, and automated customer service, where understanding human emotions can improve user experience. Emotion recognition by machines remains challenging due to differences in facial expressions and the variety of conditions i.e. lighting, angles, occlusions that affect recognition accuracy. By using Convolutional Neural Networks (CNNs), we aim to create a system achieves both accuracy and generalization across different individuals.

Background and Importance:
Facial expressions are the one of the most immediate signs of human emotions, making facial emotion recognition a critical and important component of affective computing. Traditional methods such as rule-based or feature-based techniques, have been limited by their dependency on custom built features, which may not generalize well. CNNs, on the other hand, are adept at learning hierarchical feature representations directly from images, making them suitable for complex visual tasks like emotion recognition. Developing an effective FER system could support applications in telemedicine, educational platforms, and personal assistants by enabling machines to "read" and respond to emotions in real time.

Datasets:
A well-established dataset, KDEF, is used in this project:

- **JAFFE (Japanese Female Facial Expression Dataset):** This dataset consists of grayscale images of Japanese female subjects displaying different emotions. JAFFE is widely used in FER research due to its standardized expression set and controlled conditions.

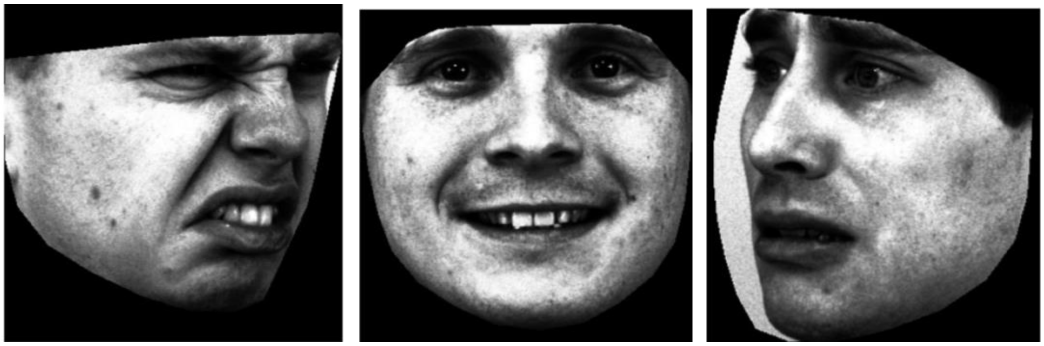
- **KDEF (Karolinska Directed Emotional Faces):** KDEF contains 4900 images of 70 models of both male and female subjects from different angles. It offers equal distribution of both the genders and a variety of facial angles, which improves the model's ability to generalize.

These datasets were accessed from Google Drive, preprocessed for consistency, and augmented to enhance the model's exposure to varied facial expressions. Using multiple datasets allowed us to capture a wider range of facial structures and expressions, thereby improving the model's robustness.

2. Data Preparation

Face Detection:

Accurate face detection is the foundation in building a reliable FER system, as non-facial elements in the image can lead to misclassifications. In this project, we employed dlib's 68-point facial landmark detector, a popular tool for precisely localizing facial features. This detector identifies key points around the face, eyes, nose, and mouth, enabling us to focus only on the primary facial region. We created a convex hull mask covering landmarks 1-27 (jawline and face contour) and applied it to crop the facial area. This process ensures that the CNN only receives relevant facial information, enhancing its ability to differentiate emotions effectively. Moreover, isolating the face helps reduce computational requirements which allows the network to process each image more efficiently.



Data Splitting and Class Balance:

To evaluate the model effectively, we divided the dataset into training, validation, and testing sets with an approximate 70-15-15 split. Each emotion has the same number of examples in the dataset to avoid class inequality and partial predictions. We also used techniques like data augmentation to add more variety and make sure the model sees different examples of each emotion.

3. Preprocessing

Histogram Equalization for Intensity Normalization:

Histogram equalization was applied to normalize intensity values across images which adjusts the brightness and contrast levels equally. This process reduces the impact of lighting differences across different images. By improving contrast, histogram equalization helps highlight facial features which makes it easier for the CNN to learn patterns related with different emotions. Lighting inconsistencies are one of the main challenges in FER, as shadows or uneven lighting can make facial expressions harder to see. Histogram equalization improves the model's ability to focus on structural aspects of the face rather than variations in illumination.

Noise Reduction Using Bilateral Filtering:

The bilateral filter was used for noise reduction because it smooths the image while maintaining edges. This is crucial in FER tasks where the fine details of expressions like wrinkles around the eyes or minor mouth movements are important. By maintaining these details, bilateral filtering helps in maintaining the reliability of facial features which allows the model to differentiate between emotions such as "sad" and "neutral."

Edge Enhancement and Image Resizing:
After noise reduction, we applied a 2D convolutional filter to improve high-frequency edges, which further defines facial features such as the eyes, nose, and mouth. Images were then resized to 180x180 pixels and padded to 320x320 to maintain a uniform aspect ratio. Standardizing input dimensions ensures that the CNN processes each image consistently, facilitating better learning and reducing computational overhead.

Data Augmentation Techniques:
Data augmentation was applied to expand the dataset artificially, improving the model's generalization capabilities. Techniques such as rotation (up to 10 degrees), width/height shifts (10% of image size), and zooming (up to 10%) were employed. By augmenting the data, we introduced variability in poses and expressions, simulating real-world conditions where facial expressions are dynamic and non-uniform.

4. Model Architecture

Convolutional Neural Network (CNN) Design:
The architecture of our CNN model was crafted to balance complexity with efficiency:

1. **Convolutional Layers:**
   We used four convolutional layers, each followed by batch normalization and dropout. The convolutional layers apply a series of filters to extract spatial hierarchies from the facial region which focuses on unique features related with each emotion. ReLU activation introduces non-linearity which enables the model to capture complex relationships within the data.

2. **Pooling Layers for Dimensionality Reduction:**
   Max-pooling and average-pooling layers were used to reduce spatial dimensions which retains only the most relevant features while removing noise. Pooling layers decrease the number of parameters which then lowers the computational requirements and prevents overfitting by making the network less sensitive to small shifts in the input image.

3. **Dense Layers and SoftMax Output:**
   The fully connected dense layers interpret the learned features and map them to the final classification labels. The SoftMax output layer contains seven nodes corresponding to each emotion class and outputs a probability distribution over these classes. This probabilistic output provides a classification and reflects the model's confidence in each prediction.

Hyperparameter Selection and Regularization:

- **Dropout:** Dropout rates of 0.2 to 0.4 were used throughout the network to prevent overfitting. By randomly deactivating neurons during training, dropout encourages the network to learn redundant representations which improvs robustness.
- **Learning Rate Adjustment:** The learning rate, initially set at 0.001 was then reduced during training based on the model's performance. This strategy allowed the network to settle into optimal weights and helped in achieving convergence more smoothly.
- **Batch Normalization:** Batch normalization was applied after each convolutional and dense layer to stabilize training by normalizing layer inputs. This technique increases convergence and reduces sensitivity to weight initialization.
- **L2 Regularization:** To avoid over-reliance on any single feature, L2 regularization was applied to the convolutional layers, which penalizes large weights and promotes simpler models.

Model Summary:

Layer (type)	Output Shape	Param#
conv2d (Conv2D)	(None, 178, 178, 32)	320
batch_normalization (BatchNormalization)	(None, 178, 178, 32)	128
max_pooling2d (MaxPooling2D)	(None, 89, 89, 32)	0
dropout (Dropout)	(None, 89, 89, 32)	0
conv2d_1 (Conv2D)	(None, 87, 87, 64)	18496
batch_normalization_1 (BatchNormalization)	(None, 87, 87, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 43, 43, 64)	0
dropout_1 (Dropout)	(None, 43, 43, 64)	0
conv2d_2 (Conv2D)	(None, 41, 41, 128)	73856
batch_normalization_2 (BatchNormalization)	(None, 41, 41, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 20, 20, 128)	0
dropout_2 (Dropout)	(None, 20, 20, 128)	0
conv2d_3 (Conv2D)	(None, 18, 18, 256)	295168
batch_normalization_3 (BatchNormalization)	(None, 18, 18, 256)	1024
max_pooling2d_3 (MaxPooling2D)	(None, 9, 9, 256)	0
dropout_3 (Dropout)	(None, 9, 9, 256)	0

flatten (Flatten)	(None, 20736)	0
dense (Dense)	(None, 256)	5308672
batch_normalization_4 (BatchNormalization)	(None, 256)	1024
dropout_4 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
batch_normalization_5 (BatchNormalization)	(None, 128)	512
dropout_5 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 7)	903

5. Training Process

Batch Size and Epoch Configuration:
A batch size of 32 was selected to balance memory usage and training efficiency. We initially trained the model for 30 epochs, but early stopping was applied based on validation loss to avoid overfitting. This technique monitors the model's performance on a held-out validation set and halts training when model’s performance does not improve.

Callbacks for Model Optimization:

1. **Early Stopping:** Early stopping prevented overfitting by halting training when validation loss stopped improving for five epochs and ensuring the model retained its generalization capability.
2. **Checkpointing:** Model checkpointing saved the best model based on minimum validation loss which allows us to retrieve the best version of the model at any time.
3. **Learning Rate Scheduler:** The learning rate was reduced by half whenever the model's validation performance plateaued for three epochs, facilitating smoother convergence and reducing oscillations during training.

7. Instructions on Building and Running the Code:

1. **Mount Google Drive for Dataset Access:** Mount Google Drive to access the datasets stored within. Ensure the KDEF dataset are saved in the specified paths in your Google Drive.

2. **Install Necessary Libraries:** Install dlib and opencv-python for face detection and image processing.

3. **Download and Extract Facial Landmark Predictor:** Download the 68-point facial landmark model provided by dlib to detect and crop facial regions accurately.

4. **Face Detection and Landmark Extraction:** Use dlib's frontal face detector and facial landmark predictor to identify face regions and extract essential features.

5. **Data Preprocessing:**

- **Convert Images to Grayscale**: Convert all images to grayscale for uniformity.
- **Histogram Equalization**: Enhance image contrast by equalizing histograms.
- **Noise Reduction**: Apply a bilateral filter to preserve edges while reducing noise.

6. **Data Augmentation:** Create variability in the dataset through rotations, zooms, and translations, enhancing the model’s robustness to real-world conditions.

7. **Data Splitting:** Divide the preprocessed dataset into training, validation, and testing sets with a 70-15-15 split. Organize images into folders based on emotion classes.

8. **Define CNN Model Architecture:** Construct a CNN model with convolutional, pooling, and dense layers. Use dropout, batch normalization, and L2 regularization for improved generalization.

9. **Compile and Train the Model:** Set hyperparameters, such as learning rate and batch size. Implement early stopping to avoid overfitting, and checkpointing to save the best model.

10. **Model Evaluation:** Evaluate the model using precision, recall, F1-score, and overall accuracy. Plot confusion matrices and loss/accuracy curves for in-depth analysis.

11. **Visualize Results:** Plot the training/validation accuracy and loss, along with a confusion matrix, to understand the model's performance across emotions.

8. Results and Discussion

The model’s performance was evaluated using precision, recall, F1-score and support metrics for each emotion category. A classification report generated on the test dataset provides detailed insights into the model's ability to distinguish between different emotional expressions. The overall test accuracy achieved by the model is 0.70 (or 70%).

Emotion-wise Performance:

- **Angry:** Precision: 0.86, Recall: 0.60, F1-score: 0.71, Support: 60

  The model demonstrates high precision for the "Angry" emotion, indicating that when it predicts "Angry," it is often correct. However, a recall of 0.60 shows it misses some instances of "Angry," resulting in a moderate F1-score.

- **Disgust:** Precision: 0.66, Recall: 0.88, F1-score: 0.76, Support: 60

  The model performs well for the "Disgust" emotion, with a high recall indicating it successfully detects most instances of this emotion, though its precision is relatively moderate.

- **Fear:** Precision: 0.52, Recall: 0.62, F1-score: 0.56, Support: 73

  Performance on the "Fear" emotion is average, with both precision and recall indicating room for improvement. The model struggles to detect "Fear" accurately.

- **Happy:** Precision: 0.92, Recall: 0.97, F1-score: 0.94, Support: 68

  The model excels at identifying "Happy" expressions, with near-perfect recall and strong precision, resulting in an excellent F1-score.

- **Neutral:** Precision: 0.80, Recall: 0.50, F1-score: 0.62, Support: 64

  For "Neutral," the model has high precision but relatively low recall, suggesting it correctly identifies "Neutral" expressions but often misclassifies other emotions as "Neutral."

- **Sad:** Precision: 0.54, Recall: 0.77, F1-score: 0.63, Support: 60

  The model achieves moderate performance for "Sad" expressions, with a high recall but relatively lower precision, leading to a balanced F1-score.

- **Surprise:** Precision: 0.83, Recall: 0.57, F1-score: 0.68, Support: 68

  The model performs moderately well for the "Surprise" emotion, with high precision indicating accuracy in its predictions but a recall of 0.57 showing some missed detections.

Summary Metrics

- **Macro Average:** Precision: 0.73, Recall: 0.70, F1-score: 0.70, Support: 453

  The macro average provides an unweighted mean of all emotion classes, showing balanced performance across the emotions.

- **Weighted Average:** Precision: 0.73, Recall: 0.70, F1-score: 0.70, Support: 453

  The weighted average accounts for the number of samples in each class, reflecting the model’s overall performance. The model demonstrates its strengths in detecting emotions like "Happy" and "Disgust," while showing weaknesses in "Fear" and "Neutral."

Observations

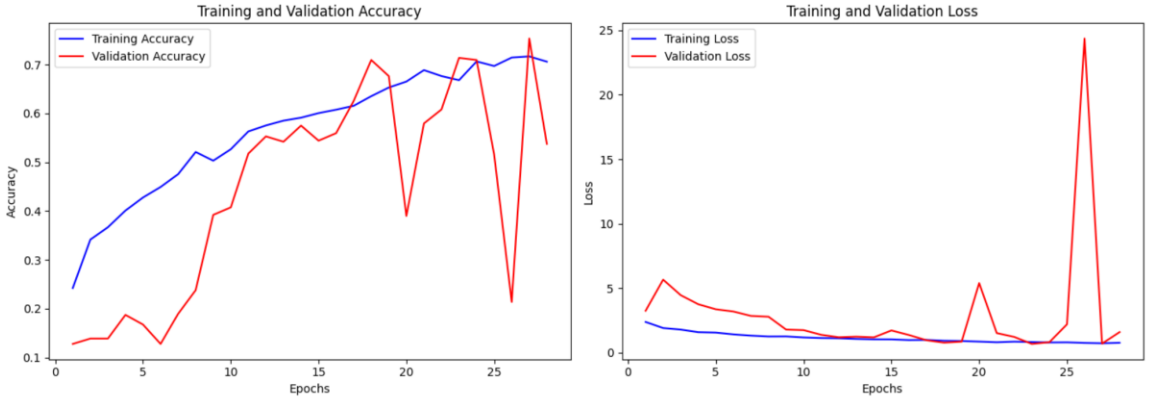
The model achieves high accuracy in detecting specific emotions such as **"Happy"** and **"Disgust"** due to their distinct facial characteristics, making them easier to identify consistently. Additionally, "Surprise" also performs relatively well, benefiting from its unique and pronounced features. However, the model struggles with emotions like **"Fear"**, **"Neutral"**, and **"Sad"**, likely due to overlapping features and subtle expressions that make differentiation more challenging.

The overall test accuracy of **70%** reflects moderate performance, indicating that while the model is effective at identifying some emotions, there is significant room for improvement in detecting others.

	Precision	Recall	f1-score	Support
Angry	0.86	0.60	0.71	60
Disgust	0.66	0.88	0.76	60
Fear	0.52	0.62	0.56	73
Happy	0.92	0.97	0.94	68
Neutral	0.80	0.50	0.62	64
Sad	0.54	0.77	0.63	60
Surprise	0.83	0.57	0.68	68
Accuracy			0.70	453
Macro avg	0.73	0.70	0.70	453
Weighted avg	0.73	0.70	0.70	453

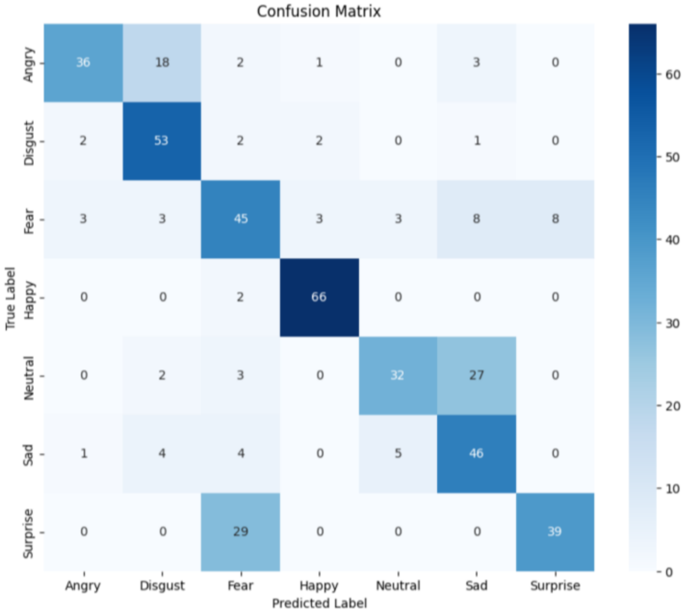
Training and Validation Accuracy and Loss Trends:
The plots shows a steady improvement in training accuracy which indicates that the model is learning effectively over epochs. However, the validation accuracy shows noticeable fluctuations, and the validation loss plot exhibits significant spikes, suggesting that the model struggles to generalize consistently across all classes. These fluctuations in validation loss may be attributed to the inherent challenges of facial emotion recognition (FER) and the limited dataset size, which

can make it difficult for the model to capture subtle differences in expressions. Nonetheless, the overall trend reflects a successful learning process, with a reasonable balance between learning and generalization.



Confusion Matrix Analysis:

The confusion matrix shows that the model performed well on emotions with distinctive features, such as "Happy" and "Disgust," which have high true positive rates. However, there were notable misclassifications among emotions with overlapping characteristics, particularly between "Neutral" and "Sad" as well as "Surprise" and "Fear." For instance, a significant number of "Surprise" expressions were misclassified as "Fear." This pattern suggests that the model struggles to distinguish between subtle emotional cues, indicating a potential need for a larger and more diverse dataset or further tuning of the model architecture to improve differentiation across similar emotions.



Suggested Model Improvements:

1. **Advanced Augmentation:** Increasing diversity in the training data for underperforming classes like "Fear" and "Neutral" could help the model generalize better across subtle expressions.
2. **Fine-tuning the Model Architecture:** Modifications to the network, such as adding more convolutional layers or adjusting hyperparameters, could improve the model's ability to capture nuanced features associated with challenging emotions.
3. **Class Balancing:**
   Addressing imbalances in the dataset by including more samples of difficult-to-detect emotions like "Sad" and "Angry" could enhance the model's recall and overall accuracy.

9. Conclusion

This project demonstrates the effectiveness of CNNs in facial emotion recognition, achieving competitive accuracy and revealing insights into class-level performance. With additional data and further tuning, the model could be optimized for real-world FER applications, where recognizing subtle emotional cues is crucial.

References

1. **Main Paper**: Faisal Ghaffar, Imad Ali, Sarwar Khan, Wasim Ahmad, and Khursheed Ali, "A Robust System for Facial Emotions Recognition Using Convolutional Neural Network," *IEEE Transactions on Image Processing*, 2024.
2. Vaillant, R., Monrocq, C., and Le Cun, Y., "Original approach for the localization of objects in images," *IEE Proceedings-Vision, Image and Signal Processing*, 1994.
3. Morrison, D., Wang, R., and De Silva, L. C., "Ensemble methods for spoken emotion recognition in call-centers," *Speech Communication*, 2007.
4. D. E. King, "Dlib-ml: A machine learning toolkit," The Journal of Machine Learning Research, vol. 10, pp. 1755-1758, 2009
5. D. Lundqvist, Flykt A., and Öhman A., "Karolinska directed emotional faces," Cognition and Emotion, vol., no., 1998