Emotion Recognition System with Adaptive User Interface

SHIMIL SHIJO

OVERVIEW

- Introduction
- Problem Statement
- Related Review
- Dataset Description
- Technologies Used
- Methodology
- Hyperparameter Tuning
- Results
- Discussions
- Conclusion & Future Scope

INTRODUCTION

- Emotion recognition is a crucial aspect of artificial intelligence, enhancing human-computer interaction (HCI).
- Applications - Healthcare, Gaming and Entertainment, Marketing and Customer Service, Education.
- **Objective** : This project aims to classify seven emotions using CNN and integrate an adaptive UI based on detected emotions.
- **Goals** : Improve user experience through a responsive, empathetic system.

PROBLEM STATEMENT

Develop an emotion recognition system that accurately classifies facial expressions into seven categories and integrates a dynamic, adaptive user interface to enhance human-computer interaction.

RELATED WORKS

[1] Proposed a CNN-based emotion recognition model using data augmentation to address class imbalance, improving recognition of minority emotions.

[2] Demonstrated the effectiveness of transfer learning with pre-trained models for subtle emotion recognition on the AffectNet dataset.

[3] Combined CNN and LSTM to capture spatial and temporal features, enhancing recognition of emotions in dynamic scenarios.

[4] Used ensemble learning with multiple classifiers to improve robustness and generalization in emotion recognition on the JAFFE dataset.

[5] Designed a lightweight CNN for real-time emotion recognition on mobile devices, optimizing for computational efficiency.

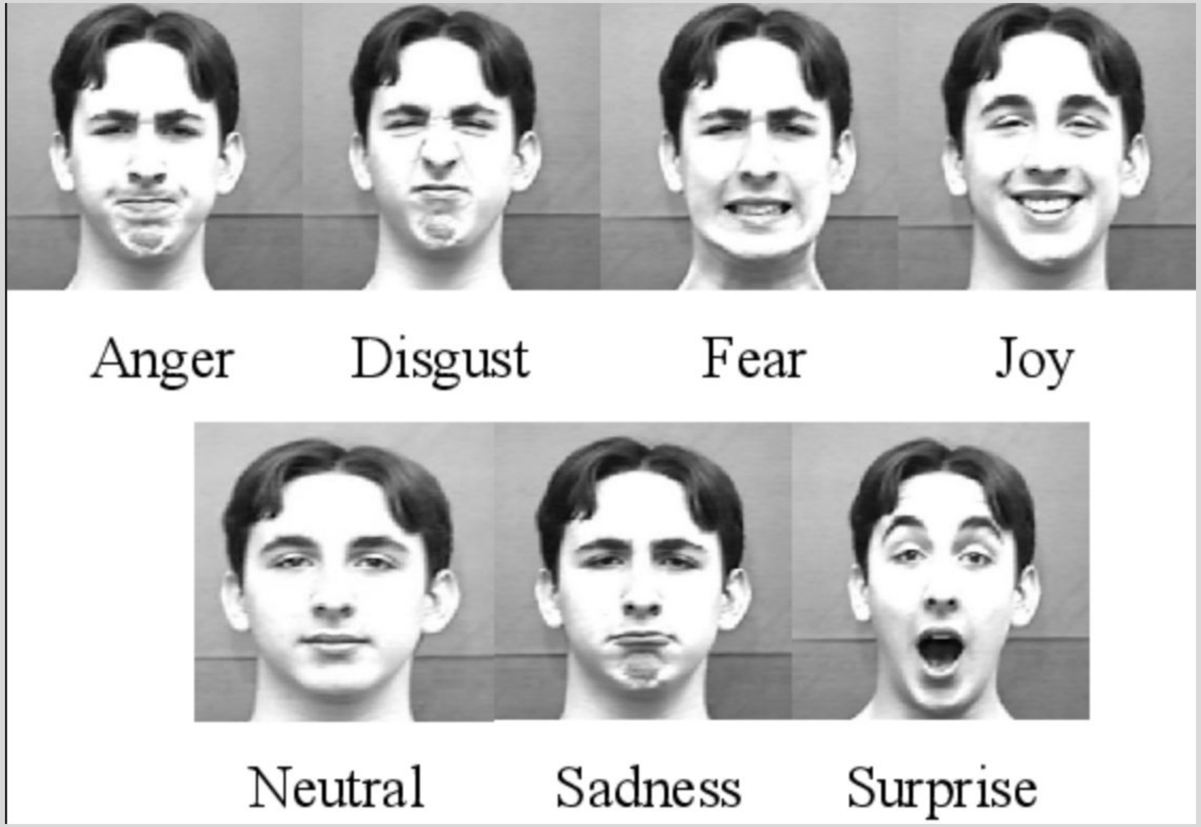
DATA DESCRIPTION

- The FER-2013 dataset(https://www.kaggle.com/datasets/msambare/fer2013) contains facial images classified into seven emotion categories: Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise.

Data Format

- The dataset is organized into two sets:
  - **Train:** Contains 28,709 images across seven emotion classes.
  - **Test:** Contains 7,178 images across the same classes.
- Each image is a grayscale JPEG file with a size of 48x48 pixels.

DATASET ANALYSIS & VISUALIZATION



TECHNOLOGIES

- Python
- Tkinter - for GUI creation (file dialogs and interface components).
- PIL (Pillow) - for image processing and manipulation.
- OpenCV - for image handling and pre-processing.
- TensorFlow & Keras - for building and training deep learning models.
- NumPy - for numerical operations and array handling.
- Matplotlib & Seaborn - for data visualization (plots, confusion matrix).
- Scikit-learn - for evaluation metrics like confusion matrix and classification report

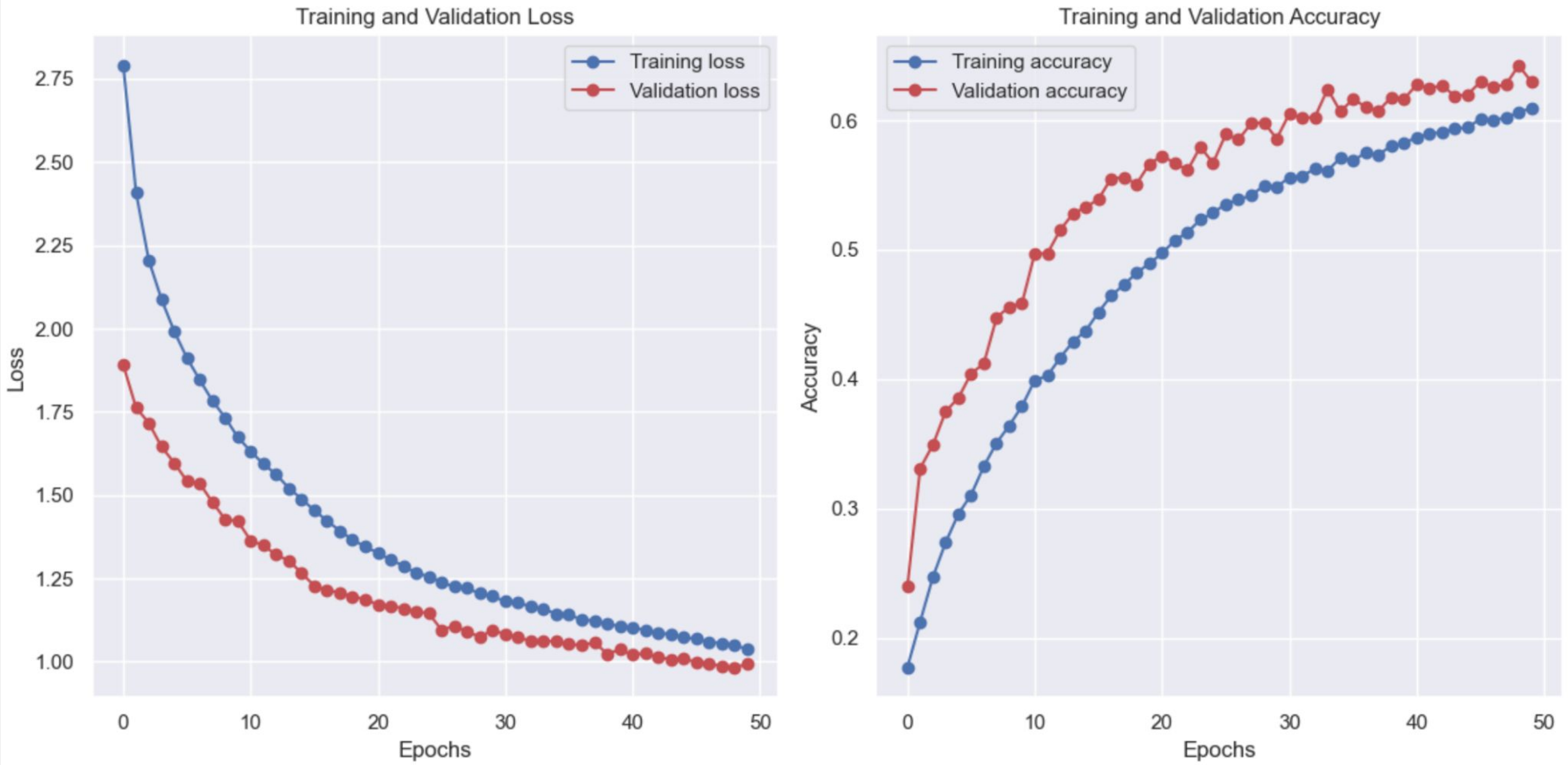
METHODOLOGY

- Used *Convolutional Neural Networks (CNNs)* for facial emotion classification.
- Preprocessing: Normalized grayscale images (48x48 pixels) to stabilize and accelerate training.Also performed Data Augmentation.
- Model Architecture:
  - Six convolutional blocks, each with a Convolutional layer, Batch Normalization, ReLU activation, and Max Pooling.
  - Flatten layer to convert 2D feature maps into a 1D array.
  - Fully Connected Layers with Dense layers, Dropout layers, and a Softmax output layer for seven emotion categories.
- Training: Compiled the model with the *Adam* optimizer, used categorical cross-entropy as the loss function, suitable for multi-class classification tasks.

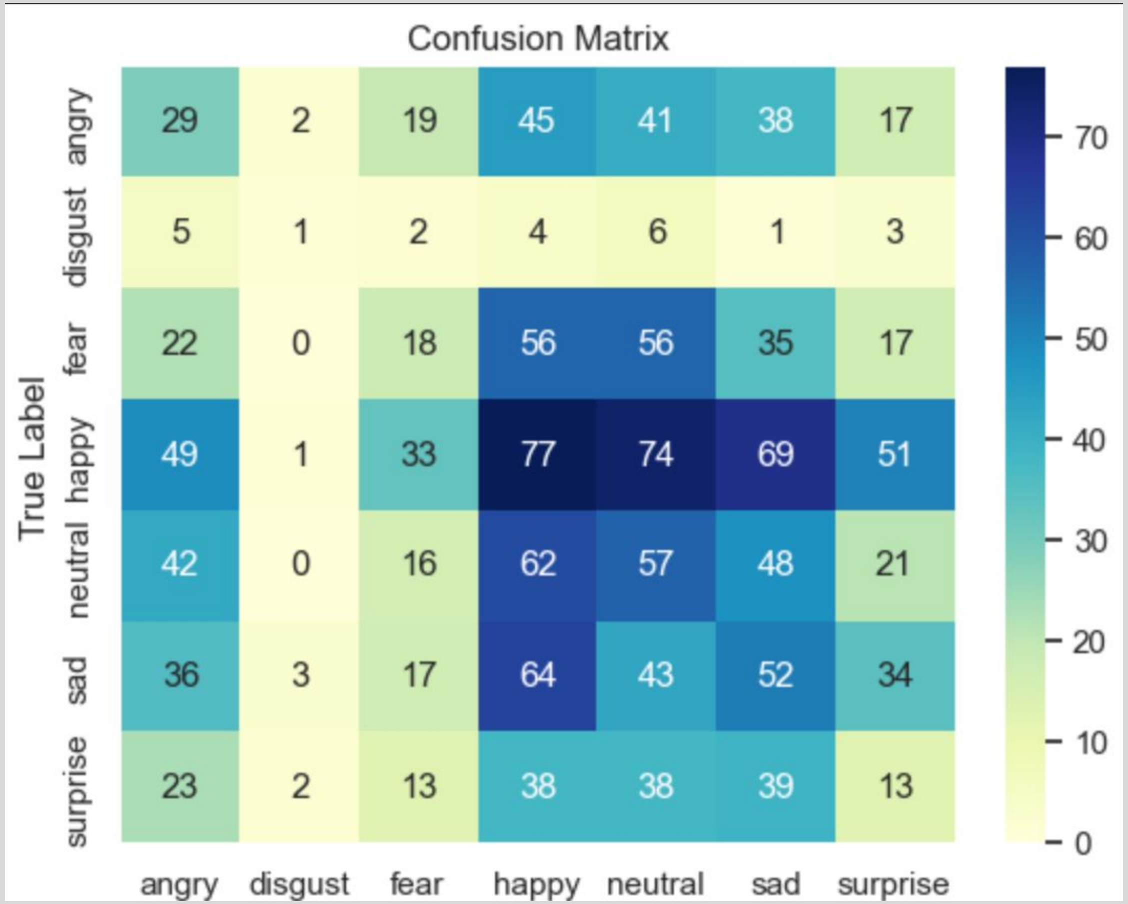
HYPERPARAMETER TUNING

- ***Train-Test Split*** - 80:20
- ***Learning Rate:*** Set to 0.0001 in the Adam optimizer for gradual, stable convergence.
- ***Batch Size:*** Fixed at 32 to balance memory use and training efficiency.
- ***Epochs:*** Defined as 50 for the number of training passes (with early stopping based on validation accuracy).
- ***Convolution Filters:*** Varying numbers (32, 64, 128, 256) capture features at different levels of complexity.
- ***Kernel Size:*** (3, 3) to detect fine patterns in the data.
- ***Dropout Rate:*** 0.25 for convolutional layers and 0.5 for fully connected layers to prevent overfitting.
- ***Callbacks:***
  - ModelCheckpoint saves the best model based on validation accuracy.
  - EarlyStopping halts training if validation accuracy does not improve after a certain number of epochs.

RESULTS - VISUALIZATION

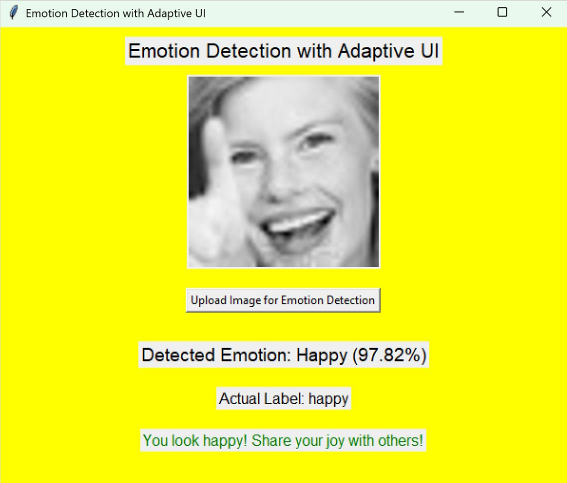


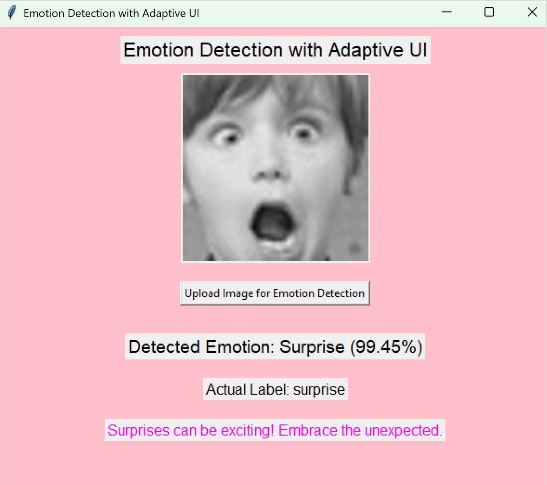
RESULTS - VISUALIZATION

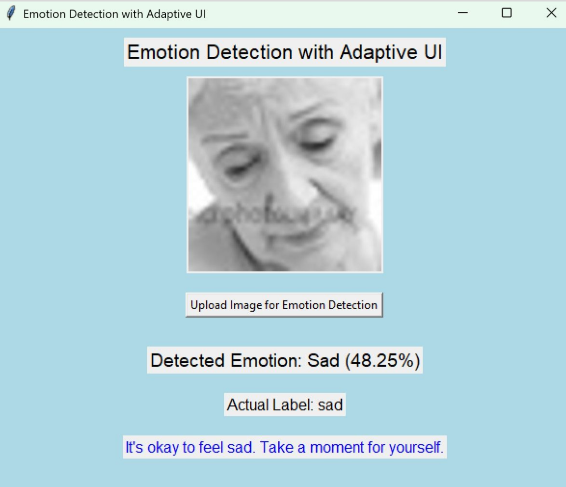


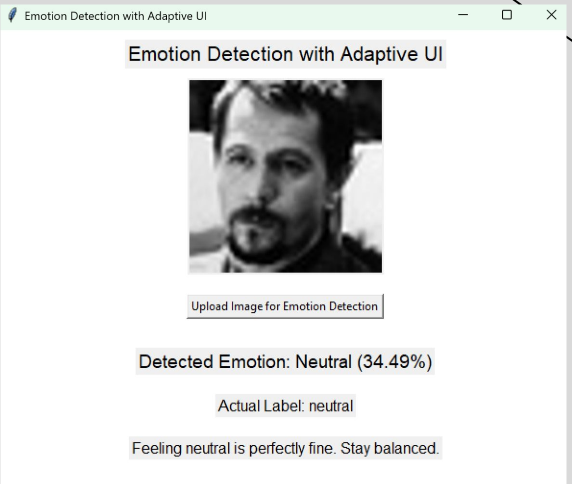

```
23/23 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.6287 - loss: 0.9865
Test Accuracy: 62.57%
23/23 ━━━━━━━━━━━━━━━━━━━━ 4s 157ms/step
Classification Report:
              precision    recall  f1-score   support

       angry       0.16      0.17      0.16       191
     disgust       0.11      0.05      0.06        22
        fear       0.17      0.10      0.12       204
       happy       0.24      0.24      0.24       354
     neutral       0.15      0.19      0.17       246
         sad       0.17      0.19      0.18       249
    surprise       0.11      0.10      0.11       166

    accuracy                           0.17      1432
   macro avg       0.16      0.15      0.15      1432
weighted avg       0.17      0.17      0.17      1432
```

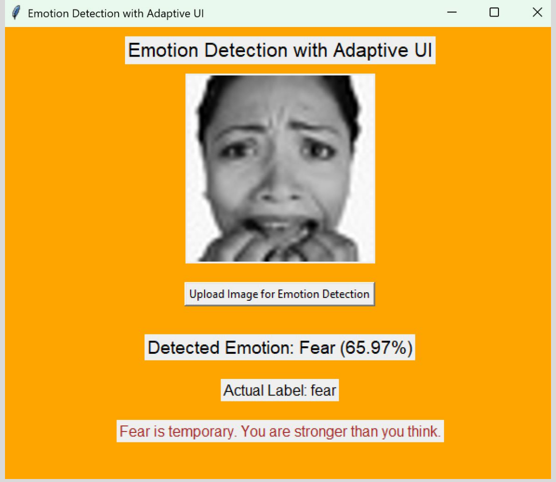
RESULTS - VISUALIZATION (UI)

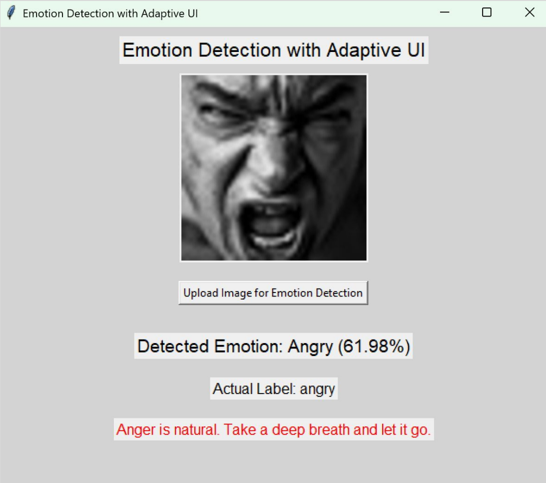


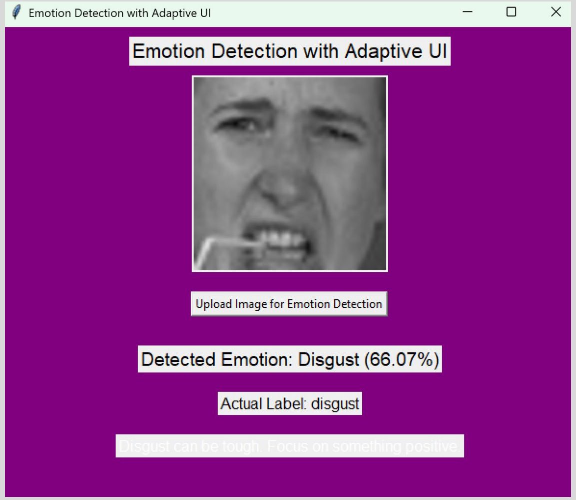












DISCUSSIONS

- *Confusion Matrix:* Shows the model's performance across seven classes. High diagonal values indicate correct predictions, such as 25 for "happy". Some classes like "disgust" have far fewer samples, leading to poor predictions and overrepresented classes like "happy" dominate predictions.
- *Evaluation Metrics Table:* Displays the accuracy, precision, recall, and F1 score. The model performs well overall but struggles with underrepresented classes, like "disgust" and "surprise."
- *Training and Validation Accuracy Graphs:* The training and validation accuracy curves converge and stabilize, showing no overfitting.
- Validation accuracy remained somewhat higher than training accuracy, at about *62.57%*.
- *Training and Validation Loss Graphs:* Loss decreases quickly in the early epochs and stabilizes, indicating effective learning.
- Both training and validation losses remain low, indicating a well-generalized model.

CONCLUSION

- This project developed an AI model to identify the facial emotions and created UI that adjust based on user behavior for enhanced experiences.

Challenges Faced:

- Class Imbalance:Emotions like "Disgust" and "Fear" are underrepresented.
- Low Resolution:Limited details due to small image size.
- Subtle Expressions:Overlap in features between similar emotions (e.g., Neutral and Sad).

Future Scope:

- Optimize for real-time processing: Focus on improving model efficiency for instant, on-the-fly decision-making and feedback.
- Explore multi-modal and user-centric personalization: Integrate diverse interaction methods and personalize the system based on individual user preferences.
- Enhance class balance using advanced techniques like GANs: Use Generative Adversarial Networks to generate synthetic data and address class imbalance issues for improved model accuracy.

REFERENCES

[1] D. L. Keltner and J. J. Gross, “Functional magnetic resonance imaging studies of emotion regulation: A review,” Psychol. Rev., vol. 126, pp.872–886, Oct. 2019.

[2] S. Li, W. Deng, and J. Du, “Deep learning for emotion recognition on small datasets using transfer learning,” in Proc. IEEE Int. Conf. Image Processing (ICIP), 2017, pp. 1192–1196.

[3] P. Ekman and W. V. Friesen, “Facial action coding system: A technique for the measurement of facial movement,” Consulting Psychologists Press, 1978.

[4] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” Ann. Stat., vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[5] Z. Zhang, “Emotion recognition using multi-modal data and machine learning,” Emotion Recognition, 2020, pp. 45–68.

Thank You