Emotion Detection from Uploaded Images

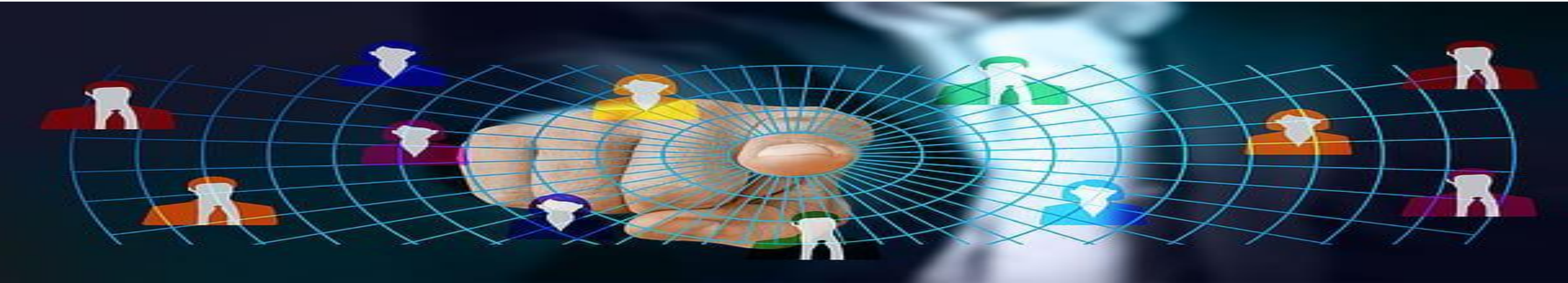
Using CNNs and Facial Feature Extraction

Introduction

Objective:-Developing a system that accurately detects and classifies emotions from facial images using CNNs.

Used In:-

- **Healthcare**: Mental health monitoring.
- **Education**: Understanding student emotions in virtual classrooms.
- **Customer Service**: Monitoring customer satisfaction during interactions.



Problem Statement

The challenge is to **accurately detect human emotions from facial images**. Traditional methods struggle due to:

Variability in Expressions:

- **Micro-expressions**: Tiny, quick facial changes that are hard to catch.
- **Subtle emotions**: Slight differences in expressions (e.g., a small smile vs. a big grin) are difficult to detect.
- **Cultural & Individual Differences**: People express emotions differently based on culture and personality.

External Factors:

- **Lighting**: Poor lighting makes it hard to read faces.
- **Occlusion**: Objects (glasses, hands) can block parts of the face.
- **Image Quality**: Blurry or low-quality images reduce accuracy.

Solution Overview

The proposed solution is to develop a comprehensive **end-to-end system** for emotion detection that combines three core components:

1. **CNN-based Emotion Classification**:
   - Utilize **Convolutional Neural Networks (CNNs)** to classify emotions from facial images.
   - The CNN model will be trained and fine-tuned using the **FER-2013 dataset**.
2. **Facial Detection and Landmark Extraction**:
   - Implement facial detection to identify and isolate the face in the uploaded images.
   - Use tools like **Dlib** or **Mediapipe** to extract key facial landmarks (e.g., eyes, nose, mouth) to improve emotion classification.
3. **User-Friendly Interface**:
   - Design a **Streamlit-based interface** that allows users to upload images easily.
   - Ensure the system is responsive, guiding the user to upload valid image files only.

Key Technologies

- **Programming Language**: Python
- **Framework & Tools**: Streamlit, CNNs (for emotion classification), Dlib/Mediapipe (for facial detection and landmarks)
- **Dataset**: FER-2013 (for training and validation)

Dataset and Preprocessing

Dataset: FER-2013

- The **FER-2013 dataset** consists of **35,887 grayscale images**, each with a resolution of **48x48 pixels**, labeled into **7 emotion categories**:
  - **Angry**, **Disgust**, **Fear**, **Happy**, **Sad**, **Surprise**, and **Neutral**.

Data Split

- The dataset is split into:
  - **Training set**: 80%
  - **Validation set**: 10%
  - **Test set**: 10%

Preprocessing Techniques

- **Rescaling**: All images are resized to the same dimensions (48x48 pixels).
- **Normalization**: Pixel values are normalized to scale them between 0 and 1 for better model convergence.
- **Data Augmentation**:
  - **Random flips**: Horizontal flipping to increase data variety.
  - **Rotations**: Randomly rotating the images to make the model more robust to orientation changes.

Sample Images

Landmark Extraction (Facial Features)

Tools Used

- **Dlib** / **Mediapipe**:
    - These libraries are used to extract **key facial landmarks** such as eyes, nose, mouth, and jawline.
    - Landmarks help identify facial structure and movements, which are critical for emotion analysis.

Role in Emotion Detection

- **Facial landmarks provide significant cues** for detecting emotions.
    - For example:
        - **Raised eyebrows** often indicate **surprise**.
        - **Frowning** or **downward mouth corners** suggest **sadness**.
    - By tracking these points, the model can better understand subtle facial changes related to emotional expressions.

CNN Architecture

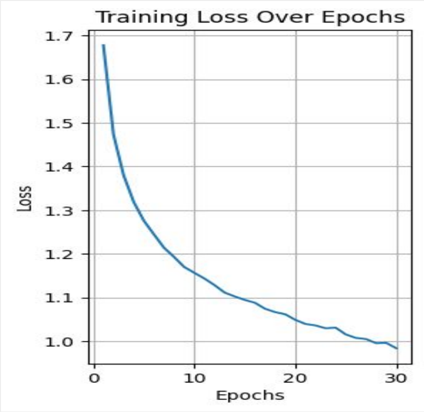
Model Overview

1. **Input Layer**:
   - 48x48 grayscale images.
2. **Convolutional Layers**:
   - **Conv1**: 32 filters, 3x3 kernel.
   - **Conv2**: 64 filters, 3x3 kernel.
   - **Conv3**: 128 filters, 3x3 kernel.
3. **Pooling Layers**:
   - MaxPooling (2x2) after each convolution to reduce dimensions.
4. **Fully Connected Layers**:
   - **FC1**: 256 neurons.
   - **FC2**: Output to 7 emotion classes.
5. **Dropout Layer**:
   - 50% dropout to prevent overfitting.
6. **Softmax Activation**:
   - Output probabilities for each emotion class.

Key Details:

- **Batch Size**: Adjustable (e.g., 32, 64).
- **Optimizer**: Adam or SGD with learning rate scheduling.
- **Loss Function**: Categorical Cross-Entropy.

Hyperparameter Tuning and Training

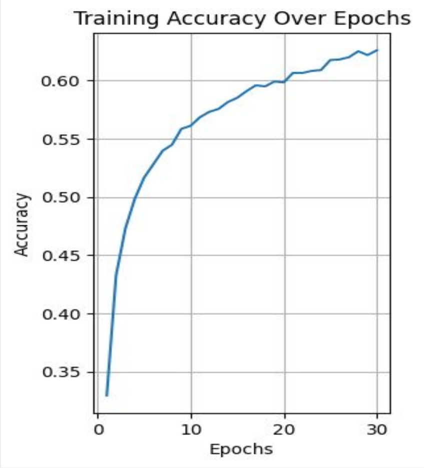


Hyperparameter Tuning

- **Learning Rate:** Experimented values (e.g., 0.001, 0.0001).
- **Batch Size:** Varying between 32, 64, 128.
- **Epochs:** Set to 30, with early stopping to avoid overfitting.

Training Process

- **Loss Function:** Minimized categorical cross-entropy.
- **Optimizer:** Used Adam with decay and momentum for stable convergence.
- **Early Stopping:** Stops training if the loss does not improve after a set number of epochs (patience), saving computation time and preventing overfitting.



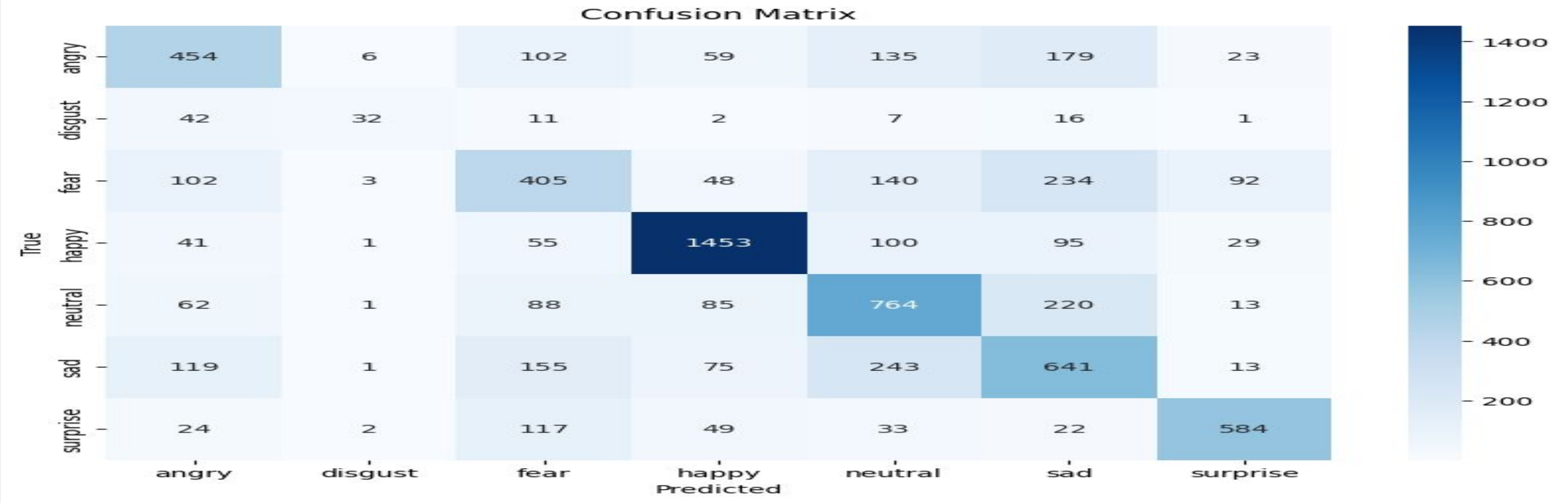
Training Loop Overview:

- Epochs: 30 total, adjusting the model for optimal performance.
- **Accuracy Tracking:** For each epoch, accuracy is calculated and printed alongside the loss.
- **Early Stopping:** Activated when the validation loss plateaus, stopping further training to save time and avoid overfitting.

Performance Metrics

Evaluation Metrics

- **Accuracy**: 0.6037 - Percentage of correctly classified images.
- **Precision**: 0.6082 - Positive predictive value for each emotion category.
- **Recall**: 0.6037 - Sensitivity, or true positive rate, for each emotion category.
- **F1-Score**: 0.6032 - Harmonic mean of precision and recall, providing a balanced performance measure.



User Interface (Streamlit App)

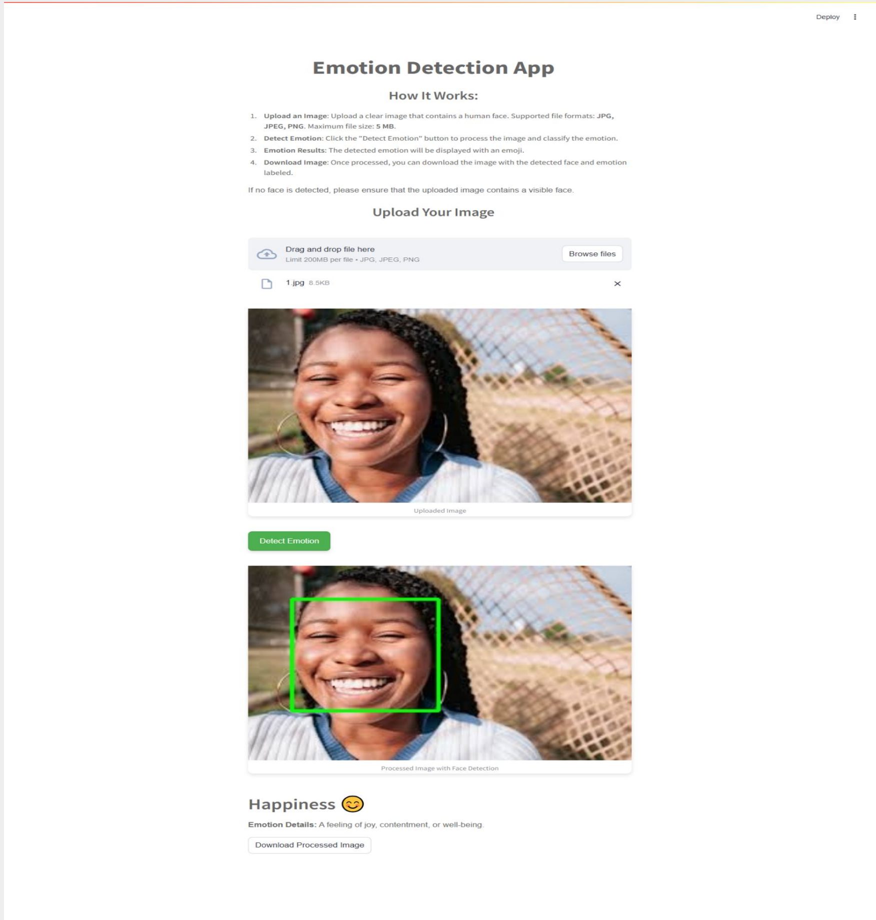
Streamlit Application

- **Image Upload**: Validates image file type (JPEG/PNG) and size to ensure only supported files are uploaded.
- **Real-time Results**: Provides immediate emotion detection feedback after the image is uploaded.
- **Responsive UI**: Designed to be intuitive and user-friendly, ensuring smooth interaction across devices.

App Features

- **Error Handling**: Displays an error message if an unsupported file type is uploaded.
- **Emotion Display**: Shows detected emotions with corresponding probability scores for transparency.

Screenshots of the Streamlit Interface



Challenges and Solutions

Challenges Faced

- **Imbalanced Dataset**: Some emotions had more samples, affecting model balance.
- **Handling Difficult Emotions**: Distinguishing between subtle emotions like neutral vs. sad.

Solutions

- **Data Augmentation**: Enhanced dataset diversity by applying transformations (e.g., flips, rotations).
- **Advanced Architectures**: Experimented with deeper models like **ResNet** and **VGG** to improve feature extraction.

Ethical Considerations

Privacy Concerns

- Emphasize **user consent** and **data protection** when dealing with facial data.

Bias in Emotion Detection

- Address **cultural bias**: Emotion expressions can vary across cultures and regions.

Mitigation Strategies

- **Bias Detection**: Regular checks to identify and reduce bias in predictions.
- **Diversified Dataset**: Including varied demographic data.
- **Transparency**: Clear communication on how predictions are made.

Future Work

Improvements

- **Video-Based Emotion Detection**: Expand from static images to real-time video (multiple frames).
- **Large-Scale Deployment**: Hosting on cloud services for scalability and availability.

<u>Conclusion</u>

Summary

- Built a **CNN-based system** for emotion detection from images.
- Achieved good performance with **Accuracy**, **Precision**, and **Recall** of **62%** across various emotions.
- **Real-World Uses**: Can be used in **healthcare** and **education** to better understand people's emotions.

.

"AI for Human Understanding"