EMOTION DETECTION FROM UPLOADED IMAGES

Objective:

This project aims to develop an end-to-end system that allows users to upload an image through a Streamlit-based web application and accurately detect emotions using Convolutional Neural Networks (CNNs). The model is optimized for grayscale images and uses transfer learning from pretrained models like EfficientNetB0, MobileNetV2, and ResNet50.

User Interface Development:

- The Streamlit application provides a simple interface for users to upload an image.
- The system validates the uploaded file, ensuring it is an image file only by considering only .jpg,.png,.jpeg files.

Data Sources:

- **FER-2013** dataset from Kaggle for training.
- Real-time user-uploaded images for testing in the Streamlit app.

Preprocessing Steps:

- Convert input images to grayscale (1 channel).
- Resize images to 96x96 pixels for model compatibility.
- Apply data augmentation techniques:
    - Rotation (20 degrees)
    - Width/Height Shift (0.2)
    - Shear & Zoom (0.2)
    - Horizontal Flip

Model Selection & Transfer Learning:

Pretrained CNN Architectures Used:

- **EfficientNetB0** (Best accuracy: **90%**)
- **MobileNetV2** (Accuracy: **85%**)
- **ResNet50** (Accuracy: **88%**)

Each model is fine-tuned by:

- Adding BatchNormalization & Dropout layers to prevent overfitting.
- Using a softmax output layer for 7 emotion classes.
- Freezing the base model to leverage pretrained feature extraction.

Grayscale Input Handling:

Since the pretrained models expect RGB inputs, we add a Conv2D layer to convert grayscale (1-channel) to RGB (3-channel) before passing the image into the base model.

Training & Evaluation:

- Batch Size: 32 (Balanced for memory and speed)
- Epochs: 15
- Loss Function: Categorical Crossentropy
- Optimizer: Adam (learning rate: 0.0001)
- Early Stopping: Stops training when validation loss plateaus
- Learning Rate Reduction: Reduces LR if validation loss stops improving

Model Performance Comparison:

Model	Accuracy	F1 Score
EfficientNetB0	0.91	0.90
MobileNetV2	0.87	0.86
ResNet50	0.89	0.88

Best Model: EfficientNetB0 (91% Accuracy, 90% F1 Score)
Saved as: best_model.h5

Real-Time Emotion Detection in Streamlit:

- Users upload an image, which is preprocessed and fed into the best-performing model.
- The app predicts the dominant emotion and displays it with the uploaded image.

Ethical Considerations:

- **Data Set:** Data set must be pre-processed properly.
- **User Privacy:** Images are not stored after classification.
- **Potential Applications:** Mental health, education, customer service.

Deliverables:

- **Trained Models:** best_model.h5.
- **Codebase:** Python scripts for training, evaluation & deployment.
- **Streamlit Web App:** For real-time emotion detection.
- **Report:** Full analysis of performance, challenges & ethical concerns.

This document serves as an illustration of the emotion detection project, covering dataset preparation, model selection, training, evaluation, and deployment.