**Facial expression recognition**

**Overview**

This Python script loads a pre-trained deep learning model (emotion\_model.h5) to predict emotions from grayscale facial images. It uses OpenCV for image processing, TensorFlow/Keras for model inference, and Matplotlib for visualization.

**Key Features:**

Loads a CNN model trained for facial emotion classification.

Reads an image in grayscale and resizes it to 48x48 pixels (matching model input).

Normalizes pixel values to range [0,1] for better model performance.

Processes the image into the correct format for the CNN model.

Predicts the emotion using the trained model and finds the most probable emotion.

Displays the image using Matplotlib with the predicted emotion and confidence score.

**How the Code Works**

**Load Dependencies**

OpenCV (cv2) for image processing

NumPy (numpy) for array operations

Matplotlib (plt) for visualization

TensorFlow/Keras (load\_model) for loading and running the trained CNN

**Load the Pre-Trained Model**

Checks if emotion\_model.h5 exists before loading.

The model classifies images into 7 emotions:

["Angry", "Disgust", "Fear", "Happy", "Neutral", "Sad", "Surprise"]

Detect Emotion from an Image

Step 1: Checks if the image file exists.

Step 2: Reads the image in grayscale using cv2.imread().

Step 3: Resizes the image to 48x48 pixels to match the model input.

Step 4: Normalizes pixel values to [0,1].

Step 5: Expands dimensions to fit the CNN input format.

Step 6: Predicts the emotion using model.predict().

Step 7: Finds the highest probability class and assigns the predicted emotion.

Step 8: Displays the image with the predicted emotion using matplotlib.pyplot.

Run an Example Image

Uses an example image (images/train/angry/1.jpg) for testing.

Prints the predicted emotion and confidence percentage.

**Expected Output**

The program prints the predicted emotion and confidence score in the console.

It also displays the image with the predicted emotion label.

**Output**

****