Import cv2

Import numpy as np

From keras.models import load\_model

Import matplotlib.pyplot as plt

# Load the trained model and Haar Cascade classifier

Model = load\_model(‘model.h5’)

faceDetect = cv2.CascadeClassifier(‘haarcascade\_frontalface\_default.xml’)

# Define label dictionary for facial expressions

Label\_dict = {0: ‘Angry’, 1: ‘Disgust’, 2: ‘Fear’, 3: ‘Happy’, 4: ‘Neutral’, 5: ‘Sad’, 6: ‘Surprise’}

# Load and process the input image

Input = cv2.imread(‘face4.jpg’)

Gray = cv2.cvtColor(input, cv2.COLOR\_BGR2GRAY)

Faces = faceDetect.detectMultiScale(gray, 1.3, 3)

For x, y, w, h in faces:

Sub\_face = gray[y:y + h, x:x + w]

# Resize and normalize the face for prediction

Resized = cv2.resize(sub\_face, (48, 48))

Normalize = resized / 255

Reshaped = np.reshape(normalize, (1, 48, 48, 1))

Result = model.predict(reshaped)

Label = np.argmax(result, axis=1)[0]

# Draw rectangles around the detected face

Cv2.rectangle(input, (x, y), (x + w, y + h), (0, 0, 255), 1)

Cv2.rectangle(input, (x, y), (x + w, y + h), (50, 50, 255), 2)

Cv2.rectangle(input, (x, y – 40), (x + w, y), (50, 50, 255), -1)

Cv2.putText(input, label\_dict[label], (x, y – 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.8, (255, 255, 255), 2)

# Generate Arousal Plot

Arousal\_level = np.random.uniform(0, 1) # Example arousal level; replace with actual if available

# Create subplots for displaying image, histogram, and arousal plot side by side

Fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(15, 5))

# Show the image with rectangles

Ax1.imshow(cv2.cvtColor(input, cv2.COLOR\_BGR2RGB))

Ax1.set\_title(“Detected Faces”)

Ax1.axis(‘off’)

# Plot Histogram of the detected face

Ax2.hist(sub\_face.ravel(), bins=256, range=[0, 256], color=’black’)

Ax2.set\_title(“Grayscale Histogram”)

Ax2.set\_xlabel(“Bins”)

Ax2.set\_ylabel(“# of Pixels”)

# Plot the Arousal level

Ax3.bar([“Arousal”], [arousal\_level], color=’blue’)

Ax3.set\_ylim(0, 1)

Ax3.set\_title(f”Emotion: {label\_dict[label]}”)

Ax3.set\_xlabel(“Arousal Level”)

Ax3.set\_ylabel(“Intensity”)

# Display all plots

Plt.tight\_layout()

Plt.show()

Cv2.destroyAllWindows()