Training Report – Day 23

Topic Covered Today:

- **Real-Time Emotion Detection** using webcam feed
- Integration of **OpenCV** for face detection
- Testing, output visualization, and project completion

Key Learning:

Today I successfully integrated the trained CNN model with **OpenCV** to detect faces from a webcam and classify emotions in real-time.

Implementation Steps:

1. Load the trained model (emotion_model.h5).
2. Use **OpenCV's Haar Cascade classifier** to detect faces from live camera input.
3. Extract the face region, preprocess it (grayscale, resize to 48×48), and feed it into the CNN model.
4. Display the predicted emotion on the video frame.

Sample Code:

```
import cv2
import numpy as np
from tensorflow.keras.models import load_model

model = load_model('emotion_model.h5')
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

emotion_labels = ['Angry', 'Happy', 'Sad', 'Surprise', 'Neutral']

cap = cv2.VideoCapture(0)

while True:
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)

    for (x, y, w, h) in faces:
        face = gray[y:y+h, x:x+w]
        face = cv2.resize(face, (48,48))
        face = face / 255.0
        face = np.expand_dims(face, axis=0)
        face = np.expand_dims(face, axis=-1)
```

```
        prediction = model.predict(face)
        emotion = emotion_labels[np.argmax(prediction)]

        cv2.rectangle(frame, (x,y), (x+w, y+h), (255,0,0), 2)
        cv2.putText(frame, emotion, (x, y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (255,255,255), 2)

    cv2.imshow('Facial Emotion Detection', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Output:

- The system successfully detected faces and displayed the corresponding emotion label in real-time.
- Tested on different expressions — results were accurate and responsive.

Activities / Assignments:

- Integrated the trained CNN model with OpenCV for live detection.
- Tested model performance on multiple users.
- Debugged camera frame and prediction lag issues.
- Prepared final documentation and project report.

Personal Reflection for Day 23:

Completing this project was a major milestone in my AI/ML training. It allowed me to combine multiple skills — **image processing, neural networks, and real-time system development**. Seeing the model correctly detect emotions through a webcam was highly satisfying.

This project taught me how machine learning models can be integrated into real applications. I also gained practical experience in deploying AI systems and understanding how theoretical concepts come alive in real-world scenarios.