Training Report – Day 22

Topic Covered Today:

- Building and training the **CNN model** for emotion detection
- Model evaluation and performance analysis
- Visualizing training accuracy and loss

Key Learning:

Model Architecture:

I designed a **Convolutional Neural Network (CNN)** to classify emotions from facial images.

Model Structure:

- Input layer: 48×48 grayscale image
- Convolution layers with ReLU activation
- Max Pooling layers to reduce dimensionality
- Flatten layer to convert 2D data into 1D
- Dense layers for learning complex features
- Output layer with **Softmax activation** for 5–7 emotion classes

Python Implementation:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(48,48,1)),
    MaxPooling2D(2,2),
    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D(2,2),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(7, activation='softmax')
])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Training the Model:

The model was trained using the FER2013 dataset for 25 epochs with 80% training and 20% validation split.

During training, the accuracy improved steadily while the loss decreased.

Visualization:

```
import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.legend()
plt.show()
```

Result Example:

- Training Accuracy: ~92%
- Validation Accuracy: ~88%

Activities / Assignments:

- Designed and trained CNN model for emotion classification.
- Visualized accuracy and loss curves.
- Evaluated model performance on test data.
- Saved trained model (emotion_model.h5) for deployment.

Personal Reflection for Day 22:

Today's session helped me strengthen my understanding of CNNs and how they can extract visual features automatically. Watching the model learn to detect facial patterns like smiles or frowns was very rewarding. It also taught me the importance of tuning hyperparameters for better performance.