**Emotion Detection Using CNN in PyTorch**

**Overview**

This project is an image classification system that detects human emotions from facial images using a Convolutional Neural Network (CNN) in PyTorch. The model is trained on grayscale 48x48 images and predicts one of seven emotions: **angry, disgust, fear, happy, neutral, sad, and surprise**.

**Features**

* Preprocessing using torchvision.transforms
* Data balancing using **SMOTE**
* CNN model with multiple convolutional and dropout layers
* Training and evaluation functions
* GUI for real-time emotion detection using **Tkinter**

**Requirements**

* Python 3.x
* PyTorch
* torchvision
* imbalanced-learn (SMOTE)
* NumPy, Pandas, Matplotlib
* PIL (Pillow)
* Tkinter

**Installation**

pip install torch torchvision imbalanced-learn numpy pandas matplotlib pillow

**Dataset**

* The dataset is organized into images/train (training) and images/validation (testing).
* Images are preprocessed (grayscale, resized to 48x48, normalized).
* The dataset distribution is visualized using pie charts.

**Model Architecture**

* **4 Convolutional Blocks** (Conv2D, BatchNorm, ReLU, MaxPooling, Dropout)
* **3 Fully Connected Layers**
* Softmax activation for classification

**Training the Model**

Run the script to train the model for **25 epochs**:

train\_acc, val\_acc = train\_and\_evaluate(CNN\_Model3(), epochs=25)

The trained model is saved as model3\_state.pth.

**Testing the Model**

The test accuracy is calculated after training:

**GUI for Emotion Detection**

* A Tkinter-based UI allows users to upload an image for emotion prediction.
* Click **Upload Image**, and the predicted emotion will be displayed.

**Output**

* Model accuracy plot
* Predicted emotion label in the GUI