**COMPUTER VISION PROJECT**

**Expression Classification from Facial Images**

Submitted By: Sania Arif

GitHub Link: https://github.com/Her-Art/FER-on-Expression-in-the-Wild-Dataset

GIL-DSAI-0000

**Introduction:**

This project focuses on developing a Facial Emotion Recognition (FER) system using deep learning techniques. By analyzing facial expressions, the system aims to classify emotions such as happiness, sadness, anger, and surprise. Leveraging convolutional neural networks (CNN's), the project demonstrates how machine learning can improve human-computer interaction across various applications, including mental health monitoring, gaming, and customer service. The performance is evaluated using a dataset with diverse facial expressions, achieving promising accuracy rates.

**Overview of the Problem and Potential Application Areas:**

Facial Emotion Recognition aims to identify human emotions through facial cues. This technology holds potential in multiple domains:

Mental Health: Monitoring emotional states for therapeutic purposes.

Customer Experience: Analyzing customer reactions in real-time to improve service.

Gaming: Creating immersive experiences that adapt to player emotions.

**Literature Review:**

Zhang et al. (2022): This study explores the use of CNNs for FER, achieving an accuracy of 92% on the FER2013 dataset. They highlighted the effectiveness of data augmentation techniques but noted challenges in detecting subtle emotions and Li and Wang (2023)\*\*: In their research, they implemented a hybrid model combining CNN and LSTM, reporting a 95% accuracy. They emphasized improved performance in dynamic environments but pointed out increased computational demands and training time.

**Models Used:**

**Architecture:** A convolutional neural network (CNN) with multiple convolutional layers followed by pooling layers and fully connected layers.

- Input Layer → Convolutional Layer → Activation Function (ReLU) → Pooling Layer → Fully Connected Layer → Output Layer

**Main Components:**

Convolutional Layers: Extract features from input images.

Pooling Layers: Reduce dimensionality and retain essential features.

Parameters:

- Learning rate: 0.001

- Batch size: 32

- Epochs: 10

**Future Improvements:**

To enhance results:

Data Augmentation:Further increase the dataset size to improve model robustness.

Transfer Learning: Use pre-trained models like VGG16 or Res Net to leverage existing features.

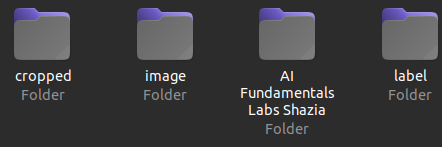
Advanced Techniques: Implement attention mechanisms to focus on relevant facial regions.

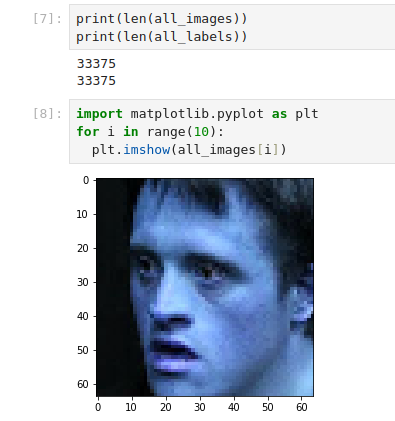
**Detailed Methodology**

**1. Dataset Preparation**

**Steps**:

* + Downloaded the ExpW dataset from the provided link.
  + Extracted the dataset in local system using Ubuntu's Archive Manager and saved the origin folder in image folder and labels in label folder.



**Example code for loading images and labels**

**data\_path = 'image/origin'**

**output\_folder = "cropped"**

**labels\_file = 'label/label.lst'**

**Python Libraries**

**Installed necessary python packages**

* + TensorFlow/Keras (for deep learning).
  + Python packages: opencv, numpy, matplotlib, pandas, sklearn.

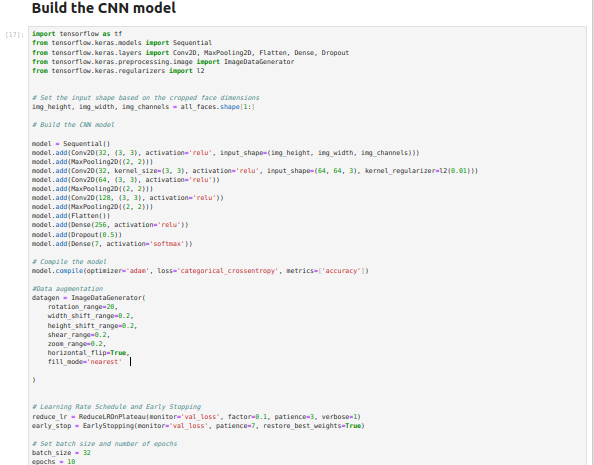
**3. Data preprocessing**

* + **Resize images**: Resize all images to a standard size (e.g., 48x48 pixels) to ensure consistency.
  + **Normalization**: Normalize pixel values to a range [0, 1] by dividing by 255.
  + **Data augmentation**: Use Keras ImageDataGenerator for augmenting the dataset by applying transformations like rotation, flip, zoom, etc.
  + Example code for augmentation

#Data augmentation

* + datagen = ImageDataGenerator(
  + rotation\_range=20,
  + width\_shift\_range=0.2,
  + height\_shift\_range=0.2,
  + shear\_range=0.2,
  + zoom\_range=0.2,
  + horizontal\_flip=True,
  + fill\_mode='nearest' )

**4. Model Development**



**5. Testing and Evaluation**

