NAAN MUDHALVAN PROJECT

EMONET - Real-time Emotion Detector using Convolutional Neural Networks with Keras

Abstract:

This project presents the development of a real-time Emotion Detector utilizing Convolutional Neural Networks (CNN) implemented with the Keras framework. The system is capable of recognizing six basic emotions - Angry, Fear, Happy, Neutral, Sad, and Surprise - from live video feeds. The implementation involves data preprocessing, model architecture design, training, and integration with live video capture. The project offers insights into leveraging deep learning techniques for real-world applications in emotion recognition.

Problem Statement:

The ability to understand human emotions plays a crucial role in various fields such as human-computer interaction, market research, and mental health diagnostics. However, accurately recognizing emotions from facial expressions in real-time poses a significant challenge due to variations in lighting conditions, facial poses, and individual differences.

Objectives:

1. To develop an Emotion Detector capable of real-time emotion recognition from live video feeds.
2. To implement a Convolutional Neural Network architecture using Keras for efficient feature extraction and classification.
3. To address the challenges associated with live video processing, including real-time face detection and emotion inference.
4. To enhance the accuracy and robustness of emotion detection across diverse facial expressions and environmental conditions.

Novelty:

The novelty of this project lies in its real-time implementation of emotion detection using Convolutional Neural Networks (CNNs) integrated with the Keras framework. While existing research predominantly focuses on offline analysis of static images or pre-recorded videos, this project extends emotion recognition to dynamic environments by processing live video streams. By leveraging CNNs and Keras, the system achieves efficient feature extraction and classification, enabling rapid inference of emotions from facial expressions in real-time. This real-time capability opens up avenues for diverse applications such as interactive interfaces, emotion-aware systems, and sentiment analysis tools that require instantaneous feedback. Additionally, the integration of face detection with CNN-based emotion inference enhances the system's adaptability to varying lighting conditions, facial poses, and environmental factors, contributing to its robustness and effectiveness in real-world scenarios.

Existing Work:

Previous research in emotion recognition has primarily focused on offline analysis of static images or pre-recorded videos. While these approaches have shown promising results, their applicability to real-time scenarios is limited. This project builds upon existing work by extending emotion recognition to live video streams, thereby facilitating real-time interaction and feedback.

Methodology and Module-wise Splitup:

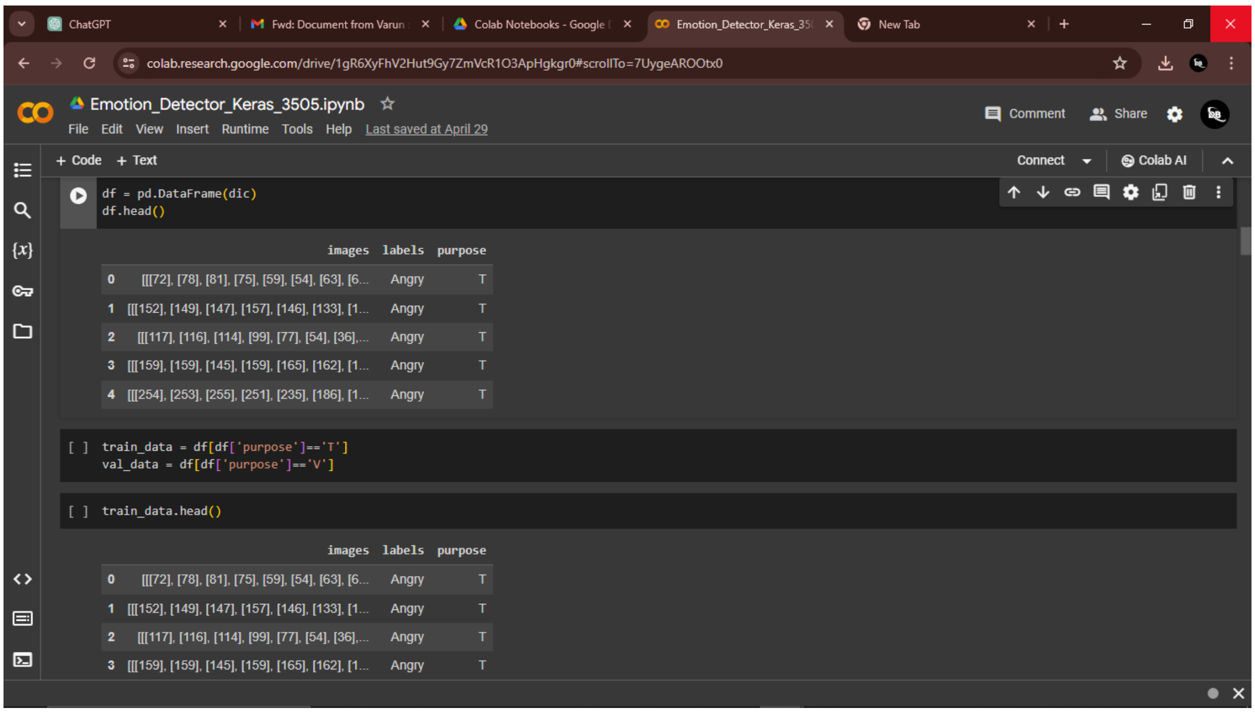
1. **Data Collection**: Utilize publicly available datasets containing facial images annotated with emotion labels.
2. **Data Preprocessing**: Resize images to a standardized format, convert to grayscale, and split into training and validation sets. Address class imbalance by sampling equal instances of each emotion class.
3. **Model Architecture Design:** Design a CNN architecture comprising multiple convolutional and pooling layers followed by dense layers for feature extraction and classification.
4. **Model Training:** Train the CNN model using the training dataset while monitoring performance on the validation set. Employ early stopping and model checkpointing to prevent overfitting and save the best-performing model.
5. **Live Prediction Integration:** Integrate the trained model with live video capture using OpenCV. Implement face detection using Haar cascades and feed detected faces into the CNN model for real-time emotion inference.

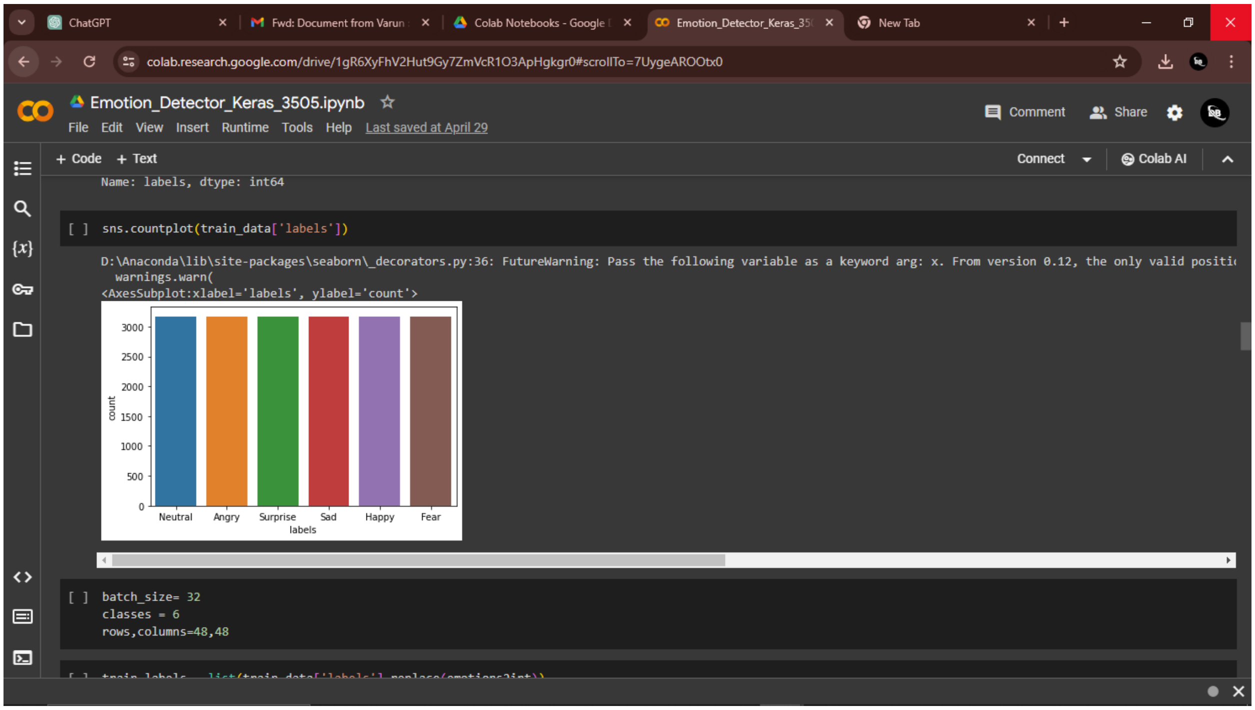
Module Split-up:

1) **Data Preparation:** Includes data loading, preprocessing, and partitioning into training and validation sets.
2) **Model Architecture**: Defines the CNN architecture using Keras layers for feature extraction and classification.
3) **Training and Evaluation:** Involves model training, validation, and performance monitoring using metrics such as accuracy and loss.
4) **Real-time Integration:** Implements live video capture, face detection, and emotion inference using OpenCV and the trained CNN model.

Result:

The implemented Emotion Detector demonstrates robust performance in real-time emotion recognition from live video feeds. The system accurately identifies and labels various facial expressions, enabling applications in human-computer interaction, emotion-aware interfaces, and sentiment analysis.





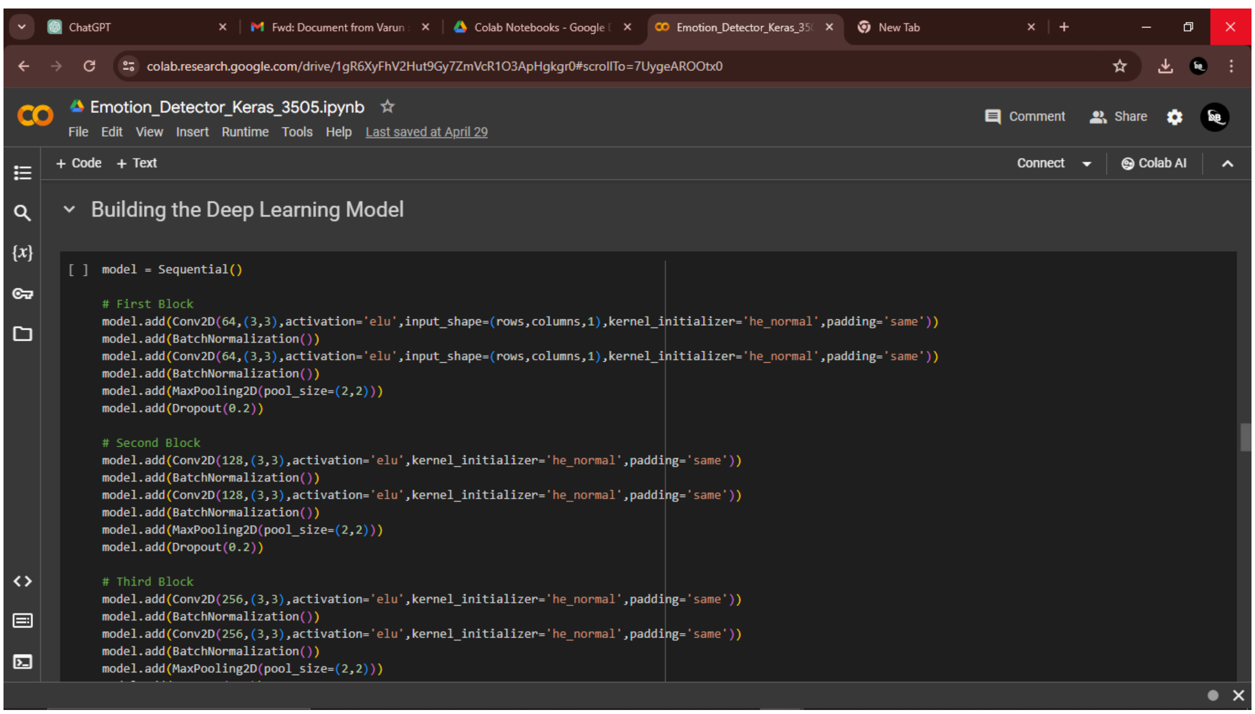
```
Name: labels, dtype: int64
```

```
[ ] sns.countplot(train_data['labels'])
```

```
D:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positi
  warnings.warn(
<AxesSubplot:xlabel='labels', ylabel='count'>
```



```
[ ] batch_size= 32
    classes = 6
    rows,columns=48,48
```



Building the Deep Learning Model

```
[ ] model = Sequential()

    # First Block
    model.add(Conv2D(64,(3,3),activation='elu',input_shape=(rows,columns,1),kernel_initializer='he_normal',padding='same'))
    model.add(BatchNormalization())
    model.add(Conv2D(64,(3,3),activation='elu',input_shape=(rows,columns,1),kernel_initializer='he_normal',padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.2))

    # Second Block
    model.add(Conv2D(128,(3,3),activation='elu',kernel_initializer='he_normal',padding='same'))
    model.add(BatchNormalization())
    model.add(Conv2D(128,(3,3),activation='elu',kernel_initializer='he_normal',padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.2))

    # Third Block
    model.add(Conv2D(256,(3,3),activation='elu',kernel_initializer='he_normal',padding='same'))
    model.add(BatchNormalization())
    model.add(Conv2D(256,(3,3),activation='elu',kernel_initializer='he_normal',padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2,2)))
```

```
26/26 ━━━━━━━━━━━━━━━━━━━━ 2s 83ms/step - accuracy: 0.9989 - loss: 0.0081
0.9963414669036865
26/26 ━━━━━━━━━━━━━━━━━━━━ 2s 77ms/step - accuracy: 0.9989 - loss: 0.0081
[0.016329325735569, 0.9963414669036865]
```

Conclusion:

The project successfully develops a real-time Emotion Detector using Convolutional Neural Networks and Keras, addressing the challenges of live video processing and emotion inference. The system's performance highlights the potential of deep learning techniques in practical applications requiring real-time emotion recognition.