Facial Emotion Recognition Using Convolutional Neural Networks

Deep Learning Project

INTRODUCTION

- Emotion detection refers to the process of identifying human emotions from various inputs, such as images, speech, or text. The main focus of this project is emotion detection from facial images using a Convolutional Neural Network (CNN). This model can predict the emotional state of a person in real-time using images and webcam input.
- We use OpenCV for real-time face detection and TensorFlow for training a CNN model on an emotion dataset. The model is capable of detecting several emotions like happy, sad, angry, surprised, fearful, disgusted, and neutral.

OBJECTIVE

- Train a Convolutional Neural Network (CNN) on an emotion dataset.
- Use the trained model to predict emotions from facial images.
- Implement real-time emotion detection using a webcam with the help of OpenCV.
- Improve the prediction accuracy by fine-tuning the model and using data augmentation techniques.

TOOLS & TECHNOLOGIES

- Programming Language: Python
- Libraries Used:
  - TensorFlow: For building and training the CNN model.
  - Keras: For handling the deep learning model's layers and training.
  - OpenCV: For face detection and webcam integration.
  - NumPy: For numerical operations.
  - Pandas: For managing datasets.
  - Matplotlib/Seaborn: For visualizing training and evaluation metrics.
- Dataset:
  - Kaggle’s "FER2013" dataset: Contains images of faces labeled with emotions like happy, sad, surprised, angry, etc

METHODOLOGY

Data Collection

The dataset used for this project is the FER2013 dataset, which consists of 35,887 grayscale images, each 48x48 pixels, labeled with one of seven emotions: happy, sad, surprise, anger, fear, disgust, and neutral.

The images are organized into training and testing sets, with corresponding emotion labels.

We used ImageDataGenerator from TensorFlow to load and augment these images.

Data Preprocessing

- Image Resizing: Images are resized to 64x64 pixels to match the input shape of the CNN.
- Normalization: Pixel values are normalized by dividing by 255.0 to scale the input data to a range of 0-1.
- Data Augmentation: Techniques like rotation, zoom, width & height shift, and horizontal flip were applied to improve the model's robustness.

CNN Model Architecture

The CNN model consists of:

- Convolutional Layers (Conv2D): These layers apply filters to the input images to extract key features.
- MaxPooling Layers: Reduce the spatial dimensions and help prevent overfitting.
- Flatten Layer: Converts the 2D matrix of features into a 1D vector.
- Dense Layers: Fully connected layers for classification, including a softmax output layer to predict the emotion classes.

Training the Model

The model was compiled with the Adam optimizer and categorical cross-entropy loss function, and trained for 25 epochs . Training was performed on the train generator using the ImageDataGenerator object, which loads images in batches.

Real-Time Emotion Detection Using OpenCV

The OpenCV library was used to capture video from the webcam, detect faces in each frame, and classify the emotion on the detected face using the trained CNN model. The face detection is achieved using Haar Cascade Classifiers in OpenCV. The real-time webcam detection works by:

- Detecting faces in each video frame.
- Extracting the face region, resizing it, and feeding it to the trained CNN model.
- Displaying the predicted emotion on the face in the video.

CONCLUSION

This project successfully demonstrated emotion detection using a CNN on facial images. By combining OpenCV with the trained CNN, we were able to implement real-time emotion detection using a webcam. The model showed promising results, achieving high accuracy in predicting emotions on test images and live video. However, future work can focus on improving accuracy by using a larger dataset, fine-tuning the model, and exploring advanced techniques like transfer learning.