Jagadeesh V
Guvi / AIML-WE-E-B2
jagdeeshv@hotmail.com

Face Emotion Detection

Abstract

Technological developments have enabled computers to identify and categorize facial expressions to determine a person's emotional state in an image or a video. This process, called "Facial Expression Recognition (FER)", has become one of the most popular research areas in computer vision. FER has been recognized for a decade, and it is a vital topic in the fields of computer vision and machine learning.

Automatic FER is useful in most of the applications such as healthcare, teaching, criminal investigation, Human Robot Interface (HRI), etc.

This project introduces a machine learning-based solution to detect and classify the emotion present in the image using Convolutional Neural Networks (CNNs) through a Streamlit, web-based application. Detection and classifying emotions in facial video captured through web cam is attempted that depicts a good result.

The dataset used is Kaggle's FER2013 that contains approx. 30000 facial RGB images of different expressions with size restricted to 48×48, and the main labels of it can be divided into 7 types: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral. By implementing image preprocessing techniques, the model's performance was enhanced.

The image / video frame is detected for face, features extracted and emotion classified with the pre-trained model.

Jagadeesh V
Guvi / AIML-WE-E-B2
jagdeeshv@hotmail.com

Introduction

Facial emotion recognition has become an important issue in many applications nowadays. The aim of facial emotion recognition is to help identify the state of human emotion based on facial images. Various communication elements, such as facial expressions, body movements, and voice, can be employed to identify human emotions. Among these, facial expressions are particularly informative, as they convey information about a person's emotional state

The challenge on facial emotion recognition is to automatically recognize facial emotion state with high accuracy. Therefore, it is challenging to find the similarity of the same emotion state between different person since they may express the same emotion state in various ways such as the individual's mood, their skin color, age, and environment surrounds.

OBJECTIVE

The primary objective of this project is to develop a simple, practical, user-friendly tool for a real-time facial emotion detection and classification. Using CNN techniques and an effective pre-processed and trained dataset model, this project is focused to build a reliable detection system.

The end goal is a fully functional web application, built with Streamlit, that allows users to upload images and receive immediate feedback on face emotion.

Supplementarily, functional application that allows users to display face through webcam and receive immediate feedback on face emotion.

Jagadeesh V
Guvi / AIML-WE-E-B2
jagdeeshv@hotmail.com

Software Requirement

IDLE Python, Jupyter LAB, installed with required libraries. Libraries include

1. os
2. cv2
3. numpy
4. keras
5. tensorflow
6. streamlit
7. Matplotlib
8. Pandas
9. Seaborn
10. `scikit-learn`
11. `imbalanced-learn`
12. Scipy

Realtime Applications

1. The system can be used in mini-marts, shopping centres to view the feedback of the customers to enhance their business,

2. The system can be installed at busy places like bus stand, airport, railway station, etc for detecting facial expressions of commuters. If there are any suspicious faces / odd emotion the system might setan internal alarm.

3. The system can be used at educational institutions to get feedback on students reaction at the class.

4. This system can be used for detecting lie amongst criminal suspects during interrogation

5. Clever marketing is feasible using emotional knowledge of a person which can beidentified by this system. Dataset and preparing pre trained model

1. Dataset

To recognise facial expression of an image, we should first have pre-processed, well trained ample models for each expression that will form library to map the emotion of the input image. Thus, the preliminary step is the dataset model.

Kaggle Fer2013 consists of 48x48 pixel grayscale images of different expressions faces, and the main labels of it can be divided into 7 types: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral. The Disgust expression has the minimal number of images – 547, Happiness with 8989 while other labels spreads evenly between 4000 to 6000 each.

The faces have been automatically registered so that the face is centred and occupies about the same amount of space in each image.

2. Pre-processing, Training and saving Model

Source code: Jag_Face_Expression_Model_creation.ipynb executed in jupyter lab

Various metrics were used to ascertain the strength and augment the model images.

Data set was analysed to with various metrics to evaluate its strength and few rectifications were made as follows:

1. Emotion class distribution was visualised by counts per emotion and plotting bar chart. Since *the emotions data is imbalanced with disgust emotion having only 1.5% of total images, Thus, we had to balance the data by oversampling [*RandomOverSampler*] technique to get even counts of 8989 for each emotion after splitting pixels aas X data and emotions as Y data.*

2. *The balanced data was pre-processed pixels by Flattening, normalizing, reshaping.*

3. *Out of total 62923 images divided 10% (6293) as test data and trained with remaining number of images. Did* One hot encoding on Train & test data and model built using conv2d 5 layers with activation RELU & final activation with softmax.

4. Model was complied with Adam optimizer and fit with 35 Epochs that took approx.. 6 ½ hours to complete. Model was evaluated for accuracy & loss metrics.

5. The test accuracy & loss of our model is 81.31% & 0.605 respectively, which is better than many exsited state-of-the-art results. Plotted these parameters as line graph for each Epoch.

6. Classification report and Confusion matrix were generated for the model with respect test data predicted vs actuals.

7. Finally saved the model's json and weights in files "Facial Expression Recognition.json" & "fer.weights.h5" for later use of predicting emotion of provided image / video faces.

Realtime Face emotion prediction

Two user interfaces are designed 1. For detecting emotion of uploaded images using streamlit and 2. For capturing face video through web cam and predicting emotion. This is an attempt made to capture faces with webcam video and predict emotions in real time that has fairly gives good result.

1. Face emotion detection of image

Source code: Jag_Detect_Emotion.py [Streamlit app executed in command prompt]

The user interface is designed with simplicity and ease of use in mind, making it accessible to anyone in need of quick and reliable face emotion detection. The interface is built using **Streamlit**, which allows for rapid development and deployment of interactive web applications interface

Image Upload Feature: The core functionality of the UI is the image upload option, where users can select and upload a human face image. This feature includes drag-and-drop functionality for ease of use, file type validation (common image formats, including JPEG and PNG) and image validation (Throws message if not face).

Real-time Display of Predictions: Once an image is uploaded, the interface displays the model's prediction along with a confidence score, helping users understand the reliability of the results. Emotion predictions are displayed in the image along with a rectangle covering the face part.

Detection is accomplished through three major stages as

Face Detection → Feature Extraction → Emotion Classification

1 & 2. Face Detection & Feature extraction

It is a pre-processing phase to recognize facial expressions of human. The image is converted to gray scale using cvtColor of cv2. An image is segmented into two parts which have faces and other non-face regions.

Feature extraction converts a pixel data of the face region into a higher-level representation of shape, colour, texture, and spatial configuration of the face or its components. Feature extraction will reduce the dimension of the input space while keeping the important

There are numerous methods used for face detection.

OpenCV provides several pre-trained models for detecting faces, eyes, smiles, etc. In this project, Haar classifier is used as accuracy is high for face detection due to suitable Haar features and computational complexity is very less due to a set of features which contribute the maximum, for the face detection problem in a training phase

These models are stored in XML file (haarcascade_frontalface_default.xml) and is loaded through CascadeClassifier. Various objects in the image are detected by detectMultiScale using the loaded pre trained model xml.

This function returns a list of rectangles where the target object is found. The function employs the Haar Cascade classifier to detect objects at different scales and locations within the image information. Feature extraction is important in formulating a better emotion categorization as the extracted facial feature given inputs to classification module which finally it categorizes different emotions.

Thus, grey scaling, Cascade Classifier & Detect Multi Scale are used to detect face & feature extraction. If the result is negative, the app displays a message that 'It is not a face'.

3. Expression Classification This stage is performed by a classifier.

If face is successfully detected, the image is resized, normalized and face region is cropped using cv2.resize, cv2.normalize, / 255.0. Then the emotion is predicted using the pre trained model json and weights. A rectangle is drawn bordering the face region of the full image, emotion text is printed with cv2.FONT_HERSHEY_SIMPLEX and the result is displayed.

This is repeated until key Q is pressed and quits on recognizing Q key press.

2. Face detection using webcam video

Source code: Jag_Webcam_test.py executed in IDLE python

This module is executed through IDLE as the feature is yet to be available in jupyter lab and Streamlit.

Web cam video is captured with cv2.VideoCapture(0).read() and converted to gray scale using cvtColor of cv2. OpenCV provides several pre-trained models for detecting faces, eyes, smiles, etc. These models are stored in XML file (haarcascade_frontalface_default.xml). This is loaded through CascadeClassifier. Various objects in the image are detected by detectMultiScale using the loaded pre trained model xml.

If face is successfully detected, the image is resized, normalized and cropped using cv2.resize, cv2.normalize, / 255.0. Then the emotion is predicted using the pre trained model json and weights. A rectangle is drawn bordering the face features of the image, emotion text is printed with cv2.FONT_HERSHEY_SIMPLEX and the result is shown.

This is repeated until key Q is pressed and quits on recognizing Q key press.

Jagadeesh V
Guvi / AIML-WE-E-B2
jagdeeshv@hotmail.com

CONCLUSION

This project proposes an approach for recognizing the category of facial expressions Face Detection and Extraction of expressions from facial images is useful in many applications, such as robotics vision, video surveillance, digital cameras, security and human-computer interaction.

In this project, seven different facial expressions of different persons' images from FER2013 dataset was analyzed. This project involves facial expression preprocessing of captured facial images followed by feature extraction using feature extraction and classification of facial expressions based on pre-trained datasets of facial images.

To measure the performance of proposed algorithm and methods and check the results accuracy, the system has been evaluated using various metrics. The same dataset was used for both training and testing by dividing the datasets into training samples and testing samples

.

Experiment results on FER2013 dataset, show that our proposed method can achieve a good performance. More efforts should be put to improve the classification performance for important applications.