EMOTION BASED SONG SUGGESTION USING DEEP LEARNING

[1]RahulRaja R, [2]Rithika B, [3]Naveen Kumar K R ,[4]Dr.P.SathishKumar

[1.]Computer Science Engineering, Bannari Amman Institute of Technology
[2.]Computer Science Engineering, Bannari Amman Institute of Technology
[3].Computer Science Engineering, Bannari Amman Institute of Technology
[4].Assistant Professor, Computer Science Engineering, Bannari Amman Institute of Technology

Abstract

Online Music Players are becoming increasingly popular, yet the suggestions offered by an online player are meaningless. So, in order to improve the app environment and improve the user experience, we propose to generate music suggestions based on the user's facial expression using Machine Learning. In this project, we are using OpenCV for detection and CNN for classifying Emotion Detection; the output from the trained model will be supplied for song suggestor, and the song recommended will be based on user facial reaction.

Keywords *– Cascade Classifier, CNN, Fer2013, OpenCV.*

I. Introduction

Songs have the ability to connect with the user if they reflect the user's feelings. This will allow the online player to connect with the user in a more efficient manner than competitors. Face Reaction has a greater influence on one's feeling; if we employ this face reaction detection in song choices, we may help the music player interact with the user better. As a result, we intended to train the emotion recognition model using CNN. In this project, we planned to identify the user's emotion utilizing the user's front camera, OpenCV, and a cascade classifier for human recognition from the user picture, which was then fed into the trained model. The model that CNN has trained. The model's output will be used to make song recommendations to the music player. It is simple to install into any current online music service.

II Modules

We structured it into segments to improve perception. We have three key modules: Human Face Detection, Emotion Recognition, and Song Suggestion. Human face detection is the initial module, which receives input from the user's front camera, which is now standard in all smartphones and computers. This module employs the OpenCV cascade classifier to detect the human face in the picture and then crop the image to contain only the face for easier calculation and identification in the following module. The second module is the emotion recognition module, which includes a CNN-trained model using the Fer2013 dataset from Kaggle. The first model's output is used as an input for this module. The next module is the recommendation module, which recommends music based on the results of the emotion detection module. In a music player like Spotify it searches the songs based on the user's emotions and displays the available song

III Dataset collection and pre-processing

The dataset used for training the CNN model is Fer2013. This Dataset is made up of 5,876 labelled images contributed by 123 individuals. This dataset has 7 labelled emotions like Happy, Sad, Angry, Fear, Surprise, Disgust, Neutral. This is utilized in the data collecting processes in both real-world and online media to capture as much data as possible. The real world has many forms of emotional photographs of friends and family members, relatives, and some recognized unknown people's various facial expressions. The data they gathered was initially saved for future research. Data set from kaggle.com is gathered from internet media. This collection of data was published to this website 6 years ago. This website's most reliable data set of emotions.

This resulted in 4848-pixel grayscale photos of faces. It is divided into two sections: pixels and sentiments. The mood part has a number code ranging from 0 to 6. Furthermore, the pixel section has a string embedded in statements for each image. Furthermore, the image should only be a portrait of a face. As a result, the gathered images are scaled and cropped portraits of people's faces.

This dataset has three columns: mood, pixels, and use. The emotion column contains integer encoded emotions, the pixel's column contains pixels in the form of a string separated by spaces, and the use column indicates whether the data was created for training or testing. As the data is not in the proper format. We must first pre-process the data. Pixels are contained in X train, X test, while emotions are contained in Y test, Y train. At this point, X train and X test have pixel numbers in the form of strings; changing them to integers is simple; we simply typecast them. We need to connect the 1D integer encoded labels in y test and y train to categorical attributes for effective training. For training purposes, you must transform the data into a 4d tensor (row num, width, height, channel). At this point, we have successfully pre-processed our data into X_train, X_test, y_train, and y_test.

IV Data visualization

The dataset can be visualized in form of table which contains 3 column namely emotion, pixels and usage.After the detection of emotion from image we planned to use a bar graph which will have labels of seven emotions. For plotting the graph we've used the matplotlib default libraries in python to display the detected emotion based amout of emotions detected.

V Selecting features and scaling

The Viola-Jones algorithm is the most used algorithm for feature detection in The features are detected from the grayscale image where the features like eyebrows, nose and teeth are identified by using edge feature, line feature and edge feature respectively. After selecting these Haar feature it goes inside the algorithm for detecting emotion of the user.

VI Fit the data in model

Image data augmentation is used to enhance the model's performance and generalization capabilities. It's usually a good idea to conduct some data augmentation before handing it to the model, which you can accomplish with ImageDataGenetrator provided by Keras. Rescale-This function normalizes pixel values by dividing them by 255. Rotation_range-The picture is rotated by 0–90 degrees. We will only use rescaling on testing data (normalization).

VII Algorithm Implementation

Sequential will be the model type we'll be implementing. In Keras, the simplest technique to create a model is sequential. It enables you to construct a model layer by layer. To add layers to our model, we utilize the 'add ()' method. Our first two layers are Conv2D. These are the convolution layers that will process our input photos, which are represented as 2-dimensional matrices. The number of nodes in each layer is 64 in the first layer and 32 in the second layer. Depending on the size of the dataset, this value can be modified to be greater or lower. In our instance, 64 and 32 work nicely, so we'll remain with them for the time being. Kernel size is the size of our convolution's filter matrix. With a kernel size of 3, we'll get a 3x3 filter matrix. The ReLU, or Rectified Linear Activation, will be the activation function for our first two layers. This activation function has been demonstrated to be effective in neural networks. Our first layer, too, accepts an input shape. As stated before, this is the form of each input picture, 28,28,1, with the 1 indicating that the photos are grayscale. There is a 'Flatten' layer between the Conv2D layers and the dense layer. Flatten acts as a bridge between the convolution and thick layers. The model will then forecast which choice has the highest likelihood. The model must then be compiled. The model is built using three parameters: optimizer, loss, and metrics. The learning rate is controlled by the optimizer. It will make use of the optimizer 'Adam'. In most circumstances, Adam is a decent optimizer to utilize. Throughout the training, the Adam optimizer changes the learning rate. The learning rate controls how quickly the model's optimum weights are computed. A slower learning rate may result in more accurate weights (up to a point), but it will take longer to compute the weights. For our loss function, we'll employ 'categorical cross-entropy.' This is the most popular categorization method. The model is doing better if the score is lower. To make things even easier to understand, while training the model, the 'accuracy' measure will be used to show the accuracy score on the validation set.

It will train their model using the 'fit ()' function with the following parameters: training data (train X), target data (train y), validation data, and the number of epochs. It will utilize the test set supplied in its dataset for validation data, which has been divided into X test and y test. The number of epochs specifies how many times the model will cycle over the data. The model will improve over time as we run additional epochs, up to a point. After that, the model will cease to improve with each epoch. We'll use 20 epochs for our model. On the validation set, it has achieved 93 percent accuracy after 20 epochs.

Coding:

We have some foreign necessary libraries for our projects like numpy, cv2, Os, webbrowser, matplotlib. Initially, cv.VideoCapture() for capturing the image from the user's front camera and it's converted to grayscale for further processing. The image is then face mapped using cv2.rectangle. The image is then processed and then converted into an array using numpy, which is then provided to the model for emotion detection. This will return an array of emotions found on the image in the descending order of percentages, which can then be plotted into a bar graph using matplotlib that can be used for reference. From the array the emotion which is greater is determined as the user emotion. The detected emotion is sent to spotipy library, which in turn searches for the playlist that matches with the detected emotion. The songs from the playlist will be suggested to the user.

VIII Results

The first major problem was the limited amount of material available for developing a wide framework. Which must be defeated for the sake of nature's structure. The most common answer to this is move learning. This technique began with a pre-planned strategy and calibrated this model using put-away information acquired from the real world. A series of preliminary studies confirmed the hypothesis that facial recognition would be more useful in online music player. There are models that make good use of such systems. So, if this receives enough data and a high-performance computer, it will be easy to increase the accuracy to more than 90 percent. It will also be able to employ that method on a different platform to achieve a different result and assist in making online music players more convenient for the users

IX ADVANTAGE AND DISADVANTAGE:

Advantages:

In this project we are using no additional equipment other than a normal front camera which is now available in all existing electronic gadgets i,e. laptop, mobile. It can be easily implemented to already existing online music players (i,e. spotify,etc.) for which the demo has been shown below. It also increases the user experience by more relevant suggestions

Disadvantage:

The accuracy of the project is not 100% yet.

X Conclusion

To better follow the framework and discover emotion expression patterns, we employed a deep learning CNN algorithm in conjunction with Keras, Tensorflow, and retraining ideas. It was feasible to identify emotions, as well as the kind of emotion, in real-world images using these approaches. To outline the results and methods more visibly, decision tree approaches have been established, which aid in determining which emotions % is high and which emotions percentage is low. Now, a large majority of feelings receive the most realistic emotions conceivable. And those with a low percentage of emotions have a worse likelihood of survival. It is now feasible to detect correct emotions as a result of this research. And robots can more correctly discern emotion and, as a result, provide a suitable reaction while also helping to prevent the same unintended occurrence. This machine may also be used to replace humans.

Reference

[1]. Chu Wang, Jiabei Zeng, Shiguang Shan, Xilin Chen, "*Multi-task learning of emotion recognition and facial action unit detection with adaptively weights sharing network*" in IEEE, 2019

[2]. Ninad Mehendale, "*Facial emotion recognition using convolutional neural networks (FERC)*" 18 February 2020

[3]. James Pao, "*Emotion Detection Through Facial Feature Recognition*" in International Journal of Multimedia and Ubiquitous Engineering, November 2017