JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Face Emotion-based Music Player

Ms. Gayatri Enugurti, Ms. Sakshi Gummalwar, Ms. Vaishnavi Pandey, Ms. Yogeshwari Ghotekar, Asst. Prof. Manisha Pise

Computer Science and Engineering Department, Rajiv Gandhi College of Engineering, Research and Technology, India

***Abstract*—Music has numerous advantages, not just for our physical health but also for our emotional well-being. Music can stimulate our bodies, relieve pain, and soothe our brains. It has a big impact on our actions, feelings, and thoughts. Music has a profound impact on our psychological well-being. The face is an important organ in one's body, which plays a crucial part in determining one's behavior as well as emotional state. We will combine these two aspects in our project, which will aid in detecting an individual's emotion through facial expression and playing music in response to the mood detected, which will alleviate or simply calm the individual, as well as get a faster song in response to the mood, saving time from looking up different songs. An integrated camera is used to capture facial expressions. The study focuses heavily on machine learning and data mining, with several algorithms being used to determine the client's sentiment and evaluate the repercussions.**

***Keywords*—Music suggestion, Facial Recognition, Viola-Jones Algorithm, CNN, Features.**

1. 1. INTRODUCTION

The study presented addressed the creation of Face Emotion Based-Music Player, a computer application aimed at a wide range of users, including music fans. Due to the time-consuming nature of music selection, most users would choose to play the songs in the playlist at random. As a result, some of the songs chosen don't correspond to the users' present moods. Furthermore, no widely used music player can play songs based on the user's mood. The suggested model can detect the user's emotion by extracting the user's facial expression. The proposed model's music player will then play songs based on the emotion category detected. Its goal is to make music listening more enjoyable for music enthusiasts. In this model, the emotions included are neutral, sad, happy, angry, fear, and surprise. The inbuilt camera turns on automatically when we use the emotion tab. It captures a picture of the user's face, and the music player then plays songs based on the mood expressed by the user. Expressions are a fantastic way to learn about a person's mental health. Face expressions are without a doubt the most important method for expressing an individual's feelings. Humans desire to link the melodies they must listen to with the feelings they are experiencing. However, dealing with the music library can be difficult at times. It would be beneficial if the music player was intelligent to deal with music that was based on the individual's current emotional state. The project sets out to use several ways for an emotion recognition framework, breaking down the consequences of the many systems used. From there, it's all about each individual's playlist. The Viola-Jones algorithm is utilized in the program for face identification and extraction of facial features. The approach has fewer memory overheads and requires less computational and processing time, lowering the cost of any added hardware such as EEG or sensors. Happy, angry, sad, fear, surprise, and neutral facial expressions are among the six categories of facial emotions that can be classified. A high-accuracy audio extraction technique is proposed that extracts significant, critical, and relevant information from an audio signal in a significantly shorter amount of time-based on particular audio properties.

2. 2. LITERATURE SURVEY

To enhance people's behavior, various strategies and procedures have been developed. The approaches provided are limited to a few basic emotions. Ramya Ramanathan presented an intelligent agent that sorts a music collection based on the emotions expressed by each song and then uses k-means clustering to recommend a suitable playlist to the user based on his or her current mood. Cohn Kanade's expanded Dataset model is used by the author. Kristin Chankuptarat presented an emotion-based music player that may recommend songs depending on the user's feelings of sadness, happiness, neutrality, fear, surprise, and anger. From a smart band or a smartphone camera, the application receives the user's heart rate or a facial image. The user's heart rate is then classified using the exact categorization to determine the user's sentiment. Here, two types of categorization algorithms are identified: heart rate-based and facial image-based methods. The software then suggests music that is in the same mood as the user's current emotion. Yading Song, Simon Dixon, and Marcus Pearce use an SVM-based technique to classify music emotions based on Last.FM tags. Happiness, anger, sadness, and relaxation are the four emotions covered in this study. Shlok Gilda demonstrated EMP, a cross-platform music player that offers songs based on the user's current mood. Emotion Module, Music Classification Module, and Recommendation Module are the three components that make up the music player. The Emotion Module uses a photo of the user's face as input and uses deep learning algorithms to accurately detect their mood with a 90.23 percent accuracy rate. The Music Classification Module uses audio features to categorize songs into four different mood classes and reaches a stunning result of 97.69 percent. F. Abdat created a facial expression-based emotion identification system.

3. Libraries Used

3.1. OpenCV: The Open-Source Computer Vision Library is a collection of real-time computer vision-related programming utilities. Its major aim is to process images in real-time. The performance of the library can be increased if the native Intel performance primitives are installed on the system via self-optimized functions. Many facial detections and identification functions are available in Open CV. It has a trainer as well as a detector. If you want to train your classified things, such as phones, pens, and so on, you can do so using Open CV[1].

1. **3.2. Tensorflow:** TensorFlow is the most popular Deep Learning framework, and it comes with pre-trained image categorization models. CNN is used to classify photos. In most cases, generating a model simply entails classifying the images and providing a similar image, which is a positive image[2].

2. **3.3. Keras:** Keras is a Google-developed high-level deep learning API for implementing neural networks. It's developed in Python and may be used to quickly create neural networks. Multiple backend neural network computations are also supported.

3. **3.4. NumPy**: NumPy is a Python library that adds support for huge, multi-dimensional arrays and matrices, as well as a large number of high-level mathematical functions to operate on these arrays[4].

4. **3.5. PIL:** Matplotlib is a Python package that allows you to create static, animated, and interactive visualizations. Matplotlib makes simple things simple and difficult things possible[5].

3. 4. PROPOSED SYSTEM

1.

2. 4.1) Methodologies

1. ***4.1.1) Viola-Jones Algorithm:*** Paul Viola and Michael Jones proposed the Viola-Jones algorithm in their paper "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. Viola-Jones is a sophisticated framework that has proven to be especially useful in real-time face identification, despite being an obsolete framework. This system takes a long time to learn but can recognize faces in real-time. The program examines numerous tiny subregions of a picture (this technique only works on grayscale images) and attempts to discover a face by looking for certain attributes in each subregion. Because an image can have multiple faces of varying sizes, it must verify many distinct positions and scales. Haar-like properties were employed by Viola and Jones. In this technique, Viola and Jones used Haar-like properties to detect faces.

The Viola-Jones method includes four key phases, which we will go over in detail in the following sections:

- Choosing Haar-like characteristics
- Creating a whole image
- AdaBoost training is being conducted.
- Cascading classifiers are created.

1. Haar-Like Characteristics: Haar-like features are called after Alfred Haar, a 19th-century Hungarian mathematician who invented Haar wavelets (kind of like the ancestor of haar-like features). The features below depict a box with a light and dark side, which the machine uses to determine what the feature is. As in the edge of a brow, one side may be lighter than the other at times. The center area of the box may be shinier than the surrounding boxes, which can be mistaken for a nose.

There are 3 types of Haar-like features that Viola and Jones identified in their research:

- Edge features
- Line-features
- Four-sided features

Edge and line characteristics can be used to detect edges and lines, respectively. Diagonal features are found using the four-sided characteristics. The feature's value is determined by subtracting the sum of pixel values in the black area from the sum of pixel values in the white area. A plain surface has a value of zero since all of the pixels have the same value and so provide no valuable information. When the areas in the black and white rectangles are very diverse, a Haar-like feature offers you a huge number. We can extract some useful information from the image by using this value. A Haar-like characteristic must give you a significant number to be useful, implying that the areas in the black and white rectangles are very distinct. There are a few features that are well-known for detecting human faces: We receive a favorable response when we apply this specific haar-like feature to the bridge of the nose, for example. Similarly, we combine a number of these characteristics to determine whether or not an image region comprises a human face.

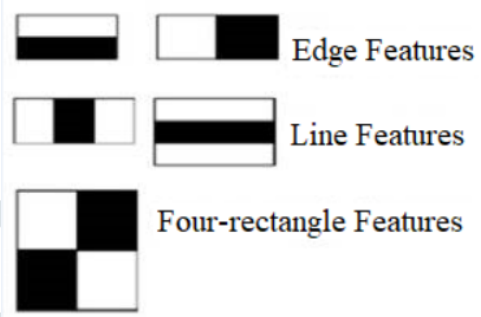

Fig 1: Haar-Like Features

2. Integral Images: We've seen that to calculate a value for each feature, we must compute all of the pixels within that feature. In actuality, because the number of pixels is substantially higher when working with a major feature, these calculations can be somewhat time-consuming. The integral image contributes to our ability to quickly run these extensive calculations to determine whether a feature of several features meets the criteria. An integral image (also known as a summed-area table) is the name of a data structure as well as the algorithm used to create it. It is used to calculate the sum of pixel values in an image or rectangle part of an image quickly and efficiently.

3. AdaBoost Training: Nearly 160,000 features are present in the 2424 detector window, however only a few of these elements are significant for identifying a face. As a result, we apply the AdaBoost method to choose the best characteristics among the 160,000. Each Haar-like feature in the Viola-Jones method represents a weak learner. AdaBoost evaluates the performance of the classifiers you submit to determine the kind and size of a feature that will be included in the final classifier. We test a classifier's performance on all sub-regions of all the images used for training to determine its performance. The classifier will respond strongly to some sub-regions. Positives indicate that the classifier believes the image has a human face. From the perspective of the classifier, subregions that do not generate a strong reaction do not contain a human face. Negatives will be assigned to them. The importance or weight of the classifiers that performed well is increased. The result is a strong classifier, also known as a boosted classifier, which combines the best weak classifiers.

So, when we train the AdaBoost to recognize essential features, we feed it information in the form of training data and then train it to learn from that data to predict. In the end, the algorithm determines whether anything may be classed as a helpful feature or not by defining a minimal threshold.

4. Cascading Classifier: Perhaps the AdaBoost will finally select the greatest features around 2500, but calculating these features for each region is still a time-consuming procedure. We have a 24x24 window that we will glide over the input image to see if any of the regions contain the face. The cascade's job is to quickly eliminate non-faces to save time and computations. As a result, the requisite speed for real-time face detection is achieved.

We devised a cascaded method in which the process of recognizing a face is divided into several steps. In the first step, we have a classifier made out of our best features; in other words, the

subregion goes through the best characteristics in the first stage, such as the feature that identifies the nasal bridge or the feature that identifies the eyes. All of the remaining features will be added in the following stages.

The first step evaluates an image sub-region when it enters the cascade. The stage's output is maybe if it evaluates the sub-region as positive, implying that it believes it is a face.

When a sub-region receives a maybe, it moves on to the next stage of the cascade, and so on until we reach the final stage.

The image is finally classified as a human face and displayed to the user as a detection if all classifiers approve it.

So, how does this assist us in increasing our speed? In general, if the first stage returns a negative result, the image is instantly dismissed as lacking a human face. It is also discarded if it passes the first step but fails the second. Essentially, the image can be deleted at any point throughout the classification process[9]-[11].

2. ***4.1.2. Convolutional Neural Network:*** A Convolutional Neural Network (CNN) is a type of neural network that specializes in processing data with a grid-like architecture, such as an image. A binary representation of visual data is a digital image. It consists of a grid-like arrangement of pixels with pixel values indicating how bright and what color each pixel should be. The second we see an image, our brain analyses a massive amount of data. Each neuron has its receptive field and is coupled to other neurons in such a way that the visual area is covered. In the biological vision system, each neuron responds to stimuli only in a limited part of the visual field called the receptive field, and each neuron in a CNN analyses data only in its receptive field. The layers are designed to detect simpler patterns (lines, curves, etc.) first, followed by more complicated patterns (faces, objects, etc.). One can give computers sight by utilizing a CNN.

A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer.

i. Convolutional Layer: The convolutional layer is the most important component of CNN. Convolution is a mathematical procedure for combining two sets of data. Convolution is applied to the input data using a convolution filter to create a feature map in our example. There are a lot of phrases being utilized, so let's go over each one individually. We have a kernel/filter matrix and an input picture matrix. Convolution is accomplished by sliding this filter over the input. We conduct element-wise matrix multiplication and total the results at each position. This total is entered into the feature map. Then we slide the filter to the right and repeat the process, this time adding the result to the feature map. We'll keep doing it this way, and the convolution results will be aggregated in the feature map.

ii. Pooling Layer: Pooling is frequently done after a convolution process to lower the dimensionality. This allows us to limit the number of parameters, which reduces training time while also preventing overfitting. Each feature map is downsampled independently by pooling layers, reducing the height and breadth while maintaining the depth. The most frequent sort of pooling is max pooling, which simply takes the pooling window's maximum value. Pooling, unlike convolution, does not have any parameters. It simply drags a window across its input and takes the window's maximum value. We provide the window size and stride in the same way that we do with a convolution.

iii. Fully Connected Layer: To complete the CNN design, we add a pair of fully linked layers after the convolution + pooling layers. This is the same fully connected ANN architecture that we discussed in Part 1 of this series.

Convolution and pooling layers both produce 3D volumes as output, but a fully connected layer requires a 1D vector of values. As a result, we convert the last pooling layer's output to a vector, which we use as the input to the fully connected layer. Flattening is essentially the process of converting a 3D volume of integers into a 1D vector[7],[8].

3. 4.2 Flow Chart

The flow of our project is given below:

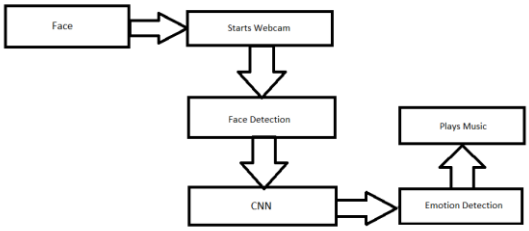

Fig. 2

4. 4.3 Working Modules

1. 4.3.1. Face Detection and Capturing Module

1. The main goal of this module is to take photos, thus we're using a common device, a camera, but you could use any physiological device instead. We're going to use the OpenCV library for this. This makes it easier to combine it with other NumPy-compatible libraries, and it's mostly used as a real-time vision system.

2. 4.3.2. Emotion Detection Module

The frame of the acquired image from the webcam stream is transformed to a grayscale image after it is captured to increase the performance of the classifier, which is used to identify the face in the picture. When the conversion is finished, the image is transmitted to the classifier algorithm, which uses feature extraction techniques to extract the face from the web camera stream frame. Individual features are taken from the extracted face and passed to the trained network to detect the user's sentiment.

3. 4.3.3. Song Selection Module

1. Following the extraction of the user's emotion, the user is presented with a playlist depending on the emotion expressed by the user. The user is shown a list of music based on the feeling, and he or she can listen to whichever song they would like to play. The songs are displayed in that order based on how frequently the user listens to them.

4. 5. CONCLUSION

The Face Emotion-Based Music Player automates and improves the end-music user's player experience. The program meets the basic demands of music listeners without bothering them in the way that other apps do: it boosts the system's contact with the user in a variety of ways. It makes the end-job user's easier by taking a picture with a camera, assessing their mood, and offering a personalized playlist using a more advanced and interactive system.

ACKNOWLEDGEMENT

We would like to thank our seminar guide, Asst. Prof. Manisha Pise for her help and guidance.

REFERENCES

[1] https://opencv.org/about/

[2] https://www.tensorflow.org/learn

[3] https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras

[4] https://www.w3schools.com/python/numpy/numpy_intro.asp

[5] https://www.geeksforgeeks.org/python-pillow-a-fork-of-pil/

[6] http://news.discovery.com/human/music-dopamine-happiness-brain110110.html

[7] https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

[8] https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

[9] https://www.mygreatlearning.com/blog/viola-jones-algorithm/#sh3

[10] https://towardsdatascience.com/the-intuition-behind-facial-detection-the-viola-jones-algorithm-29d9106b6999#:~:text=Detection,-Viola%2DJones%20was&text=The%20Viola%2DJones%20algorithm%20first,which%20will%20be%20explained%20later.

[11] https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14