Emotion based Music player

Polineni Sumanth, B.Tech Student, Department of Information Technology

Patha Nataraj, B.Tech Student ,Department of Information Technology

Pulabala Sreenadh, B.Tech Student, Department of Information Technology

Dr. G Madhukar, Assistant Professor, Department of IT,

CMR Technical Campus, Telangana, India.

ABSTRACT

Visual sentiment analysis, which investigates humans' emotional responses to visual stimuli such as images and videos, has been a fascinating and challenging problem. It attempts to recognize the high-level content of visual data. The success of current models can be attributed to the development of robust computer vision algorithms. The majority of existing models attempt to solve the problem by recommending either robust features or more complex models. The main proposed inputs are visual features from the entire image or video. Local areas have received little attention, which we believe is important to the emotional response of humans to the entire image. Image recognition is used to find people in images and analyze their sentiments or emotions. The CNN algorithm is used in this project to accomplish this task. Given an image, it will search for faces, identify them, place a rectangle in their positions, and describe the emotion found with a percentage of emotions displayed. The emotion output will be in audio format.

After detecting emotions based on predicted emotions, songs stored in the system is automatically played, such as a happy song when the emotion is happy and a sad song when the emotion is sad.

1. INTRODUCTION:

The emotional response of humans to visual stimuli such as images and videos is

studied using visual sentiment analysis. It differs from textual sentiment analysis (Pang and Lee 2008), which focuses on the emotional response of humans to textual semantics. Visual sentiment analysis has recently achieved comparable performance to textual sentiment analysis (Borth et al. 2013; Jou et al. ; You et al. 2015). This can be attributed to deep learning's success on vision tasks (Krushinski, Sutskever, and Hinton 2012), which makes understanding high-level visual semantics tractable, such as image aesthetic analysis (Lu et al. 2014) and visual sentiment analysis (Borth et al. 2013). Studies on visual sentiment analysis have focused on designing visual features ranging from pixel-level (Siersdorfer et al. 2010a) to middle attribute level (Borth et al. 2013) to recent deep visual features (Siersdorfer et al. 2010b) (You et al. 2015; Campos, Jou, and Giro-i Nieto 2016). As a result of more and more robust visual features, the performance of visual sentiment analysis systems has gradually improved. Almost all of these approaches, however, have attempted to reveal the high-level sentiment from a global perspective of the entire images. Little attention has been paid to research into where we get our sentimental responses and how the local regions approach the task of visual sentiment analysis. We are attempting to solve these two difficult problems in this work. To learn the correspondence between local image regions and sentimental visual attributes, we use a recently proposed attention model (Mnih et al. 2014; Xu et al. 2015). We are able to identify the local image regions that are relevant to sentiment analysis in this manner. Following that, a sentiment classifier is constructed using the visual features extracted from these local regions.

2. LITERATURE SURVEY

Many researchers are interested in using Face Emotion Recognition to improve the learning environment (FER). Tang et al. [3] proposed a system for analyzing students' facial expressions in order to determine the effectiveness of classroom teaching. Data acquisition, face detection, face recognition, facial expression recognition, and post-processing are the five phases of the system. For classification, K-nearest neighbor (KNN) is used, and for pattern analysis, Uniform Local Gabor Binary Pattern Histogram Sequence (ULGBPHS) is used. Savva et al. [4] proposed a web application that analyses students' emotions while they are participating in active face-to-face classroom instruction. The application collects live recordings from webcams installed in classrooms and then applies machine learning algorithms to them. Whitehill et al. proposed in [5] an approach for detecting student engagement based on facial expressions. The method employs Gabor features and the SVM algorithm to detect student engagement as they interact with cognitive skills training software. The authors obtained labels from videos that had been annotated by humans. The authors of [6] then used computer vision and machine learning techniques to identify the affect of students in a school computer laboratory as they interacted with an educational game designed to explain fundamental concepts of classical mechanics. The authors of proposed a system in [7] that identifies and monitors student emotion and provides real-time feedback in order to improve the e-learning environment for greater content delivery. In an e-learning environment, the system uses the moving pattern of the eyes and head to deduce relevant information in order to understand students' moods. Ayvaz et al. [8] created a Facial Emotion Recognition System (FERS) that recognizes students' emotional states and motivation in videoconference-based e-learning. The system achieves the highest accuracy rates by utilizing four machine learning algorithms (SVM, KNN, Random Forest, and Classification & Regression Trees).

3. METHODOLOGY

3.1 Module for Detecting Faces

Face detection is the detection of a face from an image or video input. Face detection algorithms come in a variety of flavours. For face detection, the Viola Jones algorithm is used. The main steps in Viola Jones' algorithm are as follows:

HAAR feature - Some of the characteristics of the face are represented by HAAR features. Har features are similar to convolution kernals that are used to detect the presence of a feature in an image. Each feature yields a single value, which is calculated by subtracting the sum of pixels within the white rectangle from the sum of pixels within the black rectangle. In the feature, the black regions are replaced by plus ones, while the white regions are replaced by minus ones.

Integral image - In HAAR feature calculation, every time the window moves, all pixels in the black and white regions must be added up. It is a time-consuming procedure, and the solution is integral image. Rather than summing up all pixels under a rectangle with only four corner values of the integral image, it reduces the computation. Simply add the values of the pixels to the top and left to find the value of any pixel.

Adaboost - The Viola Jones algorithm starts the evaluation of features in any given image with a 24*24 window as the base window. If we take into account all possible haar feature parameters such as position scale and type, we end up calculating 160,000+ features in this window, which is practically impossible. So the basic idea is to remove a lot of features that are redundant or not useful and only keep the features that are extremely useful. This one by Adaboost removes 160 thousand features and reduces the number of features we need to evaluate to a couple of thousands. Adaboost's extracted features are weak classifiers. Adaboost creates a weak classifier linear combination. Cascading - The basic principle of Viola Jones' face detection algorithm is to scan the detector many times through that image, each time with a different size. Despite the fact that an image should contain one or more faces, it is clear that a disproportionate number of the evaluated sub-windows may still be negatives. As a result, the algorithm should focus on quickly discarding non-faces. Because of the computation cost, a single strong classifier formed from a linear combination of all the best features is not suitable for evaluating on each window.

3.2 Facial Feature Extraction Module

CNN is used for feature extraction. To train the system for the emotion recognition module, we must use datasets containing images of happy, angry, sad, and neutral emotions. CNN has the unique ability of automatic learning to identify features from dataset images for model construction. To put it another way, CNN can learn features on its own. CNN can create an internal representation of a two-dimensional image. This is represented as a three-dimensional matrix, and operations on this matrix are performed for training and testing. What's more, in some other neural networks, such as fully connected networks, all nodes in a layer are connected to all nodes in the next layer. Weights are associated with each connection. The computational complexity will rise as a result. Even then, in CNN, nodes in one layer are only connected to valid nodes in the next layer. As a result, the computational complexity will be reduced. This includes several layers for training and testing input images. The final layer is fully connected and has a classification task, so images can be classified based on emotions. The detected emotions should fall into one of the following categories: angry, happy, sad, or neutral. The entire dataset will be divided into two before entering the CNN. 80 percent of this will be used for training, with the remaining twenty percent for testing. Following that, this model will be tested. During the testing period, accuracy can be calculated by determining whether or not the images are correctly classified. Accuracy can be improved by increasing the number of epochs or the number of images in the dataset. The neural network's convolution layer will receive the input. Filtering is the process that occurs at the convolution layer. The math behind matching is known as filtering. The first step is to align the feature and image patch. Then, for each image pixel, multiply it by the corresponding feature pixel. Add them all up and divide by the feature's total number of pixels. After filtering, one image becomes a stack of images in convolution. Nonlinear operations can be performed using ReLU, tanh, or sigmoid functions. Every negative value in ReLU is converted to zero, while positive values remain unchanged. When the unit is not active, the leaky ReLU allows for a small gradient. As a result, information will not be overlooked. This also resolves the dying ReLU issue. ReLU outperforms the other two non-linear functions. The pooling layer will come after the convolution layer. This is done to reduce the image stack produced by the convolution layer. The number of parameters will be reduced as a result [20]. A window size is selected here (usually 2 or 3). Then choose a stride (usually 2). A stride is the number of pixels that shift across the input matrix. When the stride is one, we move the filters one pixel at a time; when it is two, we move the filters two pixels at a time, and so on. The window is then dragged across the filtered images. Pooling can be classified into three types. There are three types of pooling: maximum pooling, average pooling, and sum pooling. Max pooling selects the largest element from a rectified feature map. Taking the average of the elements in the window is known as average pooling. Sum pooling is the sum of all elements in the feature map. More convolution and pooling layers can be added until the desired accuracy is achieved. Following the pooling layer, the matrix is flattened to a vector and fed to the fully connected layer. The goal of flattening is to convert a two-dimensional feature matrix into a feature vector that can be fed into a neural network or classifier. The vector elements combine to form models when the layers are fully connected. Finally, classification is accomplished through the use of activation functions such as the Softmax or Sigmoid functions.

3.3 Emotion Detection and Song Classification Module

The neural network classifier produces one of four emotion labels: happy, angry, sad, or neutral according to detected emotion songs are played.

4. IMPLEMENTATION

We used the FER2013 [12] database to collect data for training our CNN architecture. It was created with the Google image search API and displayed at the ICML 2013 Challenges. The database faces have been automatically normalized to 4848 pixels. The FER2013 database contains 35887 images with 7 expression labels (28709 training images, 3589 validation images, and 3589 test images). Each emotion is represented by a certain number of images. CNN (CNN) Putting it all together: We used the OpenCV library to capture live web camera frames and detect students' faces using the Haar Cascades method. Haar Cascades is based on the Adaboost learning algorithm. In order to provide an effective classifier result, the Adaboost learning algorithm selected a small number of significant features from a large set.we used the Image Data Generator class in Keras to perform image augmentation. This class allowed us to rotate, shift, shear, zoom, and flip the training images. The following settings are used: rotation range=10, width shift range=0.1, zoom range=0.1, height shift range=0.1, and horizontal flip=True. Then, for our CNN model, we used four convolutional layers, four pooling layers, and two fully connected layers. Following that, we used the ReLU function to provide nonlinearity in our CNN model, as well as batch normalization to normalize the activation of the preceding layer at each batch and L2 regularization to apply penalties to the model's various parameters. 22 We trained our Convolutional Neural Network model on the FER 2013 database, which contains seven emotions (happiness, anger, sadness, disgust, neutral, fear and surprise) The detected face images were resized to 4848 pixels and converted to grayscale images before being fed into the CNN model.

5. SYSTEM ARCHITECTURE:

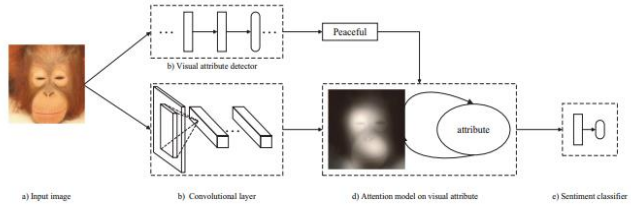


Fig no. 5.1 System Architecture

Using a Convolutional Neural Network (CNN) architecture, we describe our proposed system for analysing students' facial expressions. First, the system detects faces in the input image, and these detected faces are cropped and normalised to 4848 pixels in size. These face images are then fed into CNN. Finally, the results of facial expression recognition are displayed (anger, happiness, sadness, disgust, surprise or neutral). The structure of our proposed approach is depicted in Figure 5.1. A Convolutional Neural Network (CNN) is a deep artificial neural network that, when compared to other image classification algorithms, can identify visual patterns from input images with minimal pre-processing. This means that the network learns the filters that were hand-engineered in traditional algorithms [19]. A neuron is the most important unit within a CNN layer. They are linked together in such a way that the output of neurons at one layer becomes the input of neurons at the next. The backpropagation algorithm is used to compute the partial derivatives of the cost function. Convolution is the process of producing a feature map by applying a filter or kernel to an input image. In fact, as illustrated in Figure 5.1, the CNN model has three types of layers.

Modules:

User:

Using this module user can connect camera to application and track live video of user with different sentiment expressions and capture image of every frame and send data to CNN model which will predict and give live emotions.

Based on the analyzed data values sent from CNN model and received by user and showed on the live camera with type of sentiment (sad, happy..etc)

Dataset Collection:

collect data fer 2013 dataset which has pixlel values as features and emotions as labels.

Preprocessing:

In this stage dataset is divided in to features and labels and stored in x and y values

Initializing CNN Model:

In this stage cnn model is initialized and features and labels are passed to fit statement and algorithm is trained. Model is saved in .h5 format.

CNN Model Training:

Fer 2013 data set with facial emotions are taken as input and trained using CNN algorithm model is saved to disk for prediction of live emotions.

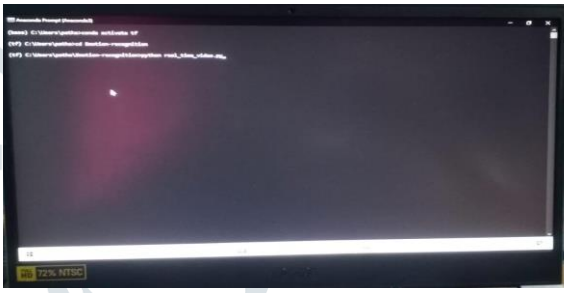
- ▶ . **Prediction:**

In this stage camera will open and faces are recognized and predicted with trained model and based on emotion song will be played..

6. CONCLUSION

Which we believe is quite relevant to the emotional response of humans to the entire image Image recognition is used to find people in images and analyse their sentiments or emotions. To accomplish this task, this project makes use of the Google platform's Vision services. Given an image, it would look for faces, identify them, draw a rectangle in their positions, describe the emotion found, and play music based on the emotion.

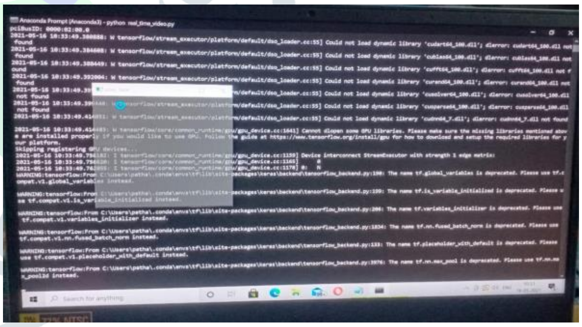
7. RESULTS



```
Anaconda Prompt (Anaconda3)
(base) C:\Users\patha>conda activate tf
(tf) C:\Users\patha>cd Emotion-recognition
(tf) C:\Users\patha\Emotion-recognition>python real_time_video.py
```

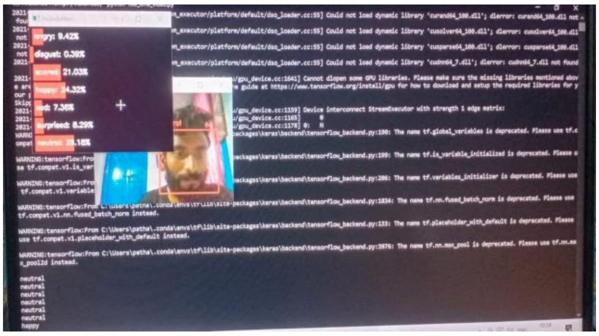
Open Anaconda through search and type commands to activate, detect emotion and to access the webcam

ACCESS TO WEBCAM



```
Anaconda Prompt (Anaconda3) - python real_time_video.py
pciBusID: 0000:02:00.0
2021-05-16 10:33:49.380888: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudart64_100.dll'; dlerror: cudart64_100.dll not
 found
2021-05-16 10:33:49.384608: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cublas64_100.dll'; dlerror: cublas64_100.dll not
 found
2021-05-16 10:33:49.388449: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cufft64_100.dll'; dlerror: cufft64_100.dll not f
ound
2021-05-16 10:33:49.392004: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'curand64_100.dll'; dlerror: curand64_100.dll not
 found
2021-05-16 10:33:49.39[illegible]m/default/dso_loader.cc:55] Could not load dynamic library 'cusolver64_100.dll'; dlerror: cusolver64_100.dll
 not found
2021-05-16 10:33:49.39[illegible] tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cusparse64_100.dll'; dlerror: cusparse64_100.dll
 not found
2021-05-16 10:33:49.41[illegible] W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudnn64_7.dll'; dlerror: cudnn64_7.dll not found

2021-05-16 10:33:49.41[illegible] W tensorflow/core/common_runtime/gpu/gpu_device.cc:1641] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned abov
e are installed properly if you would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for y
our platform.
Skipping registering GPU devices...
2021-05-16 10:33:49.7[illegible] I tensorflow/core/common_runtime/gpu/gpu_device.cc:1159] Device interconnect StreamExecutor with strength 1 edge matrix:
2021-05-16 10:33:49.7[illegible] I tensorflow/core/common_runtime/gpu/gpu_device.cc:1165]      0
2021-05-16 10:33:49.7[illegible] I tensorflow/core/common_runtime/gpu/gpu_device.cc:1178] 0:   N
WARNING:tensorflow:From C:\Users\patha\.conda\envs\tf\lib\site-packages\keras\backend\tensorflow_backend.py:190: The name tf.global_variables is deprecated. Please use tf.c
ompat.v1.global_variables instead.

WARNING:tensorflow:From C:\Users\patha\.conda\envs\tf\lib\site-packages\keras\backend\tensorflow_backend.py:199: The name tf.is_variable_initialized is deprecated. Please u
se tf.compat.v1.is_variable_initialized instead.

WARNING:tensorflow:From C:\Users\patha\.conda\envs\tf\lib\site-packages\keras\backend\tensorflow_backend.py:206: The name tf.variables_initializer is deprecated. Please use
 tf.compat.v1.variables_initializer instead.

WARNING:tensorflow:From C:\Users\patha\.conda\envs\tf\lib\site-packages\keras\backend\tensorflow_backend.py:1834: The name tf.nn.fused_batch_norm is deprecated. Please use
tf.compat.v1.nn.fused_batch_norm instead.

WARNING:tensorflow:From C:\Users\patha\.conda\envs\tf\lib\site-packages\keras\backend\tensorflow_backend.py:133: The name tf.placeholder_with_default is deprecated. Please
use tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From C:\Users\patha\.conda\envs\tf\lib\site-packages\keras\backend\tensorflow_backend.py:3976: The name tf.nn.max_pool is deprecated. Please use tf.nn.ma
x_pool2d instead.
```

RESULT



```
2021-[illegible]m_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'curand64_100.dll'; dlerror: curand64_100.dll not
 found
2021-[illegible]m_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cusolver64_100.dll'; dlerror: cusolver64_100.dll
 not found
2021-[illegible]m_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cusparse64_100.dll'; dlerror: cusparse64_100.dll
 not found
2021-[illegible]m_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudnn64_7.dll'; dlerror: cudnn64_7.dll not found

2021-[illegible]/gpu_device.cc:1641] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned abov
e are[illegible] guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for y
our [illegible]
Skip[illegible]
2021-[illegible]/gpu_device.cc:1159] Device interconnect StreamExecutor with strength 1 edge matrix:
2021-[illegible]/gpu_device.cc:1165]      0
2021-[illegible]/gpu_device.cc:1178] 0:   N
WARN[illegible]packages\keras\backend\tensorflow_backend.py:190: The name tf.global_variables is deprecated. Please use tf.c
ompat[illegible]

WARNING:tensorflow:Fro[illegible]packages\keras\backend\tensorflow_backend.py:199: The name tf.is_variable_initialized is deprecated. Please u
se tf.compat.v1.is_var[illegible]

WARNING:tensorflow:Fro[illegible]packages\keras\backend\tensorflow_backend.py:206: The name tf.variables_initializer is deprecated. Please use
 tf.compat.v1.variable[illegible]

WARNING:tensorflow:From C:\Users\patha\.conda\envs\tf\lib\site-packages\keras\backend\tensorflow_backend.py:1834: The name tf.nn.fused_batch_norm is deprecated. Please use
tf.compat.v1.nn.fused_batch_norm instead.

WARNING:tensorflow:From C:\Users\patha\.conda\envs\tf\lib\site-packages\keras\backend\tensorflow_backend.py:133: The name tf.placeholder_with_default is deprecated. Please
use tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From C:\Users\patha\.conda\envs\tf\lib\site-packages\keras\backend\tensorflow_backend.py:3976: The name tf.nn.max_pool is deprecated. Please use tf.nn.ma
x_pool2d instead.

neutral
neutral
neutral
neutral
neutral
neutral
happy
```

REFERENCES

1. A. Savva, V. Stylianou, K. Kyriacou, and F. Domenach, “Recognizing student facial expressions: A web application,” in 2018 IEEE Global Engineering Education Conference (EDUCON), Tenerife, 2018, p. 1459-1462.
2. Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. In ICLR 2015.
3. Borth, D.; Chen, T.; Ji, R.; and Chang, S.-F. Sentibank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content.
4. Borth, D.; Ji, R.; Chen, T.; Breuel, T.; and Chang, S.-F. 2013. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In Proceedings of the 21st ACM international conference on Multimedia, 223–232. ACM.
5. C. Tang, P. Xu, Z. Luo, G. Zhao, and T. Zou, “Automatic Facial Expression Analysis of Students in Teaching Environments,” in Biometric Recognition, vol. 9428, J. Yang, J. Yang, Z. Sun, S. Shan, W. Zheng, et J. Feng, Éd. Cham: Springer International Publishing, 2015, p. 439-447.
6. Campos, V.; Jou, B.; and Giro-i Nieto, X. 2016. From pixels to sentiment: Fine-tuning cnns for visual sentiment prediction. arXiv preprint arXiv:1604.03489.
7. Cao, D.; Ji, R.; Lin, D.; and Li, S. 2014. A cross-media public sentiment analysis system for microblog. Multimedia Systems 1–8.
8. J. Whitehill, Z. Serpell, Y.-C. Lin, A. Foster, and J. R. Movellan, “The Faces of Engagement: Automatic Recognition of Student Engagementfrom Facial Expressions,” IEEE Transactions on Affective Computing, vol. 5, no 1, p. 86-98, janv. 2014.
9. P. Ekman and W. V. Friesen, “Constants across cultures in the face and emotion,” Journal of Personality and Social Psychology, vol. 17, no 2, p. 124-129, 1971.