MUSIC RECOMMENDATION SYSTEM BASED ON HUMAN EMOTION

A PROJECT REPORT
Submitted by

Abhash Shrestha [Reg No: RA1611003011297]

Naresh Bohara [Reg No: RA1611003011324]

Under the guidance of
Dr. G. Niranjana, Ph. D

(Associate Professor, Department of Computer Science & Engineering)

In Partial Fulfillment
for the Degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR-603 203

APRIL 2020

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this project report titled "**MUSIC RECOMMENDATION SYSTEM USING HUMAN EMOTION**" is the bonafide work of **ABHASH SHRESTHA [Reg No: RA1611003011297], NARESH BOHARA [Reg No: RA1611003011324]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. G. Niranjana, Ph. D
SUPERVISOR
Associate Professor
Department of Computer Science and Engineering

Dr. B. Amutha
HEAD OF THE DEPARTMENT
Department of Computer Science and Engineering

Signature of the Internal Examiner

Signature of the Internal Examiner

Department of Computer Science and Engineering

SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

<u>To be completed by the student for all assessments</u>

Degree/ Course : B. Tech. / Computer Science and Engineering

Student Name : Abhash Shrestha, Naresh Bohara

Registration Number: RA1611003011297, RA1611003011324

Title of Work : Music Recommendation System Based on Human Emotion

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. Sandeep Sancheti, Vice Chancellor**, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. C. Muthamizhchelvan, Director**, Faculty of Engineering and Technology, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. B. Amutha, Professor & Head**, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Academic Advisor, **Dr.R. Annie Uthra**, Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her great support at all the stages of project work.

We would like to convey our thanks to our Panel Head, **Dr.G. Niranjana**, Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her inputs during the project reviews.

We register our immeasurable thanks to our Faculty Advisor, **Ms. Renuka Devi**, Assistant Professor (O G), Department of Computer Science and Engineering, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my supervisor, **Dr. G. Niranjana**, Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for providing me an opportunity to pursue my project under her mentorship. She provided me the freedom and support to explore the research topics of my interest. Her passion for solving the real problems and making a difference in the world has always been inspiring.

We sincerely thank staff and students of the Computer Science and Engineering Department, SRM Institute of Science and Technology, for their help during my research. Finally, we would like to thank my parents, our family members and our friends for their unconditional love, constant support and encouragement.

ABHASH SHRESTHA
NARESH BOHARA

ABSTRACT

Artificial Intelligence is one of the most prominent technologies of the modern world, it has been responsible for the revolution occurring in every major industry, as well as trade. However, it is not without its own set of challenges, and is not a 100% compatible with humans and their varied emotion. In this project we propose how an AI system can detect and understand a human's emotions, mainly by detection of their facial expression.

Human communication can happen in multiple levels, verbal, nonverbal, textual, pictorial, signs and so on. Facial expression, as we know, is a very explicit non-verbal mode of communication in humans. If a machine is capable of detecting and accurately classifying the particular emotion, it will undoubtedly improve the human-machine interaction.

In this paper we propose a system for lighting and position invariant recognition of facial expression. Information from the human face can be compared with dense model in an iterative manner. We will use a classifier to classify the images into different emotion categories. Variation in illumination, pose, distance from the camera etc. can influence the accuracy of facial recognition and emotion classification. According to the emotion, we can play a suitable musical track.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ADABOOST	ADAPTIVE BOOSTING
AFERS	AUTOMATIC FACIAL EXPRESSION RECOGNITION SYSTEM
A I	ARTIFICIAL INTELLIGENCE
ANN	ARTIFICIAL NEURAL NETWORK
CNN	CONVOLUTIONAL NEURAL NETWORK
DAN	DEEP ALIGNMENT NETWORK
DLSS	DEEP LEARNING SUPER-SAMPLING
GLCM	GRAY LEVEL CO-OCCURRENCE MATRIX
GMM	GAUSSIAN MIX MODEL
HMM	HIDDEN MARKOV MODEL
KNN	K-NEAREST NEIGHBOR
MFCC	MELFREQUENCY CEPSTRAL COEFFICIENT
ONEIRS	OPEN-ENDED NEURO-ELECTRIC INTELLIGENT ROBOT OS
OPENCV	OPEN SOURCE COMPUTER VISION LIBRARY
PRNN	PATTERN RECOGNITION NEURAL NETWORK
SER	SPEECH RECOGNITION SYSTEM
SVM	SUPPORT VECTOR MACHINE
UML	UNIFIED MODELING LANGUAGE

LIST OF SYMBOLS

ΔX	CHANGE IN THE DIRECTION X
ΔY	CHANGE IN THE DIRECTION Y
ΔC	CHANGE IN OUR FUNCTION
∂	PARTIAL DERIVATIVE
∇	GRADIENT

CHAPTER 1

INTRODUCTION

1.1 Object Detection

Object Detection is a common Computer Vision problem which deals with distinguishing and spotting object of certain classes in the image. Deciphering the object localization can be done via various methods, including constructing bounding lines around the object or marking all the pixel in the picture which encloses the object. Object detection had been a topic of research even before the mainstream popularity of CNNs in Computer Vision.

Fig 1.1: Object Detection with bounding boxes

Object detection, before the advent of Deep Learning used to be a multiple step process, which began with edge detection and feature extraction by the utilization of techniques like SIFT, HOG etc. These images were then matched with existing object templates, mostly at multi scale levels I order to find and contain objects present in the image.

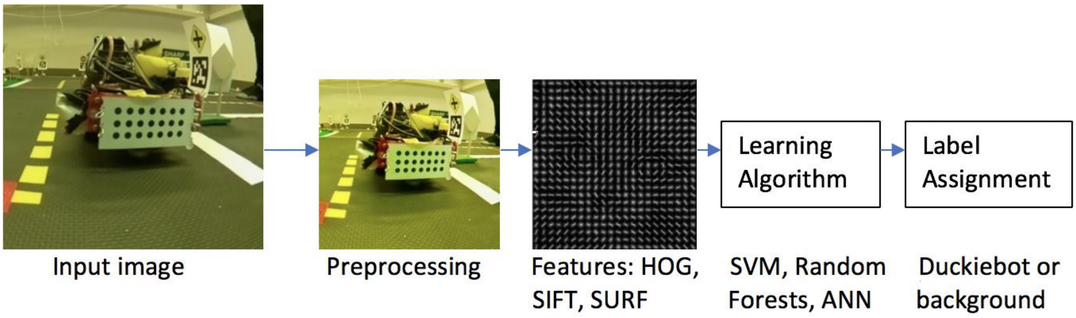

Fig 1.2: Early Object Detection

1.2 Object Recognition

Object Recognition has established itself as one of the most exciting fields in computer vision and AI these days. The capacity to quickly recognize all the objects in a certain region of interest seems to be no longer a secret of evolution. With the invention and expansion of Convolutional Neural Network architectures, underpinned by big training data along with advanced computing machinery, a computer now can exceed human performance in object recognition work under some defined settings, such as face recognition.

Object recognition an areas of computer vision that is evolving at an unprecedented rate. Every year, new algorithms/models surpass the previous ones. Today, there are many pre-trained models for object detection such as YOLO, RCNN, Fast RCNN, Mask RCNN, Multibox etc. So, it does not take much effort to spot the objects in a video or in an image.

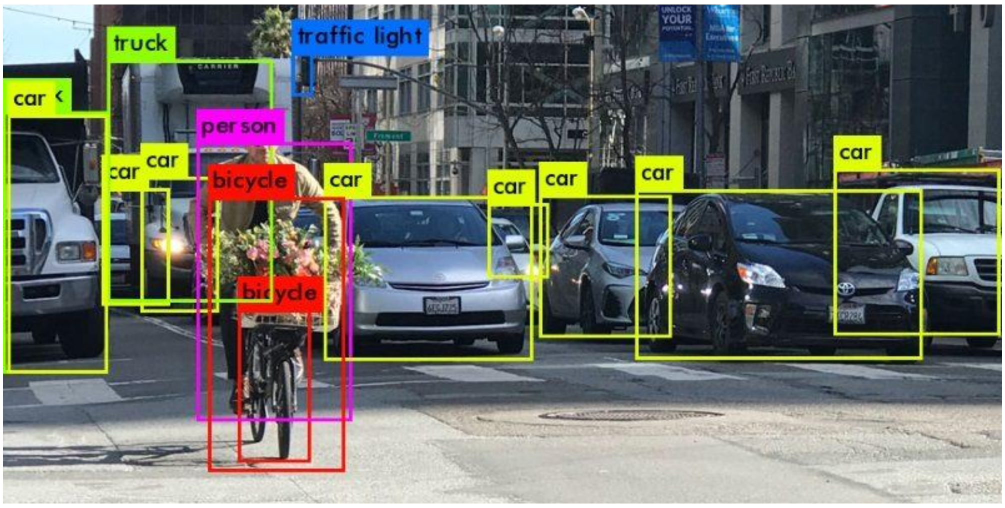

Figure 1.3: Example of Object recognition

1.3 Formal Problem Definition

The field of computer vision has experienced substantial progress recently, owing largely to advances in deep learning, specifically convolutional neural nets (CNNs). Image classification, where a computer classifies or assigns labels to an image based on its content, can often see great results simply by leveraging pre-trained neural nets and fine-tuning the last few throughput layers.

However, much research and work are being done in this field. As a result, huge leaps have been made such as widespread use of biometrics in cell phones, improvement in public monitoring and observation through the use of cameras and so on. Facial expressions are triggered for a period of time when an emotion is experienced and so emotion detection can be achieved by detecting the facial expression related to it. We also know that music can alter a person's emotion regardless of their age, gender, or language.

We have attempted tie together the object detection system and emotion detection system in order to recommend the user appropriate music for their real time emotional state, detected using their facial expression.

1.4 Motivations

Emotions are an inseparable human interaction. Training a machine to recognize and take the emotion as input can greatly influence the human-computer interaction efficiency making the use of computers reliable. As we know, Music can greatly influence a person's emotional state, the objective is to detect and classify the user's emotion and recommend the appropriate musical tracks.

A real-life scenario where this could be used is with people who are not fully physical capable, for example: old age homes, kindergartens. It might not always be possible to monitor them so we can use a system that detects their emotional state and plays the appropriate music for them in order to improve their mood, calm them down and so on.

1.4.1 Scope of System

Areas where emotion recognition can be used: Old age homes, Kindergartens, Differently abled people, faster and seamless human-computer interaction, smart devices such as phones, laptops, tablets etc.

We will use a classifier to classify the images into different emotion categories. Variation in illumination, pose, distance from the camera etc. can influence the accuracy of facial recognition and emotion classification. According to the emotion, we can play a suitable musical track.

CHAPTER 2

LITERATURE SURVEY

2.1 Literature Review

2.1.1 General

In this chapter, we show the outcomes of the formal literature survey which we had conducted for this project. We recognize and state the technological course of action and methodologies employed in the field of object recognition and emotion classification. This survey facilitated us in regards to the key features of our final model.

The word literature review means study of an article, published paper and other such academia. Literature reviews cannot be considered as original content, and are invariably regarded as paraphrasing work. In the subsection that will follow we have described the peculiar features, methodologies and differences in each of the reference papers that we examined.

2.1.2 Literature Review on Object Detection and Emotion Classification

[A]

Artificial Intelligence is a revolutionary technology leading the way in technological advancements in today’s world. However, affective computing has been a sort of a bottleneck in realizing emotional machines with advanced artificial intelligence capable of empathy with humans. With this the AI can take into account the emotions of the humans it serves while observing the surrounding environments to act for completing the goal of objective environment as well as the human user’s emotion, which is subjective.

In a paper written by **[1] Chao Gong, Fuhong Lin*, Xianwei Zhou, Xing Lü, titled "Amygdala- Inspired Affective Computing: to Realize Personalized Intercranial Emotions with Accurately Observed External Emotions"** they have proposed that an amygdala inspired affective computing framework can identify various types of human emotions. First, the neural network was able to precisely identify emotions with the help of inputs over a period of time, and not immediate inputs. The Convolutional Neural Network was compressed with the help of pruning as well as hashing tricks for even faster recognition of emergency emotions. According to the experimental results, it has shown a low level of latency and extremely high recognition accuracy. This is a major breakthrough in making A.I capable of understanding and empathizing with humans.

[B]

Human computer interaction can be greatly improved with the automatic estimation of human emotion. Continuous emotion tracking and recognition allocate a value to each frame in a sequence of emotions.

In another paper titled "End-to-end continuous emotion recognition from video using 3d Convlstm networks" by **[2] Jian Huang, Ya Li, Jianhua Tao, Zheng Lian, Jiangyan Yi, propose the use of ConvLSTM model**, on the groundwork laid by 3D convolution with the objective of constructing an end to end continuous emotion system from video. The 2D ConvNet, convolutions pay major attention to the spatial information due to which it loses temporal information contained in the input signal. However, in 3D convolution and pooling process have a 3D kernel administered to overlapping 3D cubes spatiotemporally, and this helps in retaining the temporal information of the input signals resulting in an output volume. The system will fuse feature extraction and regressor into a unified system by the use only one network. Temporal and Max pooling are analyzed for the optimization the recognition system. The outcome of the experiment makes it clear that max pooling can bolster the performance of the system. However, it is clear that temporal pooling achieves the most expedient outcome through extraordinary performance. Temporal pooling also ends up saving a lot of training time as well as memory. ConvLSTM displays a better capability and efficient performance in comparison to the 3D ConvNet notably in arousal, justifying its capability in modeling emotional spatiotemporal relationships. This tremendously facilitates the performance of continuous emotion recognition system.

[C]

Emotions can have a wide range of variety; sometimes 2 different emotions might have almost similar facial features, so it is important to analyze the micro expressions in order to precisely classify the emotion. The emotion detection accuracy might vary depending on the angle of the person with respect to the camera. Normally such expressions might only occur for a very short time, they can be detected and recognized from the minor and often overlooked movements of the facial muscles. Micro expressions contain the true information about the human's emotion. In a body of work presented by **[3] Anna D. Sergeeva, Alexander V. Savin, Victoria A. Sablina, and Olga V. Melnik, titled "Emotion Recognition from Micro-Expressions: Search for the Face and Eyes"** they have proposed the use of 2 algorithms. They proposed the Viola – Jones [7] which was suggested in the year 2001 by Michael Jones and Paul Viola. In approach the Haar features are utilized with the objective of detecting the face of a person, and detect their eyes as well. At present day, this method is the most optimal in terms of execution time and resource utilization. The second algorithm that was proposed mainly based itself upon on the hue and features of the skin in the color space RGB along withYCrCb. This method includes many stages of image processing as well as filtering. The test results show that both the algorithms are competent in the preliminary processing in the image which is focused on the identification of micro expressions. The algorithm based on hue, lighting, and intensity was deemed appropriate for images that had color and was also time efficient for images of smaller size. The Voila-Jones on the other hand can be utilized with images that are or have been converted to grayscale and also for images that have a greater resolution resolution.

[D]

For many computer vision algorithms classification of human emotions is a crucial and elusive task. Especially today, which is the age of humanoid robots and AI, and they co-exist with the humans. In a research done by **[4] Ivona Tautkute1, Tomasz Trzcinski, and Adam**

Bielski named "I Know How You Feel: Emotion Recognition with Facial Landmarks" they propose a new way of emotion recognition that is dependent of incorporating facial landmarks as a crucial element of classification function. They have also used Deep Alignment Network (DAN) in order to achieve very accurate results in the recognition of facial landmarks. Their approach uses Deep Alignment Network architecture, which was originally proposed for robust face alignment. Its main advantage is that it uses iterative process in comparison to its contemporary methods, in order to adjust the location of facial landmarks. The iterations are included into the Neural network architecture for learning, and the knowledge acquired is passed onto the next stages through the use of facial heatmaps. It can be said that DAN can deal with entire facial images and not pixels or patches. DAN is currently ranked third in terms of facial landmark recognition leaderboard. In this body of work the researchers employed their recognition model as a part of the in-car analytics system to be used in self-driving cars or vehicles. A self-driving car's operation can be altered with the emotion of the passenger, for e.g.: fear of speed detected on the passenger's face. This is an ongoing method of emotion recognition that enables the utilization of facial landmarks. This research uses the JAFFE datasets in order to produce the results and shows that there are still improvements to be made. However, this method has been said to have the potential to outperform the existing or other proposed methods. This method can be improved with the incorporation of attention mechanism on facial landmarks.

[E]

In most of the cases and model we see the face detection system's accuracy is greatly reduced if the user is wearing anything on the head or the face such as a big headphone, or sunglasses, or even normal glasses. So, in order to overcome this deficiency, in a work proposed by, **[5] Hwanmoo Yong, Jisuk Lee, and Jongeun Choi, titled "Emotion Recognition in Gamers Wearing Head-mounted Display"**, they trained the Convolutional Neural Network to detect and classify the emotion of a human wearing a Head Mounted Gear. This system does not take into account the eyes and the eyebrows of the users and solely focuses on the remaining parts such as the mouth and other facial muscles on the forehead and the cheeks. Apparently, there has be no prior research in this field. They used the Roundabout Face Datasets, which consists of 8040 images, with 8 facial expressions representing the emotions like happy, sad, angry, neutral, disgust, surprise, fear, contempt. Each emotion was analyzed and displayed with multiple gaze directions and varying angles of the camera. The intentionally pruned the original images around the face and added black rectangles around the eyes and the eyebrows in order to represent Head Mounted Gears. Later the 8040 images were randomized and jumbled, then splitting into 3 datasets, 5640 in order to perform the training, 1200 in order to conduct the validation and the remaining 1200 images for the test. The result of this experiment was that, 3 CNN were successfully trailed to determine the emotion from a fractionally masked human face. From the three DenseNet showed better functioning than Inception ResNet V2 and ResNet. However, for the recognition of fear and disgust ResNet was more effective. It can be deduced that the CNN was able to derive features from the bottom parts of the face as well, which aren't considered as an embodiment of a person's emotion. The CNN was able to predict the emotions such as Surprised, Fearful, Happy, Disgusted emotions with greater accuracy as expression of these emotions require the movement of mouth, while the eyes and eyebrows play a more poignant role in the representation of the remaining emotions.

[F]

Emotion recognition is natural trait of human beings which is an area of interest for researchers. Speech is a part containing carrying human emotions or state of mind. Speech is a mixture of utterances. A paper prepared by **[6] Gustavo Assuncão, Paulo Menezes and Fernando Perdigão on "Importance of speaker specific speech features for emotion recognition"** analyses the human speech and extracts emotion from it in real-time. In this demonstration a real time emotion recognition system was proposed which extracts speech features to build state-of-art classifiers. This paper has added more interest to the researchers working on the extraction of human based emotions using advanced technology.

[G]

Basically, emotions are categorized into six types anger, happiness, surprise, disgust, fear, sadness and neutral. Extraction of emotions is a very vast concept naturally possessed by humans. It is very hard for the machines to determine the emotional state of humans. For this human have attempted to build Speech Recognition System (SER) that extracts emotions based on the properties it is build. These systems are proved to be fruitful in teaching and learning technologies, medical services etc. This reduces the deployment of human resources and expenditure brought along with it.

In this paper proposed **by [7] Surekha Reddy B, T. Kishore Kumar on “Emotion Recognition of Stressed Speech using Teager Energy and Linear Prediction Features”** determining of emotions is done using Teager Energy Operator (TEO) and Linear Prediction Coefficient (LPC).

This paper focuses in Stressed Speech Emotion Recognition (SSER). Gaussian Mixture Model (GMM) classifier is used for categorization of emotions. Stressed emotions anger, fear, disgust and sadness are classified taking neutral as the reference. The accuracy observed was 82.7% and 88% which are higher than the previous systems. But system observed some difficulty in recognizing the fear emotion. So, some improvements are to be done the system.

[H]

Music is a form of art that carries emotions. They are like rivers that flows in different directions. Everyone has lows and highs in daily life. To rejoice and rejuvenate music is a reliable companion. It would be very effective and handy if Music players would play according to our mood. They are also helpful for the transition of emotions also. It is also stress reliever. This Emotion based player plays songs according to the accordance to the person’s mood. Recognizing human emotions is a very difficult task because there are variations in emotions of human. Many things come into act while bringing human emotions as genre, pitch, amplitude etc. are also needed. Many works needed to be done like classifiers need to be built, visualization of musical features and finally mapping the features and recommending songs.

This paper proposed by **[8] S. Deebika, K. A. Indira, Dr. Jesline on topic “A Machine Learning Based Music Player by Detecting Emotions”** focuses on music player based on emotions of the listener. It uses Convolutional Neural Network (CNN) determining the human neural conditions. Although many algorithms are there but their result is not as predicted, CNN overcomes such gap between them. First, they train the system to make the visualization more informative. Then, Music and songs are classified by Support Vector Machine (SVM). Finally, recommends to the user. This paper enhances the accuracy and efficiency of previously existing systems. Songs are classified using different filters. The basic purpose of the system is to alter or prolong the existing emotional state. This automation is used for differently abled people. This concept can increase the efficiency and correctness of the system.

[I]

Speech has been the most effective way of communication between human beings. As huge advancement took place in machines years ago but human-machine interaction has been one of the hardest things. There is variation between voice, accent etc which makes it even hard to understand human language. Many systems have developed so far but they do produce result as estimated. Many algorithms like Pattern Recognition Neural Network (PRNN) and K-Nearest Neighbor (KNN) are used in the previous systems.

The paper proposed by **[9] J. Umamaheswari, A. Akila on "An Enhanced Human Speech Emotion Recognition Using Hybrid of PRNN and KNN"** depicts the use of hybrid algorithm of Pattern Recognition Neural Network (PRNN) and K-Nearest Neighbor (KNN).

Mel Frequency Cepstral Coefficient (MFCC) and Gray Level Co-occurrence Matrix (GLCM) were used in this system for feature extraction. Weiner filter was used for filtering the noise in speech. Feature extraction is done from the emotional database containing emotional classes Angry, Happy, Sad, Neutral, Surprise and Fear. Many visualization methods for accuracy and precision were taken into consideration and various tables and columns were prepared. Gaussian Mixture Model (GMM) and Hidden Markov Model (HMM) like speech recognition techniques were used in the system.

[J]

World population is growing in a very rapid manner. With time there are people who needs to be cared like children, elderly people, differently abled people. In this busy world taking care of these age groups has become challenging. But kudos to automation and robotics many humanoid service robots are developed worldwide. But as we know there is a huge communication gap between humans and robots. There is limitation to the communication of robots. Many easy and feasible ways are already developed face recognition is one them. To track the visual gestures of the human's robots, need to be trained. If face recognition can make human-robot communication more of human-human interaction then it is feasible to serve more effectively. Humanoid robots are the most preferable type of care givers.

This paper proposed by **[10] T. M. W. Vithanawasam and B.G. D. A. Madhusanka on "Face and Upper-Body Emotion Recognition Using Service Robot's Eyes in a Domestic Environment"** is centered on the idea of face recognition and upper body recognition. According to research ,55% of interaction is body language. For robots' eye model web cam and other high-resolution cameras were used and focused on the Region of Interest (ROI) to get the result.

Face detection used Haar-cascade classifier used because of its high-level features. OpenCV is platform used for face detection. But for face detection many constraints should be fulfilled like looking into camera etc.

For upper body detection, visual feed of upper-body ROI. The proportionality of the body can be using head's height and head's width. To get more accurate features and emotions huge datasets for every emotion is required. Many obstacles like lighting, position was taken into consideration to build an emotional recognition system. Many emotional expressions like fear, anger was recognized by the system. Despite of many hardships it can be used for the references of more advanced systems.

In the following Table 2.1, we group the main approaches for image colorization in the reference papers and describe important features of each approach

Table 2.1: Summary of Literature Survey

Paper	Algorithm	Dataset
Daly et al (2016)	VGG16NET	ImageNet
Hwang et al (2016)	Learning Pipelines	ImageNet
Zhang et al (2016)	CNN + Regression Net	ImageNet
Baldasarre et al (2017)	Inception-ResNet v2	CIFAR-10
Welsh et al (2012)	Global Image Matching	CIFAR-10
Levin et al (2014)	Quadratic matching	CIFAR-10
Nie et al (2007)	Polynomial matching	ImageNet
Chen et al (2014)	Digital Matting	ImageGarden
Kekre et al (2010)	Color Traits Transfer	ImageNet
Horiuchi et al (2013)	Pseudo-coloring	ImageNet

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

The automatic facial expression recognition system (AFERS) that exist today adopt a three-stage processing: face detection followed by facial feature extraction and finally representation, and facial expression recognition. The purpose of the face-detection stage is to automatically locate the face region in a given input image. It is the first step in AFERS, and consequently its correctness and performance have a major influence on the efficiency, accuracy, and usability of the entire system. The face detector could work by distinguishing and recognizing faces frame by frame or just directly detect a face in the first frame and then track it in the subsequent images in a progression. Then the next step is to distil and portray the information about the facial expression to be recognized. A high-level description of the expression (as a function of the image pixel data) is formed as the result of the extraction process. This characterization is also known as the “feature vector” and is utilized for upcoming expression classification. Geometric characteristics which show the shape and relative positions of facial components such as the eyebrows, eyes, mouth etc. and spectral-transform-based features which are obtained by implementing image filters to face images are mostly used to depict the information of facial expressions. These data are converted to binary values and the emotion which has the binary values closest to the input is shown to the user.

3.2 Proposed System

In the work we have done, we have used the Haar Cascade Algorithm. Object detection using Haar cascade classifiers is one of the most accurate ways if effective object detection, this was originally proposed by Paul Voila and Michael Jones in their paper titled” Rapid object detection using a boosted cascade of simple features". It is a method that utilized machine learning where a cascade functions are trained to create a collection of positive as well as negative images. Positive images are images that contain the object that we would like the machine to detect and negative images are those that do not contain the object of desire. After such training is complete, it can be used to detect that particular object on other images and even real time scenarios, similar to our work.

In the Haar algorithm we use, it needs many positive images and negative images first so they can be used to train the classifier. After that comes feature extraction from it. The image is weighed in terms of pixels, each feature is a single value gained by deducting the total of pixels within a particular(focused) area from the sum of pixels under the remaining area in the image. This can be simplified as such; the feature value is attained by deduction of total pixels covered in white from the total pixels in black

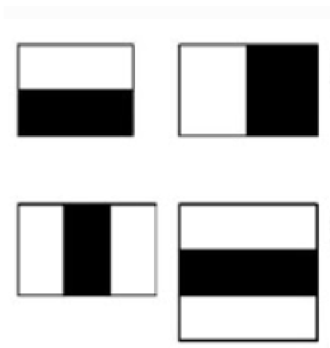

Fig 3.1: Haar cascade training images

With this method the algorithm can extract enough features. However simply doing this with every pixel would mean a huge amount of computing creating an overload. For e.g.: a 24 x 24 window would result in over 160000 features which would be too much for the computer and also it would take an enormous amount of time to do the same for every single image. So, in order to solve this concept of integrated images was used. Regardless of the number of pixels it will simplify the calculation which would only involve 4 pixels. This would make the whole process efficient as well as faster. But how is the system supposed to choose the most reliable features out of thousands of in the image? This can be done with the help of AdaBoost.

AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm. It can be used along with other kinds of learning algorithms to increase the overall efficiency and accuracy. The AdaBoost algorithm is adaptive as the subsequent weal learners are tweaked in favor of those instances misclassified by previous classifiers.

In the system we are using, it determines the best threshold which can sort the images, face images in this case, into positive and negative. There might be many misclassifications and errors, so we will choose features with the least rate of error. This means that they are the most reliable features in order to sort a face image from a non-face image. The final classification is a sum of the weak classifiers. Weak classifiers are those that alone cannot classify the image, but they can with the help of other strong classifiers. So, in theory, we can take an image of 24x24 window and apply a set of 6000 features, and determine if it is a face or not. However, this would take too much time, so we have a crafty solution for this. As we know, in most images there are a lot of non-face regions, and when they're detected it is just more efficient to discard them totally and focus on other remaining regions. By doing this no time is wasted on the analysis of a non-face region. So, in order to do this the concept of Cascade classifiers was brought in. Instead of applying all the 6000 features on a window, the features will be sorted into a number of stages and classification and detection is performed on them one-by-one. If the process is not successful at the first phase it is discarded and the remaining features on it are not considered at all. For example: if no eye is detected then the remaining features are not checked and are directly discarded. However, if the initial phase is successful, the second stage of features is applied and the process goes on. If a data (image) is able to pass all the phases it is classified as a positive image i.e. the object we are trying to detect, a face in this case.

For the classification of emotion, CNN pre-trained model is used, The CNN is trained using the Gradient decent algorithm. These are found in the form of XML files, the one we will use to detect the face is called the "haarcascade_frontalface_default.xml".

After recording the images in real-time they need to be converted into grayscale and many functions that we use in OpenCV are compatible with images in grayscale. We also want to draw a rectangle around the face of the person in real-time so that they would know that their face is being detected.

We want a system that not only recognizes the face of a person but is also capable of classifying the emotion on the person’s face. For the detection of the face we have implemented the HAAR cascade algorithm which can be used to create a Haar file, which is then imported into the python program. The system should be used under good lighting conditions and reasonable distance from the camera of the computer. Drastic variation in lighting and pose might cause inaccuracies in detection and/or classification. After the classification of emotion, we can play the suitable music track.

3.3 Advantages of Proposed System

Here we propose to use a cascade classifier, and the main advantage of cascading is that it achieves a good performance for both accuracy and time-complexity, even when you have a number of weak classifiers. The weak classifier must work at least with 51% accuracy. The cascade classifier rejects the many of samples in first node classifier with an efficient time. It is able to do so because it will reject an image if it doesn’t detect the necessary features in the early phases itself i.e. the features will be sorted into a number of stages and classification and detection is performed on them one-by-one. If the process is not successful at the first phase it is discarded and the remaining features on it are not considered at all. This will make classifier faster and more accurate.

In this project we want to create a system that not only recognizes the face of a person but is also capable of classifying the emotion on the person’s face. For the detection of the face we will implement the HAAR cascade algorithm which can be used to create a Haar file, which is then imported into the python program. For the classification of emotion, CNN trained model will used, The CNN is trained using the Gradient decent algorithm.

The system should be used with proper lighting and at a reasonable distance from the camera of the system. Drastic variation in lighting and pose might cause inaccuracies in detection and/or classification. After the classification of emotion, the system will automatically play the suitable music track.

The main objective of the system is to reduce the communication gap between the human and the computer, as the user doesn't have to give any deliberate input this makes the communication process convenient.

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

Our system architecture is shown in Figure 4.1 The process begins with the Input of the image which is the followed by image pre- processing. The processed image is now ready for the cascade classifier.

The classifier is trained using a dataset with a lot of images. Features of a particular emotional state are extracted from the images in the dataset which train the CNN.

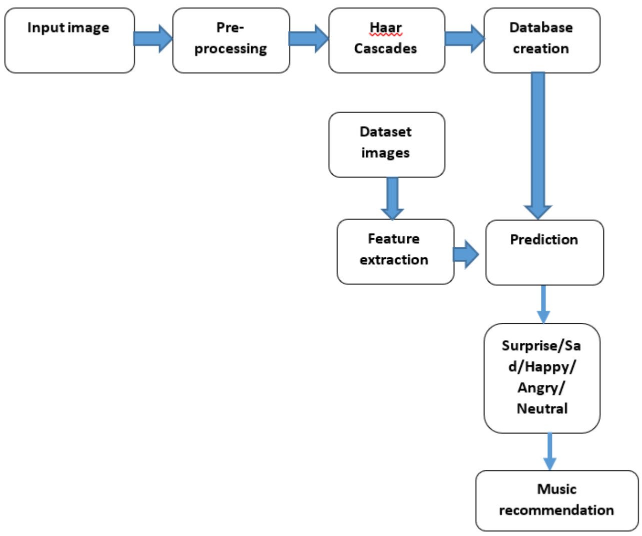

Fig 4.1: System Architecture

4.2 Cascade Classifiers

The regular cascade classifier is the very efficient in face detection. Generally, a lot of object identification tasks with fixed structure can be attended to by means of this method, not limited to face detection. The cascade classifier is a tree-based technology, in which Haar-like features for human face detection. The Haar-like characters can be utilized with all scales in the accentuated classifier and can be rapidly computed from an integral version of the image to be detected in.

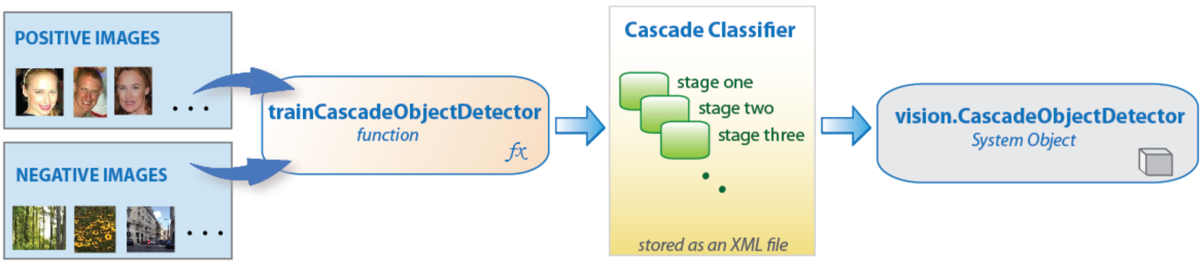

Fig 4.2: Working of Cascade Classifier

The cascade classifier is an amalgamation of a number of stages, where each stage is a band of weak learners. The weak learners are simple classifiers also known as **decision stumps.** Each stage is prepared using a technique called boosting. **Boosting** gives it the capacity of training a highly accurate classifier by taking a weighted average of the judgement made by the weak learners.

Each phase of the classifier tags the area defined by the instant location of the sliding window as either positive or negative. **Positive** implies that an object was found and **negative** implies that no objects were found. If the tag is negative, the classification of this region is accomplished, and the detector slides the window to the next location. The region is passed to the next stage if the tag is positive as the classifier has passed. The detector reveals an object spotted at the current window location when the final stage classifies the region as positive.

The phases are constructed so as to rebuff negative samples as fast as possible. The supposition is that the majority of the region does not contain the object of interest. On the other hand, true positives are not very frequent and worth taking the time to verify.

- A **true positive** transpires when a positive sample is correctly segregated.
- A **false positive** transpires when a negative sample is mistakenly classified as segregated.
- A **false negative** transpires when a positive sample is mistakenly classified as segregated.

For efficient functioning each phase in the cascade must have a low false negative rate. If a phase falsely labels an object as negative, the segregation is halts, and you cannot amend the error. However, every phase can have a great false positive rate. Even if the detector falsely marks a non-object as positive, the mistake can be appended in subsequent stages. Including more stages curtail the overall false positive rate, but it also diminishes the overall true positive rate.

A set of positive samples and a set of negative images is required by the classifier for training. A set of positive images with regions of interest must be provided which are then used as the positive samples. The image Labeller can be used in order to mark the object of interest with bounding line. The output of the Image Labeller is a table that can be used for positive samples. A set of negative images is also necessary so that the function is able to generate negative samples automatically. To attain a satisfactory detector accuracy, the number of stages, feature type, and other function parameters must be set.

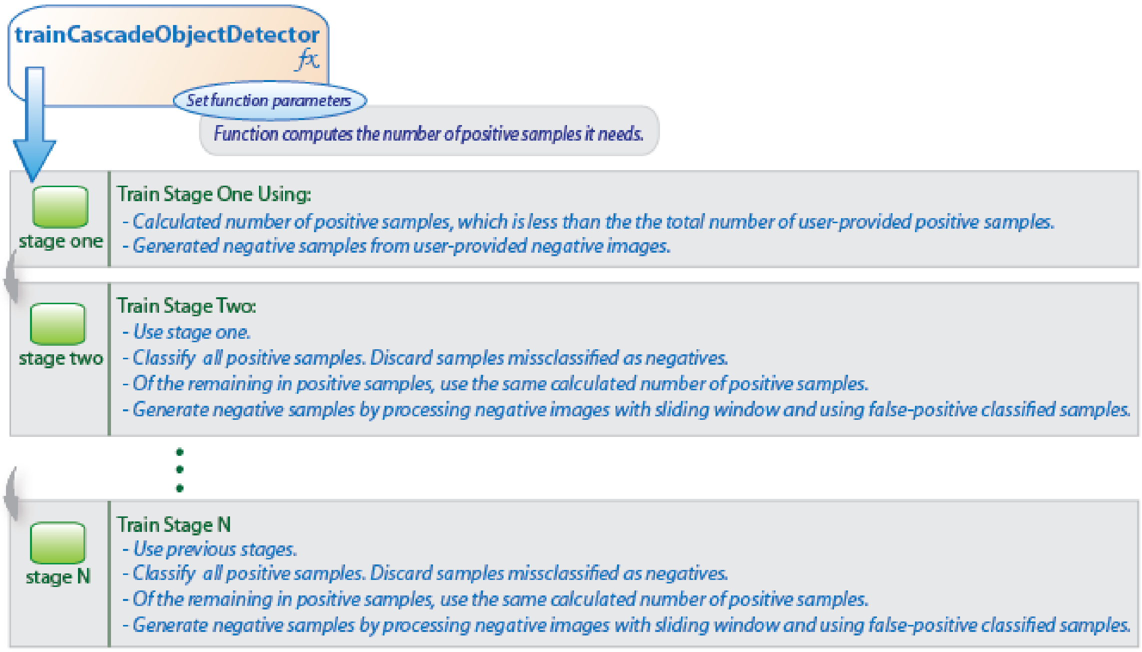

Fig 4.3: Training stages of Cascade Classifier

4.3 Neural Networks

A neural network is an interconnection of neurons, or in a contemporary sense, an artificial neural network, made up of artificial neurons or nodes. Thus, a neural network is either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problems. The association of the biological neuron is considered as weights. An excitatory connection is caused by a positive weight which implies that an inhibitory connection is caused by a negative weight. The inputs are altered by a weight and tallied. This is a process called as linear combination. The magnitude of the output is governed by an activation function. For instance, range of output between 0 and 1 is considered as acceptable, or it could be −1 and 1.

Predictive modeling, adaptive control and applications are major areas where neural networks can be used as they can be trained via a dataset. Self-learning which can be a result of experience can occur within networks, which can extract inferences from a complex and seemingly unrelated set of information.

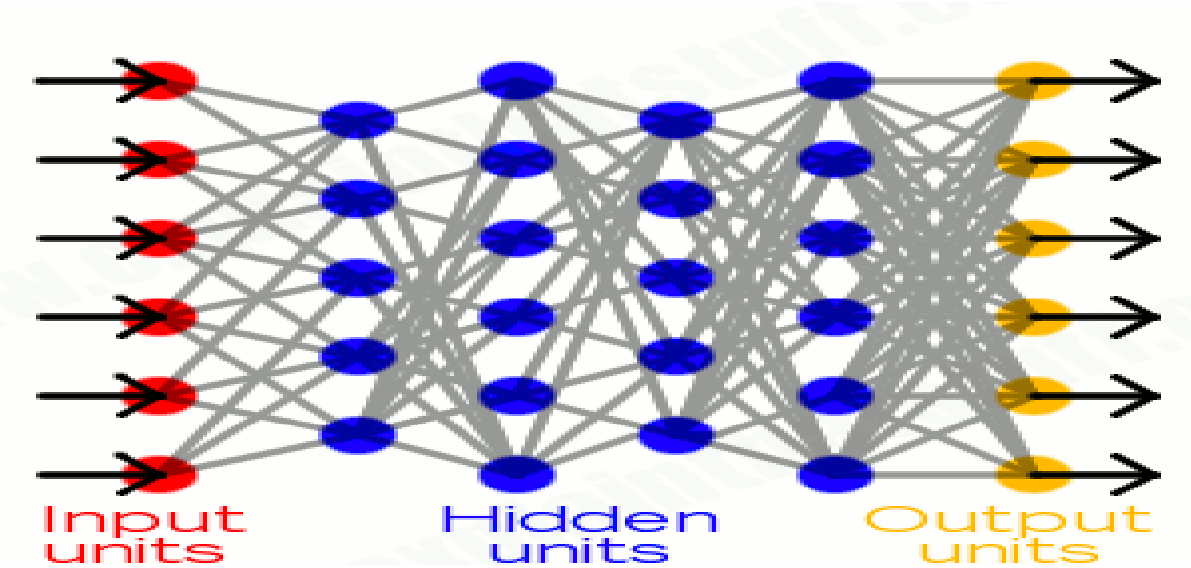

Fig 4.4: Neural Network

4.3.1 Classification

All classification tasks hinges on labeled datasets; i.e. for the neural network to recognize and learn about the interrelationship between the labels and data humans must transfer their knowledge to the. This is known as supervised learning.

- Spot faces, distinguish people in images, perceive facial expressions (angry, joyful)
- Spot and distinguish objects in images (stop signs, pedestrians, lane markers…)
- Notice gestures in video
- Notice voices, identify speakers, write out speech to text, notice sentiment in voices
- Distinguish text as spam (in emails), or counterfeit (in insurance claims); notice sentiment in text (customer feedback)

Any labels that a person can produce, any results that you are concerned about and which correlate to data, can be used to train a neural network.

4.3.2 Clustering

Clustering or grouping is the apprehension of resemblances. Labels are not required by deep learning to detect similarities. Unsupervised learning is when the learning is done without labels. Most of the data in the world is unlabeled. A fundamental of machine is that for an algorithm to be more accurate it is required to train on more and more data. Therefore, unsupervised learning has the capability to generate highly accurate models.

- Search: Contrasting documents, photos or audio files to surface similar items.
- Anomaly detection: The flipside of detecting similarities is detecting anomalies, or unusual behavior. In many instances, uncommon behavior correlates highly with things you want to detect and prevent, such as fraud.

4.4 UML Diagrams

UML (Unified Modeling Language) is used to visualize the model before development of the project. It makes easier to develop a project and eliminate the possible threats. There are different types of UML diagrams. We have drawn two of them. Possible flow the system can be seen in the below diagrams. State diagram shows the transition of the different states in the system. Activity diagram shows the tasks taking place between user, system and the end result. We produced the UML diagrams using Star UML software. The general working of the system can be seen from the below diagrams.

4.4.1 State Diagram

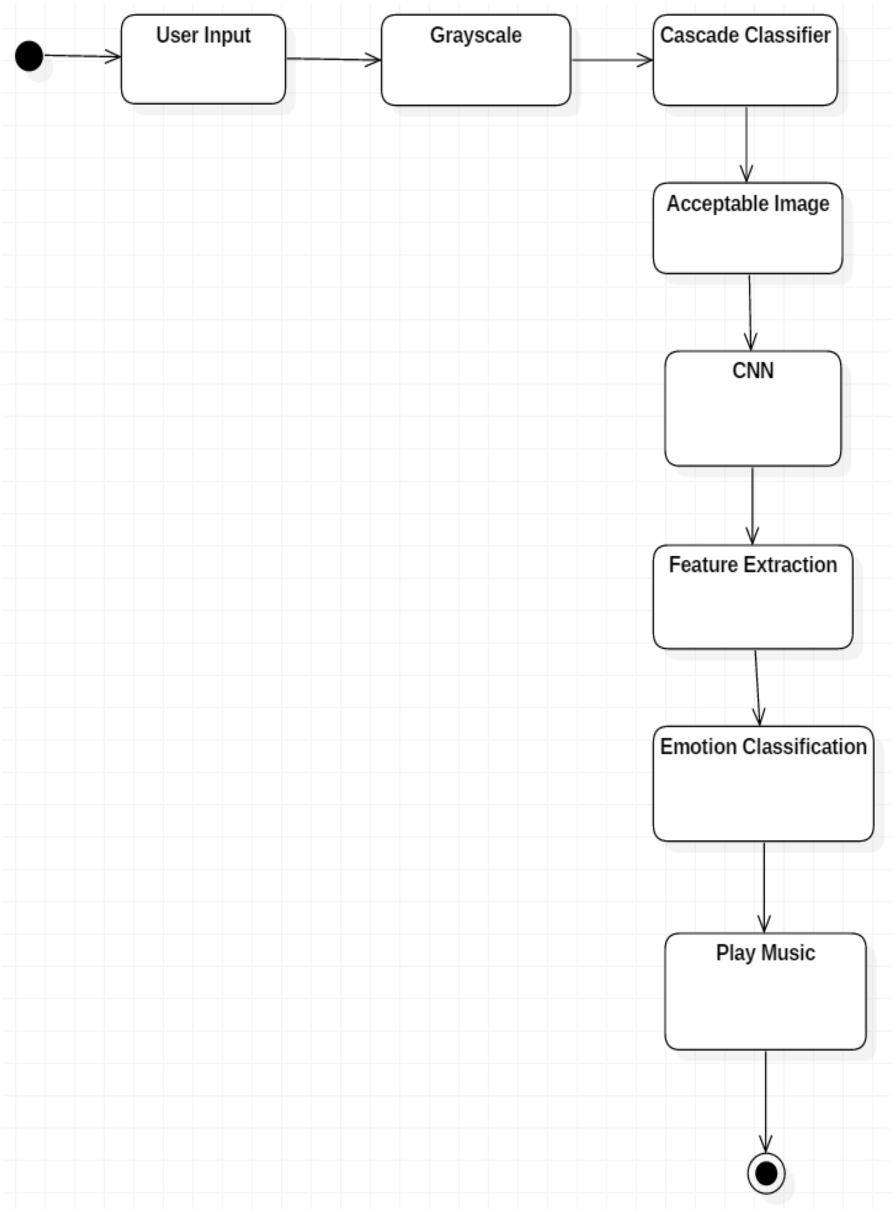

Fig 4.5: State Diagram

4.4.2 Activity Diagram

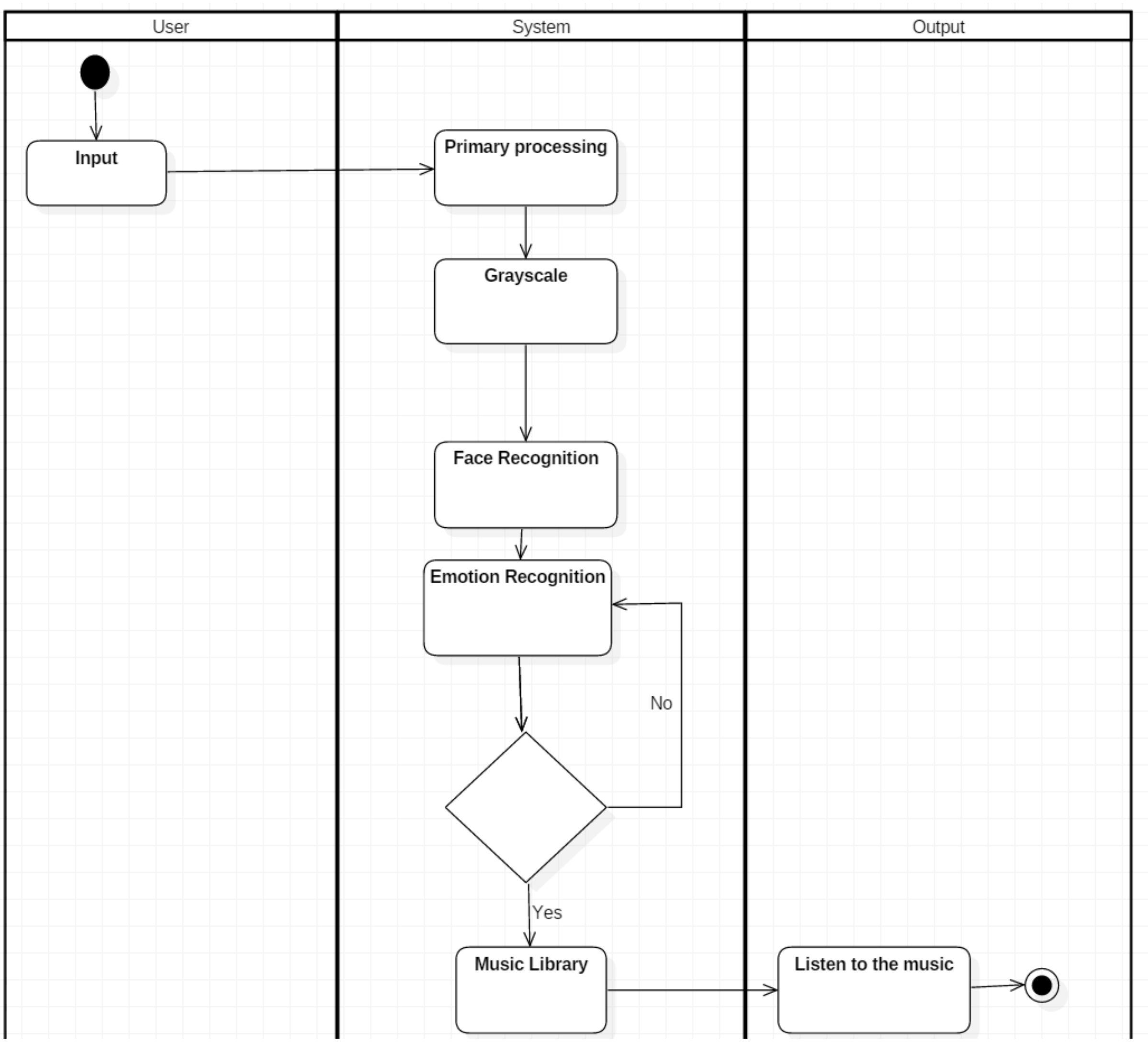

Fig 4.6: Activity Diagram

Music Recommendation system is fully based on the human emotion. It is very difficult to detect the exact human emotions. Emotions may vary with difference in predicament of the object. In this system, Convolutional Neural Network (CNN) is used to extract the human emotion. After extracting the human emotions, the music can be played accordingly.

CHAPTER 5

SYSTEM REQUIREMENTS AND SPECIFICATIONS

5.1 Hardware Requirements

Any normal home PC (laptop or desktop) will suffice as a client as long as it can compile and run python files and has a relatively decent internet connection. i.e. The minimum requirements are:

- 2GB RAM
- 8GB HDD space

5.2 Software Requirements

5.2.1 Windows OS

Windows is a group of several proprietary graphical operating system families, all of which are developed and marketed by Microsoft. Each family caters to a certain sector of the computing industry. Active Microsoft Windows families include Windows NT and Windows IoT; these may encompass subfamilies, e.g. Windows Server or Windows Embedded Compact (Windows CE). Windows 10 is a series of operating systems produced by Microsoft and released as part of its Windows NT family of operating systems. It is the successor to Windows 8.1. One of Windows 10's most notable features is its support for universal apps, an expansion of the Metro-style apps first introduced in Windows 8. Universal apps can be designed to run across multiple Microsoft product families with nearly identical code including PCs, tablets, smartphones, embedded systems, Xbox One, Surface Hub and Mixed Reality.

5.2.2 Python 3.7

Python is a language that has become widespread and especially popular among machine learning communities due to its simplicity and powerful libraries. It was written by Guido van Rossum in the year 1991 for helping C/C++ module integration, but has quickly become an all-purpose language which has support for thousands of modules and plugins openly available. We also chose Python for our project because of the simplicity of the syntax and tight integration with many machine learning and image processing libraries (Python, 2016). We had also considered using R, but the advantages of using R over Python were overcome by its module install difficulties. Here, we are using Python 3.7.3 which is the latest stable version of Python 3 that is available at this point in time.

5.2.3 Anaconda (Python Distribution)

Anaconda is a distribution of Python, which comes preloaded with all the tools that are required to do complex scientific calculations and metric processing. Some of the modules that come pre-installed with an up to date anaconda package are NumPy, scikit-image, scikit-learn and so on. We have used anaconda for building our tensor-flow dependencies easily. Anaconda is used in the StackOS Linux distribution but is supported on Windows, Linux and Mac OS.

5.2.4 TensorFlow

TensorFlow is an open-source machine learning library that is used by developers and enterprises for the ML and deep learning needs. TensorFlow is a big package to install and easily exceeds 500MB with all the plugins and mappers. TensorFlow provides the easy to use high level Caffe and Keras APIs for easing machine learning model creation and training. TensorFlow can be run on the CPU as well as the GPU, which can really speed up machine learning tasks (Ranier, 2016).

5.2.5 OpenCV

OpenCV (Open Source Computer Vision Library) is one of the popular computer vision and machine learning software libraries. OpenCV was constructed to equip a common framework for computer vision applications and to expedite the use of machine perception in the profit-making products. OpenCV is a BSD-licensed product and helps businesses to utilize, tweak and customize the code.

OpenCV consists of over 2500 optimized algorithms, which consists of a broad set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to notice and distinguish faces, identify objects, segregate human actions in videos, track camera movements, track objects in motion, distill 3D models of objects, create 3D point clouds by using stereo cameras, merge images together to generate a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images captured using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV boasts a user community of more than 47 thousand people and estimated number of downloads exceeding 18 million.

5.2.6 NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
It contains among other things: a powerful N-dimensional array object, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities.

5.2.7 Keras

Keras a popular and widely used open-source neural-network library typed in Python. It is proficient in running on Microsoft Cognitive Toolkit, R, Theano, TensorFlow, or PlaidML. Constructed to facilitate a reliable and high rate of experimentation with deep neural networks, it focuses on being modular, easy to use, and expansible. It was invented as a part of the research effort in the project named ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its leading author and maintainer is François Chollet is also the creator of the XCeption deep neural network model.

5.2.8 Haar File(haarcascade_frontalface_default.xml)

The “haarcascade_frontalface_default.xml” is a Haar cascade designed by OpenCV to detect the frontal face. A Haar Cascade works by training the cascade on thousands of negative images with the positive image superimposed on it.

5.2.9 Emotion recognition model (_mini_XCEPTION.102-0.66. hdf5)

A huge dataset has been used in order to train this face recognition model. In fields ranging from gaming to marketing to healthcare facial expression recognition has been a topic of great curiosity. The objective is to classify images of human faces and categorize them into one of seven basic human emotions. We use Convolutional Neural Network (CNN) model. As CNNs use a large number of filters they are better equipped to capture special features in the input image, and this makes them better than any other for image recognition.

What makes are project unique is that is can be run from any home desktop or PC with average specs. This is because the software required for the project are free to download from the internet. Python can also be run from USB in demo mode or on a virtual machine over a Windows environment. The dependencies can be satisfied by a copying the emotion files into the anaconda environment. Most of the important machine learning and data science libraries and templates are already bundled in the Anaconda Distribution. When it comes to hardware dependencies any average computer running a windows OS will work. The computer must have a working webcam.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Algorithm Explanation

6.1.1 Haar Cascade Algorithm

In the work we have done, we have used the Haar Cascade Algorithm. Object detection using harr cascade classifiers is one of the most accurate ways if effective object detection. It is a method that utilized machine learning where a cascade functions are trained to create a collection of positive as well as negative images. Positive images are images that contain the object that we would like the machine to detect and negative images are those that do not contain the object of desire. After such training is complete, it can be used to detect that particular object on other images and even real time scenarios, similar to our work.

After that comes feature extraction from it. The image is considered in terms of pixels, each feature is a single value obtained by subtracting the sum of pixels within a particular(focused) area from the sum of pixels under the remaining area in the image. This can be simplified as such; the feature value is obtained by subtraction of sum of pixels covered in white from the sum of pixels in black. With this method the algorithm can extract enough features.

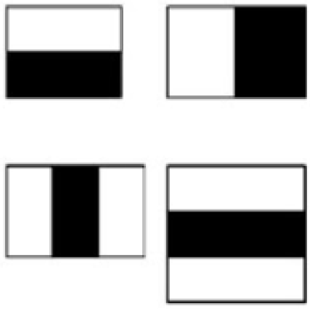

Fig 6.1: Haar Cascade Training Images

The features will be sorted into a number of stages and classification and detection is performed on them one-by-one. If the process is not successful at the first phase it is discarded and the remaining features on it are not considered at all. For example: if no eye is detected then the remaining features are not checked and are directly discarded. However, if the initial phase is successful, the second stage of features is applied and the process goes on. If a data(image) is able to pass all the phases it is classified as a positive image i.e. the object we are trying to detect, a face in this case.

In our experiment we are using OpenCV in order to detect the face as well as the classification of emotion of the person's face. OpenCV contains many pre-trained classifiers in order to detect face through features such as the eyes, mouth etc. These are found in the form of XML files, the one we are using to detect the face is called the "haarcascade_frontalface_default.xml". The first task to perform is to load the xml classifier, and then load our input images, which are taken in real-time through the webcam, into grayscale.

The code for this is as such:

```
import cv2, sys, numpy, os
haar_file = 'haarcascade_frontalface_default.xml'
datasets = 'datasets'  #All the faces data will be present this folder
sub_data = 'neutral'
```

After this initial phase we have to use the webcam in order to record the persons face, the code we have used for this purpose is as such:

```
while count < 100:
    for i in range(0,2):
        if i==0:
            ##WEBCAM
            (_, im) = webcam.read()
```

After recording the images in real-time they need to be converted into grayscale and many functions that we use in OpenCV are compatible with images in grayscale. We also wanted to draw a rectangle around the face of the person in real-time so that they would know that their face is being detected. The code for this is as such:

```
gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray, 1.3, 4)
for (x,y,w,h) in faces:
    cv2.rectangle(im,(x,y),(x+w,y+h),(255,0,0),2)
    face = gray[y:y + h, x:x + w]
    face_resize = cv2.resize(face, (width, height))
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(im, 'F A C E',(x+5,y-10), font, 0.5, (0, 255, 0), 2, cv2.LINE_AA)
```

6.1.2 Gradient descent algorithm

This algorithm is used for training the Convolutional neural network for emotion recognition and classification.

The objective of our algorithm is to minimize this cost function, for this purpose we find a set of weights and biases to make it as small as possible. And the primary method to achieve this goal is an algorithm called Gradient Descent.

The neural network is a feed forward network which has one unit is working at a time. The input is always from the unit to the left. Input multiplied by the weight of the connection. The inputs accumulate and if the sum is greater than the threshold the unit to the right are triggered. Having a diverse range of training data results a more accurate system. For eg: face of different people with different expressions.

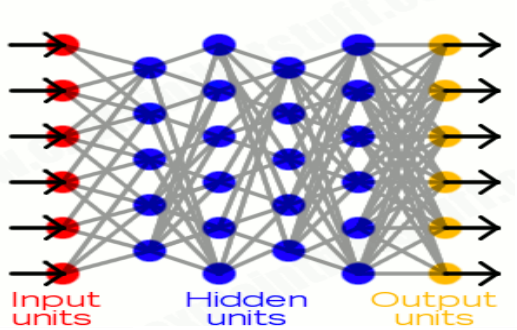

Fig 6.2: Neural Network

We define a cost function in order to calculate the error. This is fault in anticipating the correct output that our network has. The difference between the expected and the predicted output yields the cost function. As the output is closer to the real value the cost function become. Gradient decent algorithm's major objective is to minimize this cost function by finding a set of weights and biases to make it as minimal as possible.

$$\text{Cost function: } C(w, b) = \frac{1}{2n} \sum_x \|y(x) - a\|^2 \tag{6.1}$$

As seen in the graph

-function has a global minimum

-function has a global maximum

-where the function achieves the maximum

or minimum value that it can have

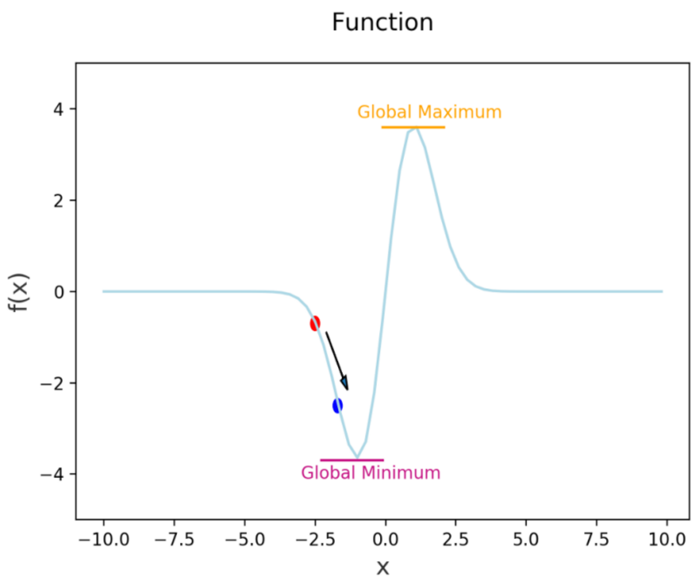

Fig 6.3: Gradient Descent Algorithm

$$\Delta C \approx \frac{\partial C}{\partial x} \Delta x + \frac{\partial C}{\partial y} \Delta y \qquad (6.2)$$

Δx change in the direction x

Δy change in the direction y

ΔC change in our function

Apply partial derivative:

$$\nabla C \equiv \left(\frac{\partial C}{\partial x} , \frac{\partial C}{\partial y} \right) \qquad (6.3)$$

Finally, we can say:

$$\Delta C \approx \nabla C . \Delta X \qquad (6.4)$$

The learning rate is the amount that we decide to move in any direction and it is what defines how fast we move in the direction of the global minimum. Choosing a very small number might mean that we must make too many moves to reach the point of global minimum. On the other hand, choosing a very big number might mean that we move past the desired point. Our objective is to minimize the cost function. Choosing an optimal learning rate is the challenge here. The weights and biases are updated continuously as the neural network is trained with images. This process is repeated.

6.2 Performance Evaluation

We followed the human 'Turing Test' methodology for testing our results. We ran the system 25 times and made it classify the emotion each time. Some of the result screenshots have been shown below.

Table 6.1: Turing Test

Number of correct guesses	Number of Incorrect guesses	'Turing Test' accuracy
19	25	76%

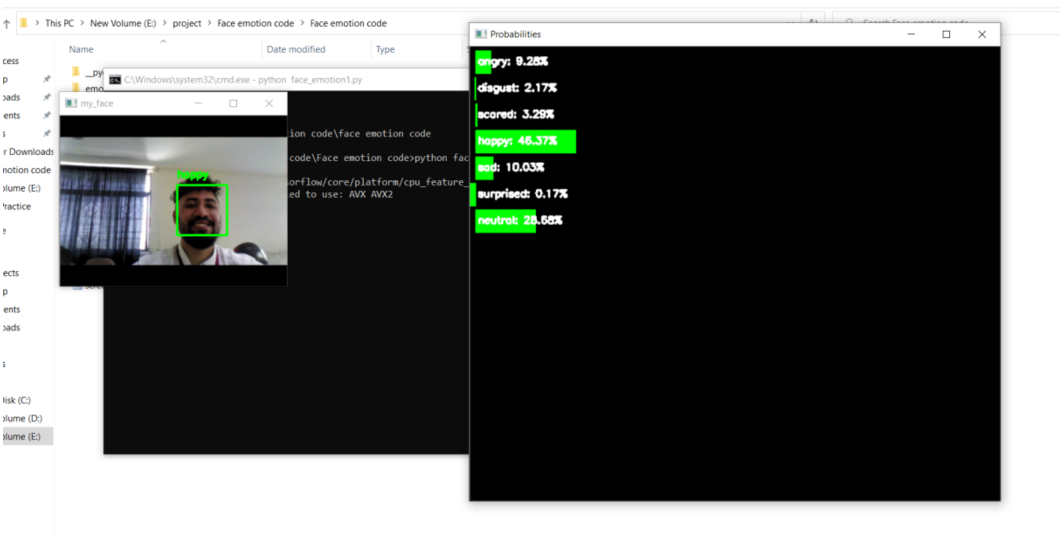

Fig 6.4: Result shows happy (Correct)

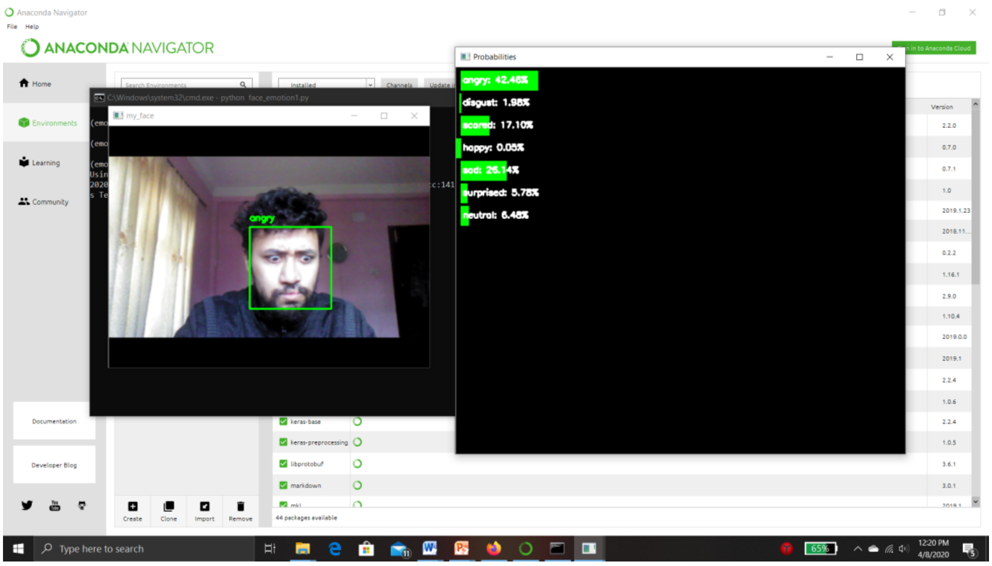

Fig 6.5: Result shows angry (Correct)

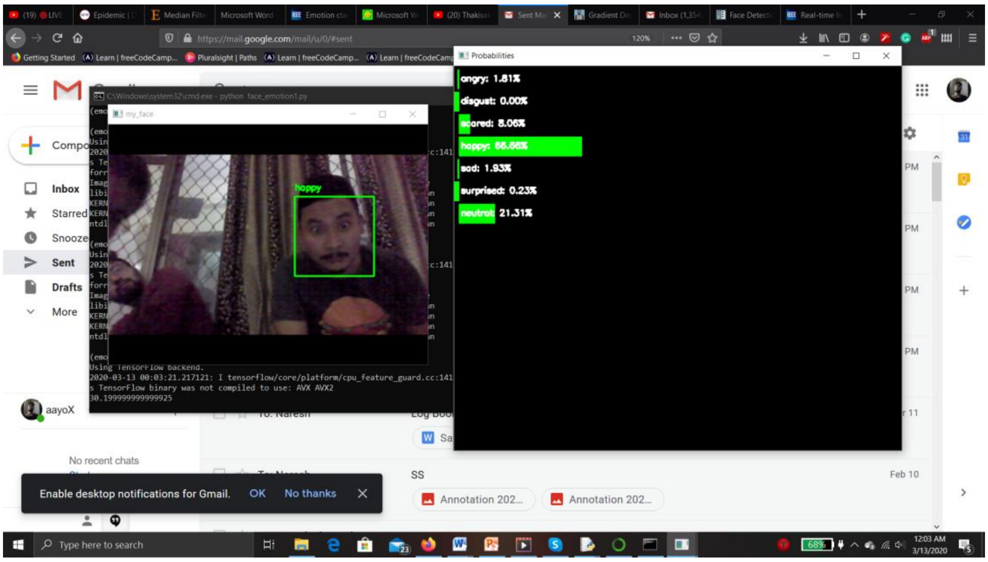

Fig 6.6: Result shows happy (Incorrect)

6.3 Implementation Screenshots

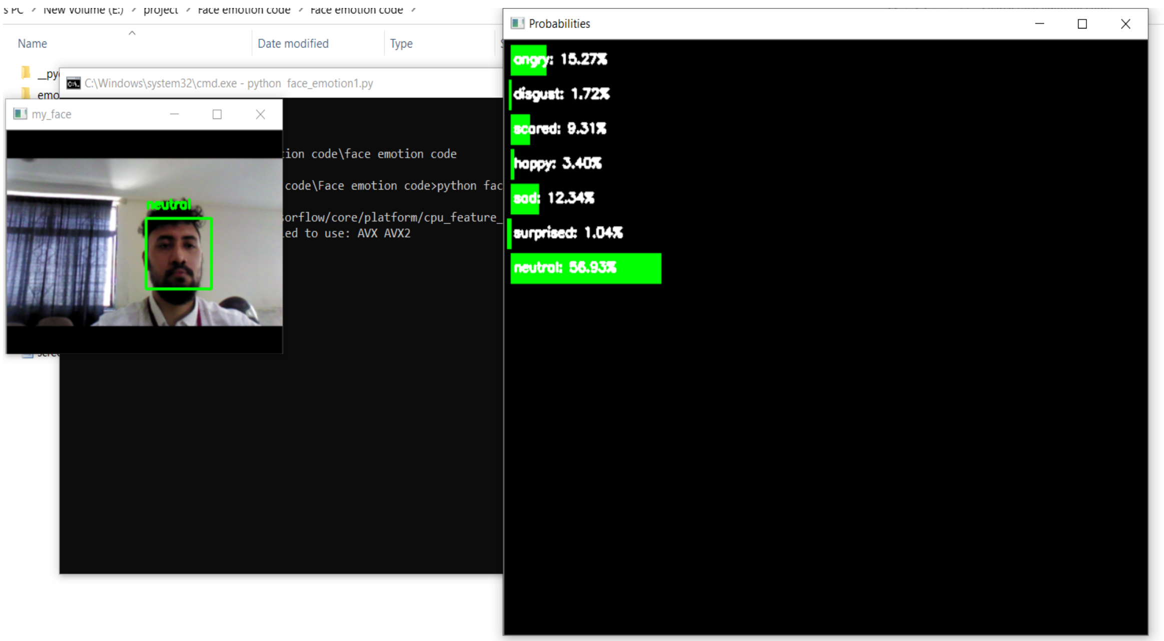

Fig 6.7: Implementation Results

Our system is based on the Haar cascade algorithm proposed by Paul Voila and Michael Jones in their paper titled” Rapid object detection using a boosted cascade of simple features” in tandem with the Convolutional Neural Network trained using the gradient descent algorithm.

The system is able to work accurately under good lighting conditions when the participant remains still. Under poor lighting conditions, however, the system could not detect face and even if it did it misclassified the emotion. Our system tries to forego the step of deliberate human input and tries to make the human computer interface faster and more convenient.

This being said, there are still a lot of improvements that can be made to our model, which we will discuss on about in the last chapter of this report.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

Emotions are an inseparable human interaction. Training a machine to recognize and take the emotion as input can greatly influence the human-computer interaction efficiency making the use of computers reliable. As we know, Music can greatly influence a person's emotional state. In the model we have used Haar-cascade algorithm in order to detect the user's face and Convolutional Neural Network is used to recognize and classify the particular emotion on the user's face. Advances in the architecture of CNNs used and optimization of the training methodologies will help improve the accuracy of the emotion detection and classification. Furthermore, promising techniques such as Deep Learning Super-Sampling (DLSS), Local Binary Pattern (LBP) algorithm combined with advanced image processing techniques such as Contrast Adjustment, Bilateral Filter, Histogram Equalization and Image Blending to address some of the issues hampering face recognition accuracy so as to improve the LBP codes, can greatly improve the accuracy of the overall face recognition system.

7.2 Future Enhancement

By use of technologies such as DLSS, LBP, as mentioned above the model could be greatly improved. From a paper written by Mohammad Faridul Haque Siddiqui, Dhwani Mehta, and Ahmad Y. Javaid it can be stated that LBP-kNN can recognize almost all AUs with high accuracy (>94%) while other methods show this level of accuracy only for few AUs (Action Units). One major thing to keep in mind is that the average accuracy of recognition of intensity of emotion goes down for all classes of emotions with an increase in intensity; however, LBP+SVM has yielded better and reliable results than other algorithms such as Gabor-SVM and HOG-SVM on average.

REFERENCES

[1] C. Gong, F. Lin, X. Zhou and X. Lü, "AMYGDALA-INSPIRED AFFECTIVE COMPUTING: TO REALIZE PERSONALIZED INTRACRANIAL EMOTIONS WITH ACCURATELY OBSERVED EXTERNAL EMOTIONS," in China Communications, **Volume 16**, (Number 8), pp. 115-129, Aug. 2019, doi: 10.23919/JCC.2019.08.011.

[2] Huang, Jian & Li, Ya & Tao, Jianhua & Lian, Zheng & Yi, Jiangyan. (2018). “END-TO-END CONTINUOUS EMOTION RECOGNITION FROM VIDEO USING 3D CONVLSTM NETWORKS”, pp6837-6841. 10.1109/ICASSP.2018.8461963.

[3] A.D. Sergeeva, A. V. Savin, V. A. Sablina and O. V. Melnik, “EMOTION RECOGNITION FROM MICRO-EXPRESSIONS: SEARCH FOR THE FACE AND EYES” 2019 8th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2019, pp. 1-4, doi: 10.1109/MECO.2019.8760029.

[4] Tautkute, Ivona & Trzcinski, Tomasz & Bielski, Adam. (2018) “I KNOW HOW YOU FEEL: EMOTION RECOGNITION WITH FACIAL LANDMARKS”, Tooploox Polish-Japanese Academy of Information Technology 3Warsaw University of Technology

[5] H. Yong, J. Lee and J. Choi, “EMOTION RECOGNITION IN GAMERS WEARING HEAD-MOUNTED DISPLAY” 2019 IEEE CONFERENCE ON VIRTUAL REALITY AND 3D USER INTERFACES (VR), Osaka, Japan, 2019, pp. 1251-1252, doi: 10.1109/VR.2019.8797736.

[6] G. Assunção, P. Menezes and F. Perdigão, “IMPORTANCE OF SPEAKER SPECIFIC SPEECH FEATURES FOR EMOTION RECOGNITION” 2019 5th Experiment International Conference (exp.at'19), Funchal (Madeira Island), Portugal, 2019, pp. 266-267, doi:10.1109/EXPAT.2019.8876534.

[7] S. R. Bandela and T. K. Kumar, “EMOTION RECOGNITION OF STRESSED SPEECH USING TEAGER ENERGY AND LINEAR PREDICTION FEATURES” 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT), Mumbai, 2018, pp. 422-425, doi: 10.1109/ICALT.2018.00107.

[8] S. Deebika, K. A. Indira and Jesline, “A MACHINE LEARNING BASED MUSIC PLAYER BY DETECTING EMOTIONS" 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2019, pp. 196-200, doi: 10.1109/ICONSTEM.2019.8918890.

[9] J. Umamaheswari and A. Akila, "AN ENHANCED HUMAN SPEECH EMOTION RECOGNITION USING HYBRID OF PRNN AND KNN" 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, pp. 177-183, doi: 10.1109/COMITCon.2019.8862221.

[10] T. M. W. Vithanawasam and B. G. D. A. Madhusanka, “FACE AND UPPER-BODY EMOTION RECOGNITION USING SERVICE ROBOT’S EYES IN A DOMESTIC ENVIRONMENT” 2019 International Research Conference on Smart Computing and Systems Engineering (SCSE), Colombo, Sri Lanka, 2019, pp. 44-50, doi: 10.23919/SCSE.2019.8842658.

[11] Mehta, Dhwani & Siddiqui, Mohammad Faridul Haque & Javaid, Ahmad. (2019). “RECOGNITION OF EMOTION INTENSITIES USING MACHINE LEARNING ALGORITHMS: A COMPARATIVE STUDY”, Sensors, **Volume 19**. 10.3390/s19081897.

APPENDICES

APPENDIX 1

CODING AND TESTING

1.1 Create_data.py

```
#creating database
import cv2, sys, numpy, os
haar_file = 'haarcascade_frontalface_default.xml'
datasets = 'datasets'  #All the faces data will be present this folder
sub_data = 'neutral'
####sub_data = 'hai'  #These are sub data sets of folder, for my faces I've used my name
path = os.path.join(datasets, sub_data)
if not os.path.isdir(path):
   os.mkdir(path)
(width, height) = (130, 100)    # defining the size of images
face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0) #'0' is use for my webcam, if you've any other camera attached use '1' like this
#webcam1 = cv2.VideoCapture(1)
# The program loops until it has 30 images of the face.
cntt=1;
count = 1
while count < 500:
   for i in range(0,2):
      if i==0:
            ##WEBCAM
         (_, im) = webcam.read()
            ##IPCAM
```

```
##          imgPath=urllib.urlopen(url)
##          imgNp=np.array(bytearray(imgPath.read()),dtype=np.uint8)
##          im=cv2.imdecode(imgNp,-1)
        #else:
            ##WEBCAM
            #(_, im) = webcam1.read()
            ##IPCAM
##          imgPath=urllib.urlopen(url2)
##          imgNp=np.array(bytearray(imgPath.read()),dtype=np.uint8)
##          im=cv2.imdecode(imgNp,-1)    (_, im) = webcam.read()
        gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, 1.3, 4)
        for (x,y,w,h) in faces:
            cv2.rectangle(im,(x,y),(x+w,y+h),(255,0,0),2)
            face = gray[y:y + h, x:x + w]
            face_resize = cv2.resize(face, (width, height))
            font = cv2.FONT_HERSHEY_SIMPLEX
            cv2.putText(im, 'FACE',(x+5,y-10), font, 0.5, (0, 255, 0), 2, cv2.LINE_AA)
            cv2.imwrite('%s/%s.png' % (path,count), face_resize)
        count += 1
        cntt +=1

        cv2.imshow('OpenCV', im)
        key = cv2.waitKey(10)

break
```

1.2 face_emotion.py

```
from keras.preprocessing.image import img_to_array
import imutils
import cv2
from keras.models import load_model
import numpy as np
```

```
from playsound import playsound
import time
import argparse
import matplotlib.pyplot as plt
parser = argparse.ArgumentParser()
parser.add_argument('-f', '--file',
   help='Path to video file (if not using camera)')
parser.add_argument('-c', '--color', type=str, default='gray',
   help='Color space: "gray" (default), "rgb", or "lab"')
parser.add_argument('-b', '--bins', type=int, default=16,
   help='Number of bins per channel (default 16)')
parser.add_argument('-w', '--width', type=int, default=0,
   help='Resize video to specified width in pixels (maintains aspect)')
args = vars(parser.parse_args())

# parameters for loading data and images
detection_model_path = 'haarcascade_files/haarcascade_frontalface_default.xml'
emotion_model_path = 'models/_mini_XCEPTION.102-0.66.hdf5'
# hyper-parameters for bounding boxes shape
# loading models
face_detection = cv2.CascadeClassifier(detection_model_path)
emotion_classifier = load_model(emotion_model_path, compile=False)
EMOTIONS = ["angry" ,"disgust","scared", "happy", "sad", "surprised",
 "neutral"]
# starting video streaming
cv2.namedWindow('my_face')
camera = cv2.VideoCapture(0)
time. sleep(2)
color = args['color']
bins = args['bins']
resizeWidth = args['width']
# Initialize plot.
fig, ax = plt.subplots()
if color == 'rgb':
   ax.set_title('Histogram (RGB)')
```

```
elif color == 'lab':
    ax.set_title('Histogram (L*a*b*)')
else:
    ax.set_title('Histogram (grayscale)')
ax.set_xlabel('Bin')
ax.set_ylabel('Frequency')
# Initialize plot line object(s). Turn on interactive plotting and show plot.
lw = 3
alpha = 0.5
if color == 'rgb':
    lineR, = ax.plot(np.arange(bins), np.zeros((bins,)), c='r', lw=lw, alpha=alpha, label='Red')
    lineG, = ax.plot(np.arange(bins), np.zeros((bins,)), c='g', lw=lw, alpha=alpha,
label='Green')
    lineB, = ax.plot(np.arange(bins), np.zeros((bins,)), c='b', lw=lw, alpha=alpha, label='Blue')
elif color == 'lab':
    lineL, = ax.plot(np.arange(bins), np.zeros((bins,)), c='k', lw=lw, alpha=alpha, label='L*')
    lineA, = ax.plot(np.arange(bins), np.zeros((bins,)), c='b', lw=lw, alpha=alpha, label='a*')
    lineB, = ax.plot(np.arange(bins), np.zeros((bins,)), c='y', lw=lw, alpha=alpha, label='b*')
else:
    lineGray, = ax.plot(np.arange(bins), np.zeros((bins,1)), c='k', lw=lw, label='intensity')
ax.set_xlim(0, bins-1)
ax.set_ylim(0, 1)
ax.legend()
plt.ion()
plt.show()
while True:
    frame = camera.read()[1]
    #reading the frame
  # Resize frame to width, if specified.
    if resizeWidth > 0:
        (height, width) = frame.shape[:2]
        resizeHeight = int(float(resizeWidth / width) * height)
        frame = cv2.resize(frame, (resizeWidth, resizeHeight),
            interpolation=cv2.INTER_AREA)
    # Normalize histograms based on number of pixels per frame.
```

```
    numPixels = np.prod(frame.shape[:2])
    if color == 'rgb':
        cv2.imshow('RGB', frame)
        (b, g, r) = cv2.split(frame)
        histogramR = cv2.calcHist([r], [0], None, [bins], [0, 255]) / numPixels
        histogramG = cv2.calcHist([g], [0], None, [bins], [0, 255]) / numPixels
        histogramB = cv2.calcHist([b], [0], None, [bins], [0, 255]) / numPixels
        lineR.set_ydata(histogramR)
        lineG.set_ydata(histogramG)
 lineB.set_ydata(histogramB)
   elif color == 'lab':
        cv2.imshow('L*a*b*', frame)
        lab = cv2.cvtColor(frame, cv2.COLOR_BGR2LAB)
        (l, a, b) = cv2.split(lab)
        histogramL = cv2.calcHist([l], [0], None, [bins], [0, 255]) / numPixels
        histogramA = cv2.calcHist([a], [0], None, [bins], [0, 255]) / numPixels
        histogramB = cv2.calcHist([b], [0], None, [bins], [0, 255]) / numPixels
        lineL.set_ydata(histogramL)
        lineA.set_ydata(histogramA)
        lineB.set_ydata(histogramB)
    else:
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        cv2.imshow('Grayscale', gray)
        histogram = cv2.calcHist([gray], [0], None, [bins], [0, 255]) / numPixels
        lineGray.set_ydata(histogram)
    fig.canvas.draw()
####
  frame = imutils.resize(frame,width=300)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces                                                                      =
face_detection.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flag
s=cv2.CASCADE_SCALE_IMAGE)
    canvas = np.zeros((600, 700, 3), dtype="uint8")
    frameClone = frame.copy()
    if len(faces) > 0:
```

```
        faces = sorted(faces, reverse=True,
        key=lambda x: (x[2] - x[0]) * (x[3] - x[1]))[0]
        (fX, fY, fW, fH) = faces
                # Extract the ROI of the face from the grayscale image, resize it to a fixed 28x28
pixels, and then prepare
            # the ROI for classification
        roi = gray[fY:fY + fH, fX:fX + fW]
        roi = cv2.resize(roi, (64, 64))
        roi = roi.astype("float") / 255.0
        roi = img_to_array(roi)
        roi = np.expand_dims(roi, axis=0)
    else: continue
    preds = emotion_classifier.predict(roi)[0]
    emotion_probability = np.max(preds)
    label = EMOTIONS[preds.argmax()]
    for (i, (emotion, prob)) in enumerate(zip(EMOTIONS, preds)):
            # construct the label text
            text = "{}: {:.2f}%".format(emotion, prob * 100)
              w = int(prob * 300)
            cv2.rectangle(canvas, (7, (i * 35) + 5),
            (w, (i * 35) + 35), (0, 255, 0), -1)
            cv2.putText(canvas, text, (10, (i * 35) + 23),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45,
            (255, 255, 255), 2)
            cv2.putText(frameClone, label, (fX, fY - 10),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 255, 0), 2)
            cv2.rectangle(frameClone, (fX, fY), (fX + fW, fY + fH),
                          (0, 255, 0), 2)
  cv2.imshow('my_face', frameClone)
    cv2.imshow("Probabilities", canvas)
   if cv2.waitKey(1) & 0xFF == ord('q'):
        break
camera.release()
cv2.destroyAllWindows()
```

1.3 face_emotion1.py

```
from keras.preprocessing.image import img_to_array
import imutils
import cv2
from keras.models import load_model
import numpy as np
from playsound import playsound
import time

# parameters for loading data and images
detection_model_path = 'haarcascade_files/haarcascade_frontalface_default.xml'
emotion_model_path = 'models/_mini_XCEPTION.102-0.66.hdf5'

# hyper-parameters for bounding boxes shape
# loading models
face_detection = cv2.CascadeClassifier(detection_model_path)
emotion_classifier = load_model(emotion_model_path, compile=False)
EMOTIONS = ["angry" ,"disgust","scared", "happy", "sad", "surprised",
 "neutral"]

# starting video streaming
cv2.namedWindow('my_face')
camera = cv2.VideoCapture(0)
time.sleep(2)
VarHappy=0
VarSad=0
Varangry=0
Vardisgust=0
Varscared=0
Varsurprised=0
Varneutral=0
Thresh=50
while True:
    frame = camera.read()[1]
```

```
    #reading the frame
    frame = imutils.resize(frame,width=500)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

faces=face_detection.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSize=(30,3
0),flags=cv2.CASCADE_SCALE_IMAGE)

    canvas = np.zeros((600, 700, 3), dtype="uint8")
    frameClone = frame.copy()
    if len(faces) > 0:
        faces = sorted(faces, reverse=True,
        key=lambda x: (x[2] - x[0]) * (x[3] - x[1]))[0]
        (fX, fY, fW, fH) = faces
                    # Extract the ROI of the face from the grayscale image, resize it to a fixed 28x28
pixels, and then prepare
            # the ROI for classification via the CNN
        roi = gray[fY:fY + fH, fX:fX + fW]
        roi = cv2.resize(roi, (64, 64))
        roi = roi.astype("float") / 255.0
        roi = img_to_array(roi)
        roi = np.expand_dims(roi, axis=0)

    else: continue
    preds = emotion_classifier.predict(roi)[0]
    emotion_probability = np.max(preds)
    label = EMOTIONS[preds.argmax()]

    for (i, (emotion, prob)) in enumerate(zip(EMOTIONS, preds)):
                # construct the label text
                text = "{}: {:.2f}%".format(emotion, prob * 100)

##                # draw the label + probability bar on the canvas
##                emoji_face = feelings_faces[np.argmax(preds)]

                w = int(prob * 300)
```

```
            cv2.rectangle(canvas, (7, (i * 35) + 5),
            (w, (i * 35) + 35), (0, 255, 0), -1)
            cv2.putText(canvas, text, (10, (i * 35) + 23),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45,
            (255, 255, 255), 2)
            cv2.putText(frameClone, label, (fX, fY - 10),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 255, 0), 2)
            cv2.rectangle(frameClone, (fX, fY), (fX + fW, fY + fH),
                          (0, 255, 0), 2)
##    for c in range(0, 3):
##        frame[200:320, 10:130, c] = emoji_face[:, :, c] * \
##        (emoji_face[:, :, 3] / 255.0) + frame[200:320,
##        10:130, c] * (1.0 - emoji_face[:, :, 3] / 255.0)

    cv2.imshow('my_face', frameClone)
    cv2.imshow("Probabilities", canvas)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
    if label=='happy':
        VarHappy=VarHappy+0.2
        if VarHappy>Thresh:
            print(VarHappy)
            playsound('E:/Face emotion code/playhappy.mp3')
            VarHappy=0
            VarSad=0
            Varangry=0
            Vardisgust=0
            Varscared=0
            Varsurprised=0
            Varneutral=0
    if label=='sad':
          VarSad=VarSad+1
          if VarSad>Thresh:
            VarSad=VarSad+1
            playsound('E:/Face emotion code/playsad.mp3')
```

```
        VarHappy=0
        VarSad=0
        Varangry=0
        Vardisgust=0
        Varscared=0
        Varsurprised=0
        Varneutral= 0
if label=='angry':
    Varangry=Varangry+1
    if Varangry>Thresh:
        playsound('E:/Face emotion code/playangry.mp3')
        VarHappy=0
        VarSad=0
        Varangry=0
        Vardisgust=0
        Varscared=0
        Varsurprised=0
        Varneutral=0
if label=='surprised':
    Varsurprised=Varsurprised+1
    if Varsurprised>Thresh:
        playsound('E:/Face emotion code/playsurprised.mp3')
        VarHappy=0
        VarSad=0
        Varangry=0
        Vardisgust=0
        Varscared=0
        Varsurprised=0
        Varneutral=0
if label=='disgust':
    if Vardisgust>Thresh:
        Vardisgust=Vardisgust+1
        playsound('E:/Face emotion code/playdisgust.mp3')
        VarHappy=0
        VarSad=0
```

```
            Varangry=0
            Vardisgust=0
            Varscared=0
            Varsurprised=0
            Varneutral=0
    if label=='neutral':
        Varneutral=Varneutral+1
        if Varneutral>Thresh:
            playsound('E:/Face emotion code/playneutral.mp3')
            VarHappy=0
            VarSad=0
            Varangry=0
            Vardisgust=0
            Varscared=0
            Varsurprised=0
            Varneutral=0
camera. release ()
cv2.destroyAllWindows()
```

APPENDIX 2

CONTRIBUTIONS

2.1 Abhash Shrestha

- Researched about the project.
- Decided to use Haar Cascade Algorithm.
- Decided to use Anaconda platform because of its wide range of software.
- Studied and analyzed reference papers.
- Major role in emotion classification and feature extraction.
- Partly wrote “Survey on Music Recommendation System Using Human Emotion”.
- Pivotal in generating report and documentation of report.

2.2 Naresh Bohara

- Collected necessary materials required.
- Learned about various python features and related libraries.
- Learned about face detection and played major role in face detection part.
- Incorporated sound libraries and played the music accordingly.
- Partly wrote “Survey on Music Recommendation System Using Human Emotion”.
- Pivotal in generating published paper and associated documentation.

PLAGIARISM REPORT

Format - I

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES
(**To be attached in the dissertation/ project report**)

1	Name of the Candidate **(IN BLOCK LETTERS)**	
2	Address of the Candidate	**Mobile Number :**
3	Registration Number	
4	Date of Birth	
5	Department	
6	Faculty	
7	Title of the Dissertation/Project	
8	Whether the above project/dissertation is done by	Individual or group : (Strike whichever is not applicable) a) If the project/ dissertation is done in group, then how many students together completed the project : b) Mention the Name & Register number of other candidates :
9	Name and address of the Supervisor / Guide	**Mail ID : Mobile Number :**
10	Name and address of the Co-Supervisor / Co- Guide (if any)	**Mail ID : Mobile Number :**

11	Software Used	
12	Date of Verification	
13	**Plagiarism Details: (to attach the final report from the software)**	

Chapter	**Title of the Chapter**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self citation)**	**% of plagiarism after excluding Quotes, Bibliography, etc.,**
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
Appendices				

I / We declare that the above information have been verified and found true to the best of my / our knowledge.

Signature of the Candidate	**Name & Signature of the Staff (Who uses the plagiarism check software)**
Name & Signature of the Supervisor/Guide	**Name & Signature of the Co-Supervisor/Co-Guide**

Name & Signature of the HOD

PAPER PUBLICATION

From: **ETERNAL SPACE Publication company**
<espubco@gmail.com>
Date: Mon, 4 May, 2020, 8:17 PM
Subject: Paper Published Link 04/05/2020
To: <sudeeshreddy11@gmail.com>, Lavi Raju
<laviraju88@gmail.com>, Suseela G
<suseelarajeshkanna@gmail.com>, Srinivasan Selvam
<srinivasan.selvam18@gmail.com>, Shivam Kumar Singh
<shivusingh3335@gmail.com>, John Justin
<johnjustin.er@gmail.com>, charansai kundeti
<charansai15799@gmail.com>,
<stewartk.sse@saveetha.com>, Magesh S
<mageshs.sse@saveetha.com>, chandar G
<Chan.it13@gmail.com>, <tejaswinimurthy77@gmail.com>,
Sangamithrai K <sangamithraik.sse@saveetha.com>, Karthik
Elangovan <karthik.tes@gmail.com>, <jvkk1999@gmail.com>,
Lavanya Naidu <lavanyachowthry5@gmail.com>,
publication2019 batch <batchpublication16@gmail.com>

Dear Authors

Please Find below paper published link,

S.no	ID	Name	Title	Published link
1	SCOPUSAPR004	V.Sudeesh	REVIEW ON DEMORALIZATION DYNAMIC ENERGY SHARING STRATEGY FOR HYBRID ENERGY STORAGE SYSTEM IN DC MICROGRID	http://www.testmagzine.biz/index.php/testmagzine/article/view/7029/5406
2	SCOPUSAPR005	V.Sudeesh	DEMORALIZATION DYNAMIC ENERGY SHARING STRATEGY FOR HYBRID ENERGY STORAGE SYSTEM IN DC MICROGRID	http://www.testmagzine.biz/index.php/testmagzine/article/view/7030/5407
3	SCOPUSAPR030	Dr.G.Niranjana	Image based Emotion aware Music Recommendation System using Machine Learning	http://www.testmagzine.biz/index.php/testmagzine/article/view/7031/5408
4	SCOPUSAPR039	S.Srinivasan	SPEECH INTERFACE FOR MOBILE APPLICATION	http://www.testmagzine.biz/index.php/testmagzine/article/view/7032/5409
5	SCOPUSAPR045	Shivam Kumar Singh	HAND GESTURE RECOGNITION AND CURSOR CONTROL	http://www.testmagzine.biz/index.php/testmagzine/article/view/7033/5410
6	SCOPUSAPR048	Charan Sai Kundeti	SECURITY OF CLINICAL BIGDATA IN MEDICINALSERVICES USING CLOUD SERVER	http://www.testmagzine.biz/index.php/testmagzine/article/view/7034/5411
7	SCOPUSAPR093	Dr.S. Magesh	ANALYSIS OF QUEUING BASED VIRTUAL INFRASTRUCTURE MANAGEMENT FOR RESOURCE SCHEDULING IN CLOUD COMPUTING	http://www.testmagzine.biz/index.php/testmagzine/article/view/7035/5412
8	SCOPUSMAR127	J Venkata Krishnakanth	A REVIEW BEHAVIORAL DETECTION FRAMEWORKUSING NETSPAM	http://www.testmagzine.biz/index.php/testmagzine/article/view/7036/5413
9	SCOPUSMAR164	N.Lavanya	Detection and classification using support vector machine in CNN	http://www.testmagzine.biz/index.php/testmagzine/article/view/7037/5414
10	SCOPUSAPR024	KONANGI TEJASWINI PRIYA	TRAVEL GUIDE FOR ADVENTUROUS TRIP	http://www.testmagzine.biz/index.php/testmagzine/article/view/7047/5424

--

Regards,
Dr.G.Niranjana
Associate Professor,
Department of CSE,
SRM Institute of Science and Technology,
Kattankulathur