

Partial Project Report
On

REAL TIME EMOTION RECOGNITION USING FACIAL EXPRESSIONS

Submitted to the
Savitribai Phule Pune University, Pune
In partial fulfilment for the award of the Degree of
Bachelor of Engineering in Information Technology

By

407016 Rugved Chaudhary
407007 Mahesh Auti
407026 Deval Sonkusare
407038 Ankul Gupta

Under the guidance of

Prof. R.S. Sonar



Sinhgad Institutes

Department of Information Technology
Sinhgad College of Engineering

S. No.44/1, Off. Sinhgad Road, Vadgaon (Bk.), Pune-411041, Maharashtra, India
(2017 - 2018)



Sinhgad Institutes

Sinhgad College of Engineering, Pune-411041

Department of Information Technology

CERTIFICATE

This is to certify that

Rugved Chaudhary (Roll No. 407016)

Mahesh Auti (Roll No. 407007)

Deval Sonkusare (Roll No. 407026)

Ankul Gupta (Roll No. 407038)

have successfully completed Partial Project Report on

Real Time Emotion Recognition Using Facial Expressions

Towards the partial fulfillment of Bachelor's
Degree In Information Technology of Savitribai Phule Pune University, Pune
in the academic year 2017-2018.

Prof. R. S. Sonar

Project Guide

Prof. G. R. Pathak

HOD-IT

Dr. S. D. Lokhande

Principal

Acknowledgement

We are highly indebted to our guide Prof R. S. Sonar for her guidance and constant supervision as well as for providing necessary information regarding the report and also for the support in completing the report. We would like to express our special gratitude and thanks to Prof. B. P. Vasgi and other Staff Members for giving us such attention and time.

This acknowledgement would be incomplete without expressing our special thanks to Prof. G. R. Pathak, Head of the Department (Information Technology) for his support during the work.

We would like to extend our heartfelt gratitude to our Principal, Dr. S. D. Lokhande who provided a lot of valuable support, mostly being behind the veils of college bureaucracy.

We would also like to express our gratitude towards our parents and friends for their kind co-operation and encouragement which help us in completion of this report. Our thanks and appreciations also go to our colleague in developing the report and people who have willingly helped us out with their alignments.

Rugved Chaudhary (Roll No. 407016)

Mahesh Auti (Roll No. 407007)

Deval Sonkusare (Roll No. 407026)

Ankul Gupta (Roll No. 407038)

Abstract

Increasing and maintaining human productivity of different tasks in stressful environment is a challenge. Music is a vital mood controller and helps in improving the mood and state of the person which in turn will act as a catalyst to increase productivity. Continuous music play requires creating and managing personalized song playlist which is a time consuming task. It would be very helpful if the music player itself selects a song according to the current mood of the user. The mood of the user can be detected by a facial expression of the person. A facial expression detection system should address three major problems: detection of face from an image, facial feature extraction and facial expression classification. The first stage is of face detection from an image for which various techniques used are model based face tracking which includes real-time face detection using edge orientation matching , Robust face detection using Hausdorff distance, weak classifier cascade which includes Viola and Jones algorithm, and Histograms of Oriented Gradients (HOG) descriptors. The next stage is to extract features from detected face. The proposed system will use classic Histograms of Oriented Gradients (HOG) along with facial landmark detection technique; these detected features then passed through SVM classifier to predict the mood of the user. This predicted mood will stimulate the creation of playlist.

Contents

1	EMOTION RECOGNITION USING FACIAL EXPRESSIONS	
1.1	Introduction	1
1.2	Motivation	2
1.3	Objectives of Work	2
2	LITERATURE SURVEY	3
3	PROBLEM STATEMENT	4
3.1	Recognition of Need	4
3.2	Major Problems	4
3.2.1	Inefficient Human Machine interaction.....	4
3.2.2	Time consumption in search of a suitable song..	.5
4	SYSTEM ANALYSIS	6
4.1	Software Requirements	6
4.1.1	Django	6
4.1.2	Bootstrap7
4.1.3	Python	7
4.1.4	CSS	7
4.2	Data Collection7
4.3	Information Gathering Tools.....	.8
4.3.1	Internet	8
4.3.2	Our Own Knowledge8
4.4	Algorithm8
4.5	UML Diagrams	11
4.5.1	Class diagram	11
4.5.2	State diagram.....	11
4.5.3	Use Case diagram	12
4.5.4	Sequence diagram13
4.5.5	Activity diagram14
4.5.6	Component diagram.....	15
4.5.7	Deployment diagram	16
5	CONCLUSION	17
6	REFERENCES	

List of Figures

4.5.1	Class Diagram.....	11
4.5.2	State Diagram	11
4.5.3	Use Case Diagram	12
4.5.4	Sequence Diagram	13
4.5.5	Activity Diagram	14
4.5.6	Component Diagram	15
4.5.7	Deployment Diagram	16

Chapter 1

EMOTION RECOGNITION USING FACIAL EXPRESSIONS

This chapter gives a brief description about our project, and motivation behind designing this project.

1.1 Introduction

This project is based on the principle of detection of human emotions using image processing, and to play music which is appropriate for enhancing that emotional state. It works when mathematical operations are performed using the framework of signal processing which uses an image or a series of images as input.

The state of mind and current emotional mood of human beings can be easily observed through their facial expressions. The different muscles beneath the face act as action units which signal different emotions. The Institute of Neuroscience and Psychology researched that „wide open eyes“ were the signals of happy and sad nature, whereas the „wrinkled nose“ depicted anger or disgust. This project was made by taking three basic emotions (happy, sad, and neutral) into consideration. The face detection in this project is made by creation of face chips on the dimensions of a human face by creating and joining multiple feature points on chin, cheeks, lips, forehead, etc. An algorithm is created and inserted into the system which classifies the emotions into their respective order based on various machine learning techniques. Music is often described as a „language of emotions“ throughout the globe. Let it be a 80 year old man or a 12 year old girl, everyone has their taste and liking towards a type of music. Hard hitting evidence on why human brain reacts to music differently is not available but scientists have discovered some findings which state that the brain through cerebellum activation synchronizes the pulse of music with the neural oscillators. While processing music brain's language centre, emotional centre and memory centre are connected thereby stimulating a thrill obtained by expected beats in a pattern to provide a synesthetic experience. This project was therefore aimed to provide people with befitting music using facial recognition, saving the time which is required to go into the files and scroll at a never ending list of songs to choose from thereby enhancing user experience.

1.2 Motivation

The main motivation of this system is to take a step forward in the field of artificial intelligence and image processing applications. This system will help the users to minimise their efforts in managing large playlists.

Generally people have a large number of songs in their database or playlists. Thus to avoid trouble of selecting a song, most people will just randomly select a song from their playlist and some of the songs may not be appropriate for the current mood of the user and it may disappoint the user. As a result, some of the songs are not matching to the user's current emotion. Moreover, there is no commonly used application which is able to play songs based on the current emotions of the user.

1.3 Objectives of Work

Proposed system aims to extract user's facial expressions and features to determine the current mood of the user. Once the emotion is detected, playlist of songs suitable to the mood of the user will be presented to him.

Main objectives are –

- Enhance human machine interactions
- Determine the facial expressions such as : Happy, Sad, Stressed, surprised
- Build an application (Music Player) to present the proposed model

Chapter 2

LITERATURE SURVEY

There is ample research and work carried out in the field of emotion detection from faces. Some relevant work which was surveyed were mainly industry standard and leading research in the field.

Viola and Jones [1], proposed a robust way to detect human faces in the image. They introduced a new image representation known as “Integral Image”. They implemented a simple and efficient classifier by using Adaboost learning algorithm to select a small number of critical features from large set of features. Lastly, they introduced a method for combining classifiers in a cascade like structure which quickly discards background regions of the image.

Another popular technique for face detection is the classic Histograms of Oriented Gradients (HOG). Deniz, Bueno, Salido, and Torre [2] contributed towards, firstly to minimize the errors occurring during the process of face detection due to partial occlusion, pose, and effect of brightness. Secondly it captures major structure for face detection.

For feature detection in detected face Visutsak[3], proposed an image based approach for emotion classification through lower facial expression. He used A-SVM classifier to classify the features in seven different emotions namely neutral, disgust, happy, sad, angry, fear, and surprised. Kazemi and Sullivan developed a efficient technique for landmark extraction from detected face in there paper One Millisecond Face Alignment with an Ensemble of Regression Trees [4]. They addressed issue of effectively estimating the face"s landmark position. The landmark positions are the important points on the face which will be consider as feature for classification. They showed an ensemble of regression trees which can be used to estimate the faces landmark position directly from a sparse subset of pixel intensities. This technique gives very high quality predictions.

For classification of feature points Dumas [5], showed that SVM can be applied to the problem of classifying emotions on human faces. Her experiments showed that performance of SVM was equivalent to the performance of neural network

Chapter 3

PROBLEM STATEMENT

3.1 Recognition of Need

Generally people have a large number of songs in their database or playlists. Thus to avoid trouble of selecting a song, most people will just randomly select a song from their playlist and some of the songs may not be appropriate for the current mood of the user and it may disappoint the user. As a result, some of the songs are not matching to the user's current emotion. Moreover, there is no commonly used application which is able to play songs based on the current emotions of the user.

The main motivation of this system is to take a step forward in the field of artificial intelligence and image processing applications. This system will help the users to minimise their efforts in managing large playlists

3.2 Major Problems

- Inefficient Human Machine interaction
- Time Consumption in search of a suitable song
- Trouble of creating and managing playlists

3.2.1 Inefficient Human Machine interaction

Humans and machines have been interacting together for a while now. The world is getting more and more digitized. Since the advancement in technology, we have come long way. Earlier, Machines only had a sense of touch. But now the machines can see and hear as well.

Why not make the use of these senses to create something useful? The state of mind and current emotional mood of human beings can be easily observed through their facial expressions.

3.2.2 Time consumption in search of a suitable song

Whenever we go to listen a random song, we have to go through so many files and scroll at never ending list of songs to choose an appropriate song to enhance our mood.

This project is therefore aimed to provide people with befitting music using facial recognition, saving the time which is required to go into the files and scroll at a never ending list of songs to choose from thereby enhancing user experience

Chapter 4

SYSTEM ANALYSIS

4.1 Software Requirements

Django is a free and open-source web framework, written in Python, which follows the model-view-template (MVT) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established as a non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Some well-known sites that use Django include the Public Broadcasting Service, Instagram, Mozilla, TheWashingtonTimes, Disqus, Bitbucket and Nextdoor.

Applications:

Some well-known sites that use Django include :

1. PublicBroadcastingService,
2. Instagram
3. Mozilla,
4. TheWashingtonTimes,
5. Disqus,
6. Bitbucket
7. Nextdoor.

4.1.2 Bootstrap

Bootstrap is a free and open source front end web framework for designing websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

4.1.3 Python

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

4.1.4 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language.^[1] Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.^[2]

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts.^[3] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.

4.2 Data Collection

We collected the data from the internet and some reference books. We also got some data professionals who gave us guidance how to do the project. Mostly we used

the knowledge to the project and also collect data from various sources like books, Wikipedia, internet, etc.

4.3 Information Gathering Tools

4.3.1 Internet

Wikipedia and YouTube are good information gathering tools for us. It contains information about any concept, any person required. That is why information about components as well as implementation of the whole project was easy.

4.3.2 Our Own Knowledge

We also went through a couple of books and IEEE papers, and contributed with our own knowledge to accomplish this particular project.

4.4 Algorithm

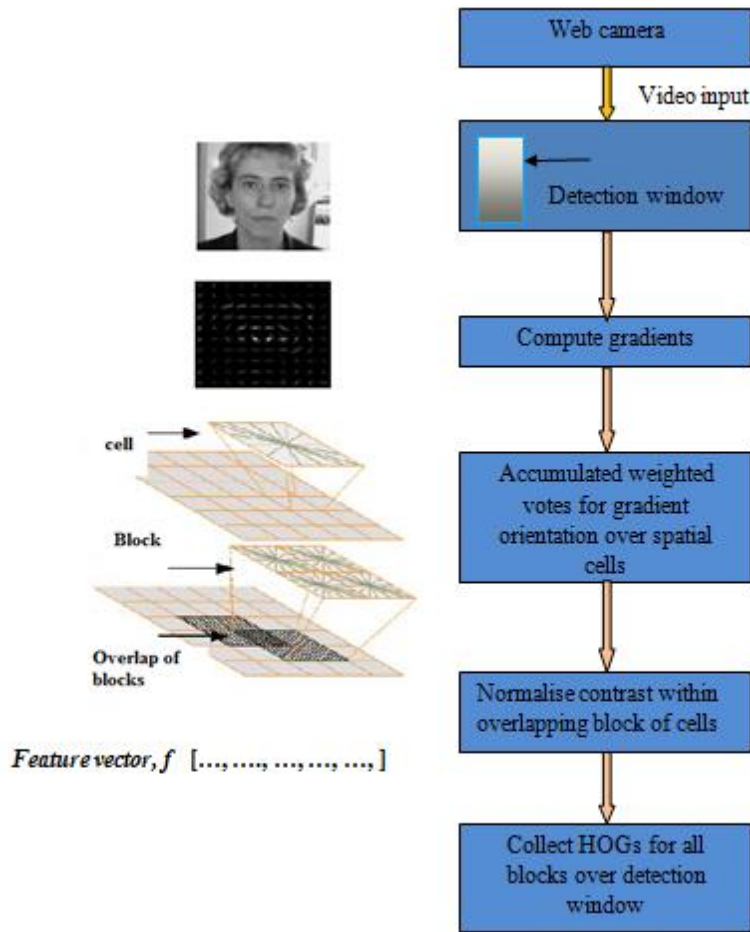
Histogram of Oriented Gradient (HOG) Algorithm

Histogram of Oriented Gradient (HOG), a method of intensive descriptors that is used for local overlapped images, constitutes features by calculating the local direction of gradient.

At present, the approach combining HOG with Support Vector Machine (SVM) has been widely applied to image recognition and achieved a great success especially in human detection. The advantage of HOG feature is that it is based on histogram of oriented gradient. It can not only describe the feature of face contours, but also be not sensitive to light and small offset. Obtain the human facial features by combining the features of all blocks in line.

Take the input image of 256×256 as an example shown in fig shows the procedure of extracting depth image's HOG features, we calculate the HOG feature as follows:

- 1) Input an image which is the video frame captured continuously by web camera.
- 2) Gradient calculation: use the $[-1, 0, 1]$ and $[1, 0, -1]$ median filter to perform filtering, calculate the vertical gradient and horizontal gradient of the image, and then calculate the gradient direction and gradient magnitude of each pixel.
- 3) Divide the inputting image into average small cells (including 256×256 pixels) and combine four cells into a small block, one block is constituted by a cell of 2×2 .
- 4) The selection of direction channel: divide 00° - 180° or 00° - 360° into n channels averagely. In this paper, we divide $+900$ to -900 into thirteen equal parts, that is, thirteen channels in total. So there are $4 \times 13 = 52$ features in each block.



5) The acquisition of the histogram: get the statistics of each pixel in each cell of their histogram of orientated gradient. The abscissa of the histogram represents the thirteen direction channels selected in step 3, and the ordinate represents the summation of the gradient, belonging to a certain direction channel. Thus, we get a set of vectors.

6) The process of normalization: normalize the vectors in blocks in which pixels correspond with the vectors. Block normalization corrects local contrast variations and histograms for the cells of each block are normalized.

HOG method uses 6 basic parameters recalling: number of orientation bins, range of orientations to be considered, cell

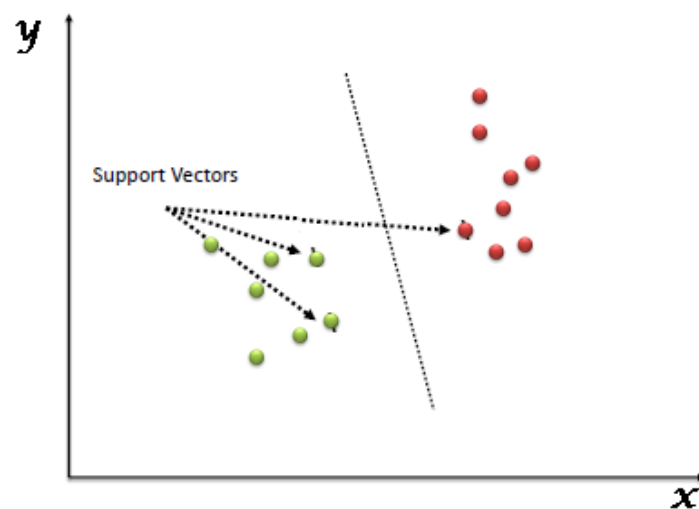
size, block size, overlap and normalization rule. In this paper, hog descriptor implemented by the $L2$ -norm suggested in equation

$$f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}$$

7) Form HOG features: combine all the vectors processed above and then form a set of vectors, which are the HOG features. These steps ensure that little information is lost during the encoding process. Overall encoding focuses on capturing relevant fine grained features and adding the required degree +900 to -900 of invariance at each step to detect face or non face.

Support Vector Machine Algorithm

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot)



How to Implement SVM in python?

In Python, scikit-learn is a widely used library for implementing machine learning algorithms, SVM is also available in scikit-learn library and follow the same structure (Import library, object creation, fitting model and prediction)

4.5 UML Diagrams

4.5.1 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. A collection of class diagrams represent the whole system. Class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system. The following diagram shows class diagram.

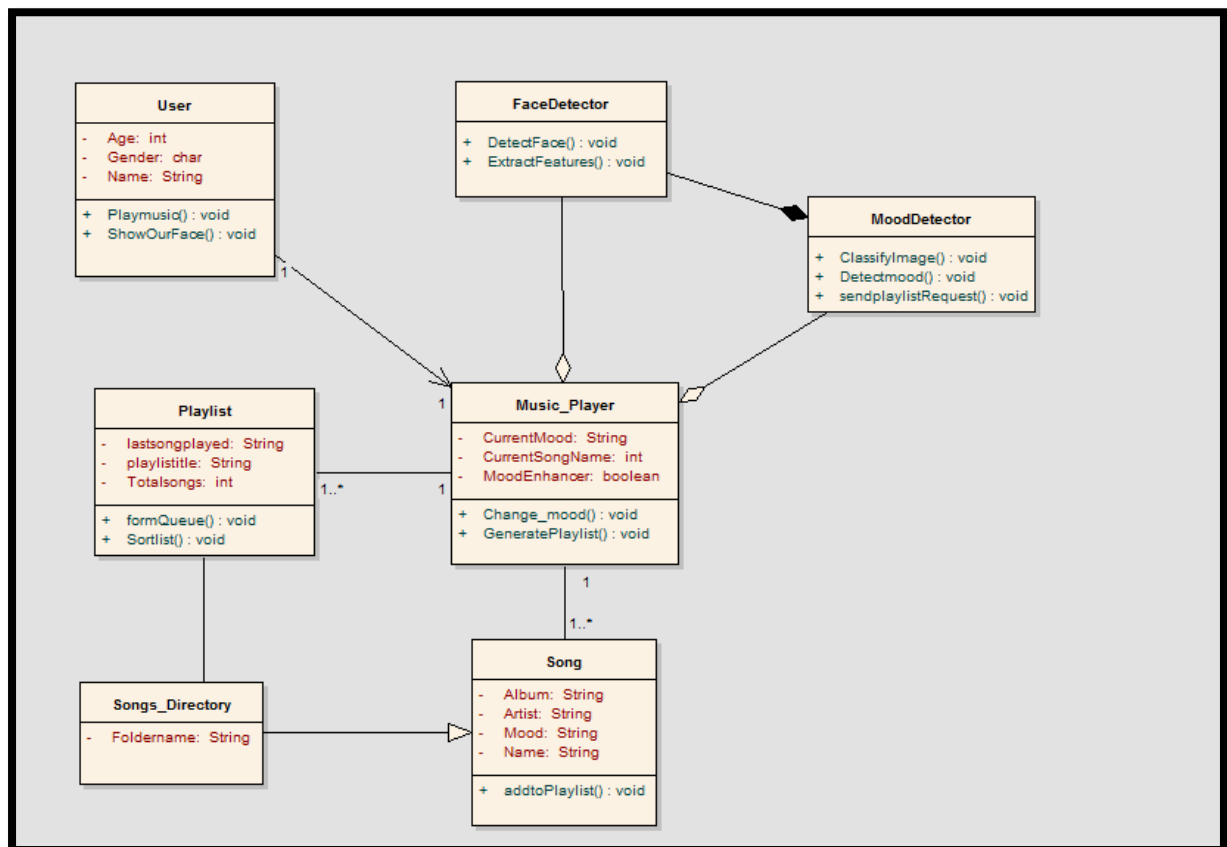


Fig 4.5.1: Class Diagram

4.5.2 State Diagram

State diagram are used to give an abstract description of the behaviour of system. This behaviour is analyzed and represent as series of events that can occur in one or more possible states which is described in the figure below.

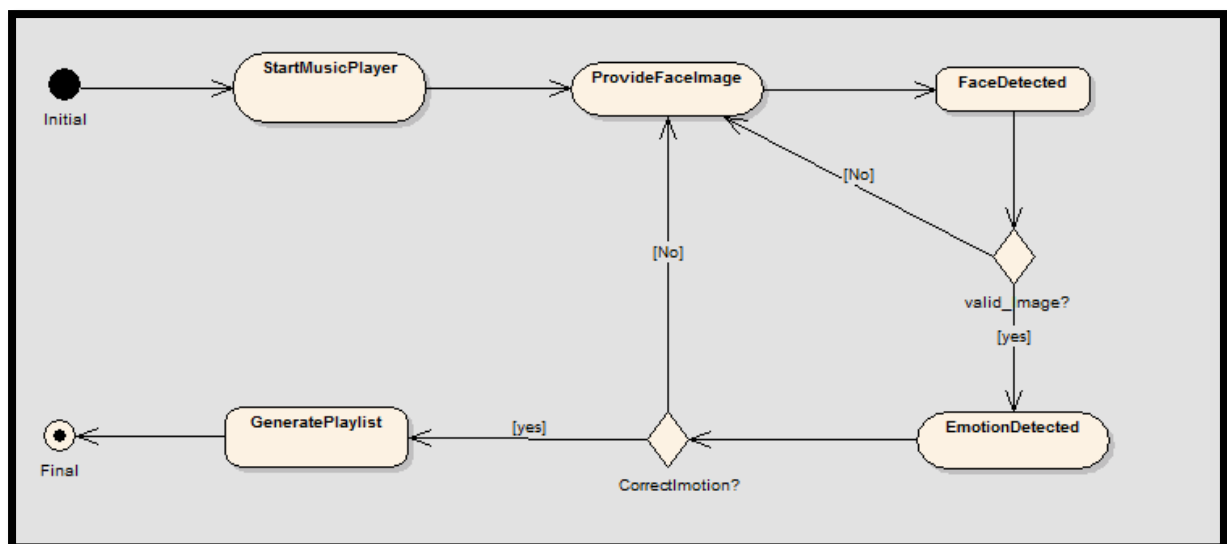


Fig 4.5.2: State Diagram for Proposed System

4.5.3 Use Case diagram

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating. Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior.

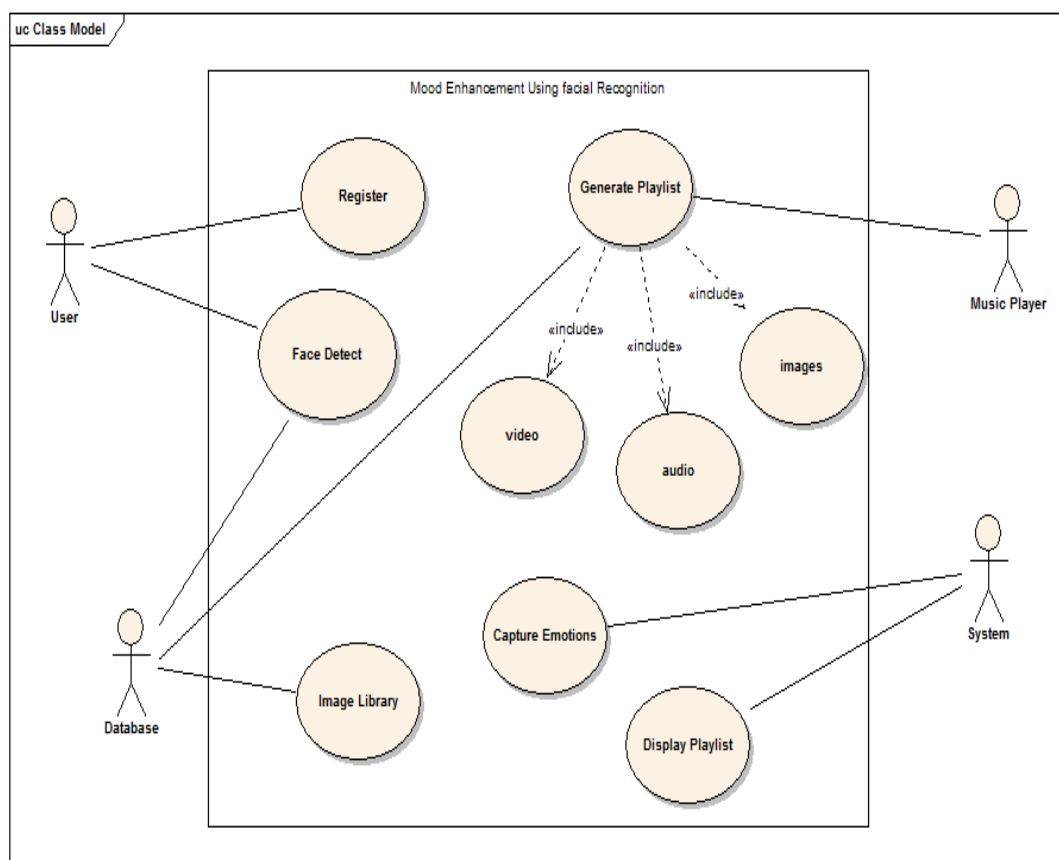


Fig 4.5.3: Use Case Diagram

4.5.4 Sequence Diagram

Sequence diagram is an interaction diagram that shows how objects interact with one another and in what order. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. The given figure is example of sequence diagram.

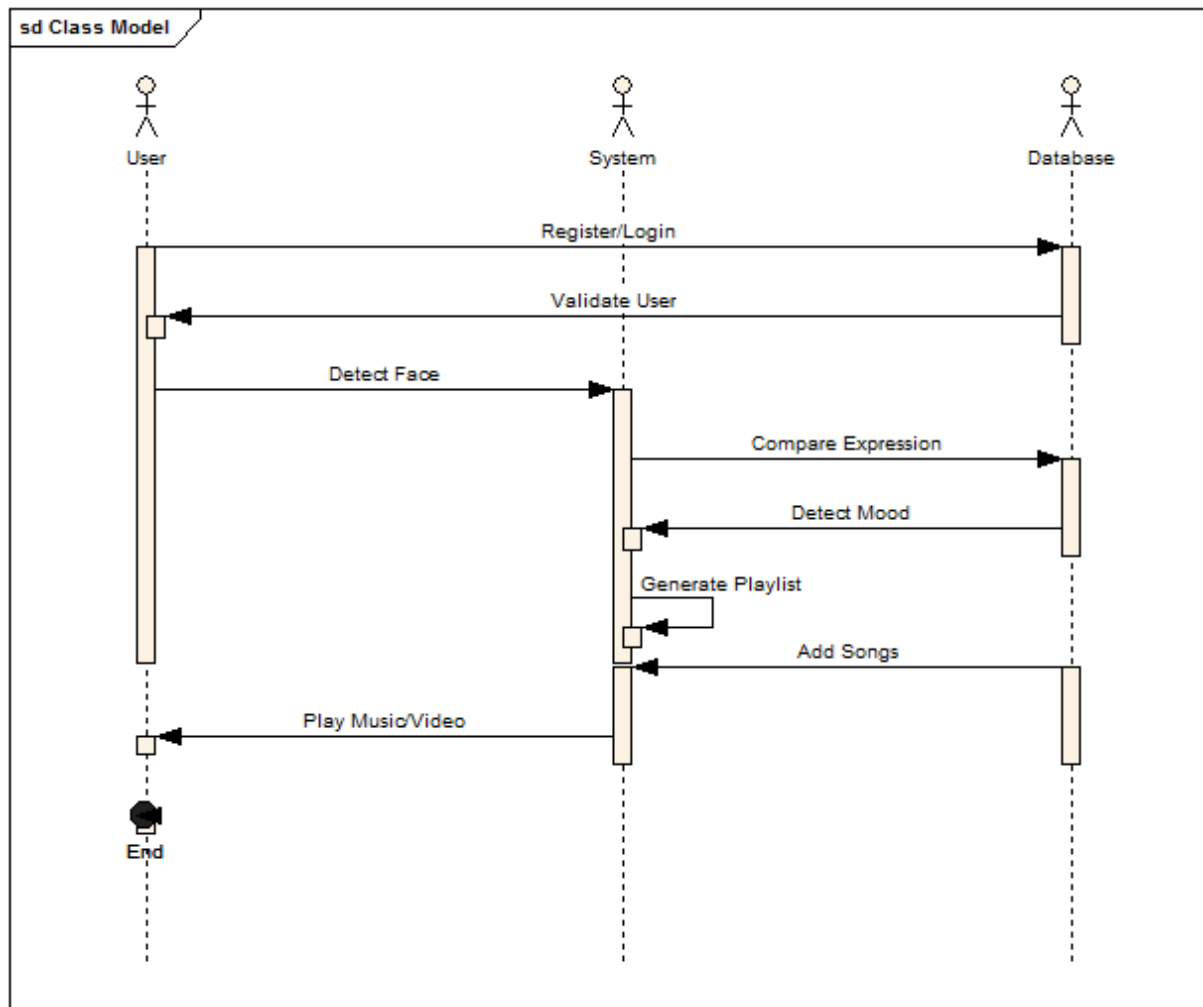


Fig 4.5.4: Sequence Diagram for proposed system.

4.5.5 Activity diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. Control flow is drawn from one operation to another. Basic workflow of our system is as shown in below figure.

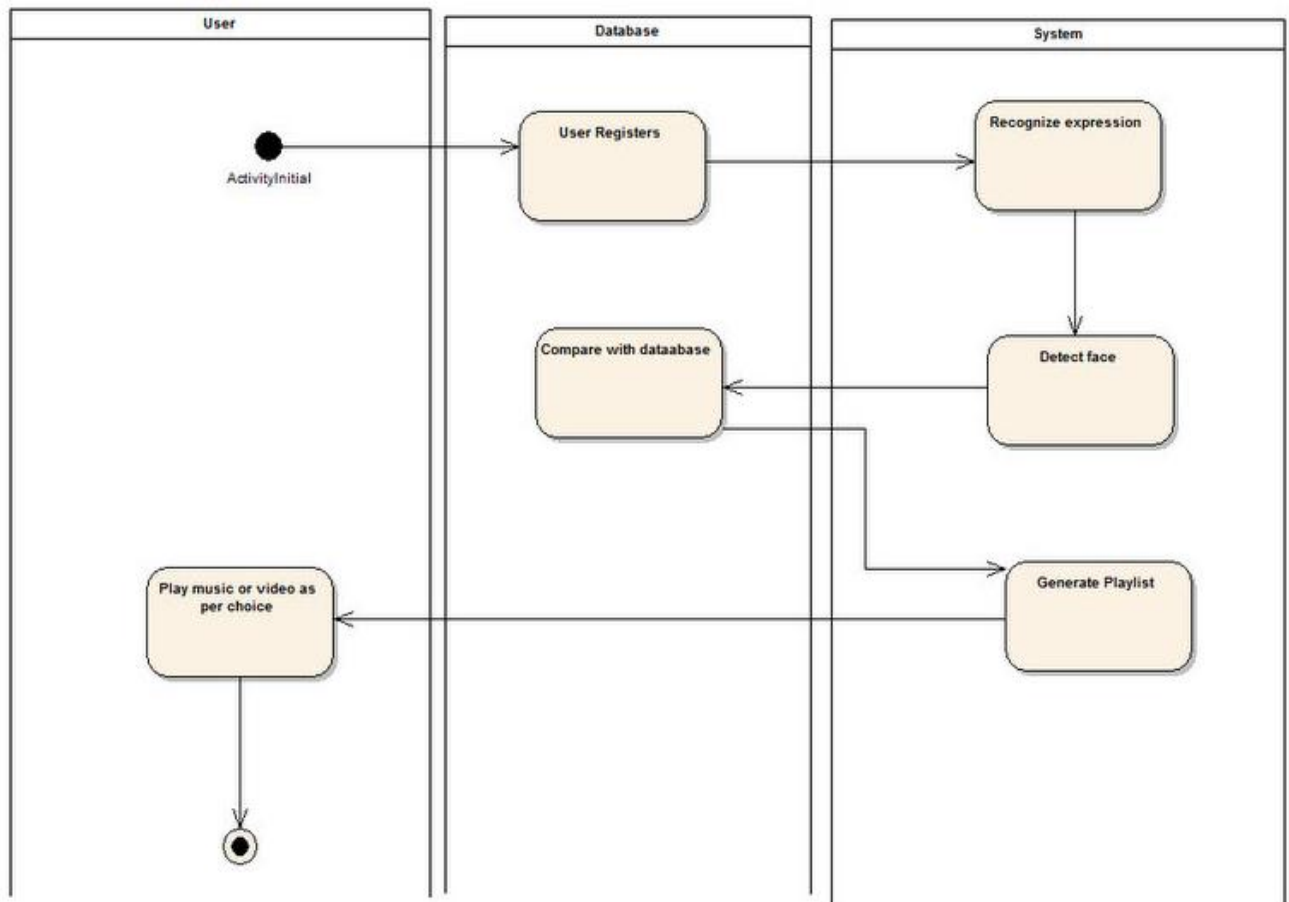


Fig 4.5.5: Activity Diagram

4.5.6 Component Diagram

Component diagram shows components, provided and required interfaces, ports, and relationships between them. This type of diagrams is used in Component-Based Development (CBD) to describe systems with Service-Oriented Architecture (SOA). Component-based development is based on assumptions that previously constructed components could be reused and that components could be replaced by some other "equivalent" or "conformant" components, if needed.

The artifacts that implement component are intended to be capable of being deployed and re-deployed independently, for instance to update an existing system.

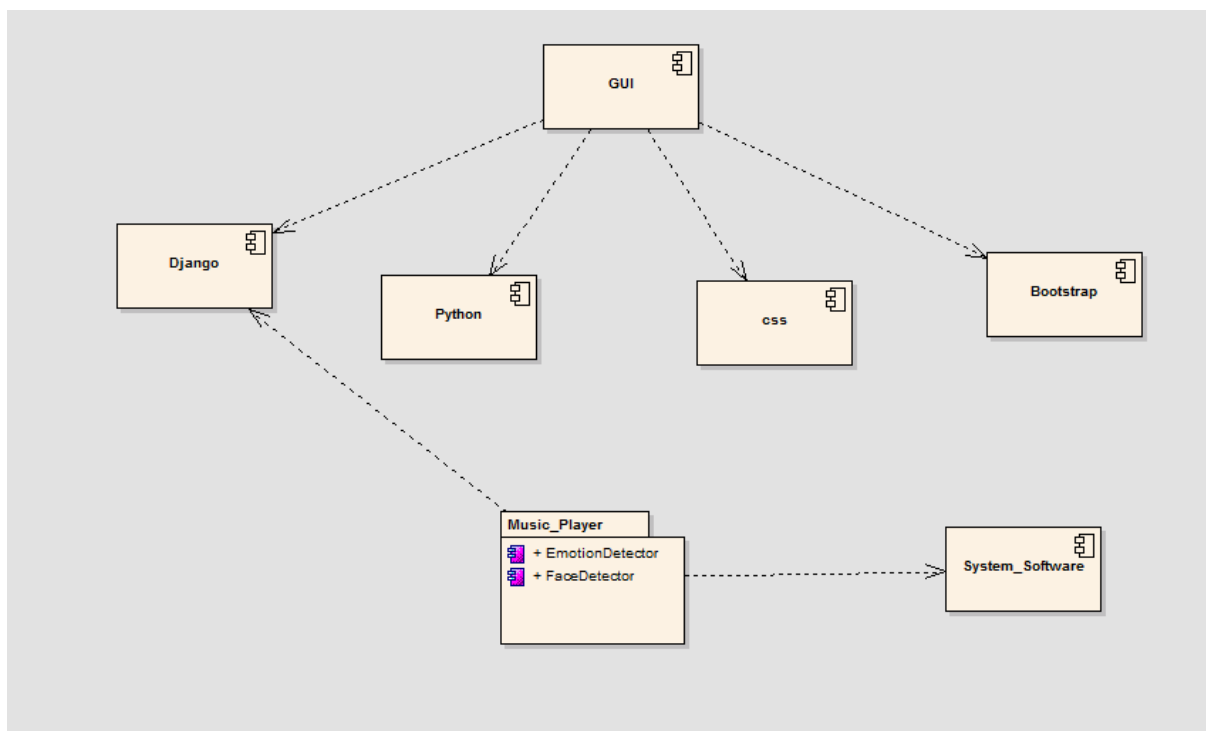


Fig 4.5.6: Componenet Diagram for Proposed System

4.5.7 Deployment Diagram

Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. Deployment diagrams could describe architecture at specification level (also called type level) or at instance level (similar to class diagrams and object diagram). In deployment diagram, the three-dimensional boxes, known as nodes, represent the basic software or hardware elements, or nodes, in the system.

The following diagram is an example of Deployment diagram.

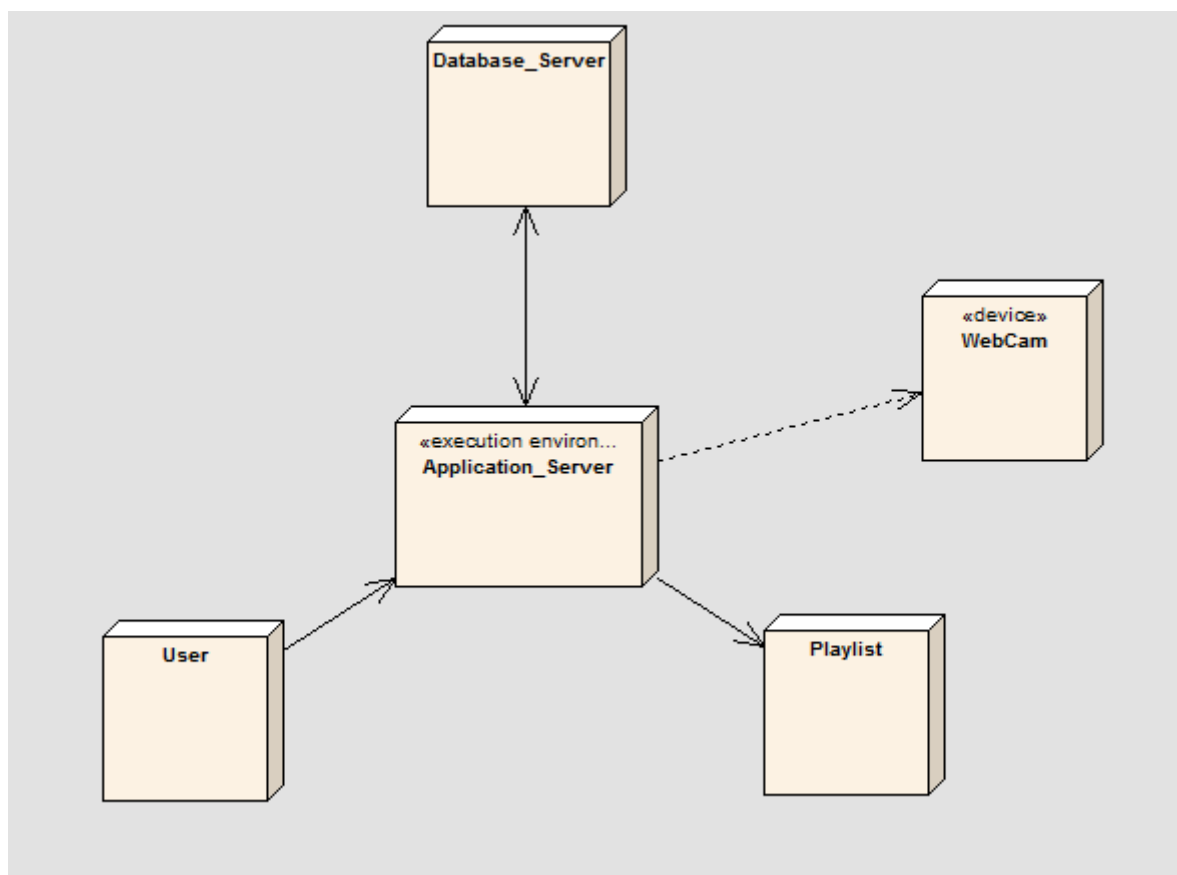


Fig 4.5.7: Deployment Diagram for Proposed System

Chapter 5

CONCLUSION

We have planned to design Emotion based music player which uses a face detector which is based on Histograms of Oriented Gradients (HOG) descriptors technique and a facial landmark detection system. Further, this facial landmark points will be classified into three different moods using CSVM with Radial Basis Function kernel to have high accuracy. The system will be able to effectively categorize the songs based on the detected mood.

Chapter 5

REFERENCE

- [1] PAUL VIOLA Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA And MICHAEL J. JONES Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, USA Robust Real-Time Face Detection International Journal of Computer Vision 57(2), 137–154, 2004_c 2004 Kluwer Academic Publishers. Manufactured in The Netherlands
- [2] Face recognition using Histograms of Oriented GradientsO. Déniz, G. Bueno, J. Salido, F. De la Torre Universidad de Castilla-La Mancha, E.T.S. Ingenieros Industriales, Avda. Camilo Jose Cela s/n, 13071 Ciudad Real, Spain Carnegie Mellon University, Robotics Institute, 211 Smith Hall, 5000 Forbes Ave., Pittsburgh, PA 15213, USA Pattern Recognition Letters 32 (2011) 1598–1603
- [3] Porawat Visutsak Emotion Classification through Lower facial Expressions using Adaptive Support Vector Machines
- [4] Vahid Kazemi and Josephine Sullivan One millisecond face alignment with an Ensemble of regression Trees Computer Vision Foundation CVPR2014
- [5] Melanie Dumas Department of computer science,University of california,san Diego CA 921920114