A Project Report on

EMOTION AWARE SMART MUSIC RECOMMENDED SYSTEM USING CNN

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science Engineering **Sathyabama Institute of Science and Technology**

(Deemed to be University)

Submitted in partial fulfillment of the requirements
for the award of Bachelor of Engineering Degree in
Computer Science and Engineering

By

PINISETTI JAYAPRAKASH
REG. NO 38110405

PODAKATLA KUMARA SWAMY
REG. NO 38110409

DEPARTMENT OF COMPUTER SCIENCE ANDENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV
GANDHISALAI, CHENNAI – 600119,
TAMILNADU

MARCH 2022

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC (Established Under Section 3 of UGC Act, 1956) JEPPIAAR NAGAR,RAJIV GANDHI SALAI CHENNAI– 600119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **PINISETTI JAYA PRAKASH (38110405) and P.kumarswamy(38110409)** who carried out the project entitled "**Emotion aware Smart Music Recommender System using CNN** " under my supervision from NOV 2021 to MAR 2022.

Internal Guide

Ms.T. Sasikala, M.E.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Dr. LAKSHMANAN L, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

InternalExaminer **ExternalExaminer**

DECLARATION

I **PINISETTI JAYAPRAKASH** and **P.KUMARSWAMY** here by declare that the project reportentitled **Emotion aware Smart Music Recommender System using CNN** done by me under the guidance of **Dr. T. Sasikala, M.E,**is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

PLACE: **SIGNATURE OF THECANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E ., Ph.D**,**Dean**, School of Computing, **Dr.S.Vigneshwari,M.E.,Ph.D.andDr.L.Lakshmanan,M.E.,Ph.D.,Heads of the Department** of **ComputerScience and Engineering** for providing me necessary support and details at the right time during the progressivereviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr .T. Sasikala , M.E.,Ph.D** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TABLE OFCONTENTS

CHAPTERNO.	TITLE	PAGENO.
	ABSTRACT	**V**
	LISTOFFIGURES	**VIII**
	LISTOFABBREVIATIONS	**IX**
1	**INTRODUCTION**	**1**
		1
		2
		2
		2
		2
		3
	• CNN ALOGRITHM	**10**
	• CONVULATION LAYER	**11**
	• POOLING LAYER	**13**
	• FULLY CONNECTED LAYER	
	• PROBLEMSTATEMENT	**`15**
	• PROBLEMDESCRIPTION	
2	**LITERATURESURVEY**	**16**
	• MOOD BASED MUSIC RECOMMENDER SYSTEM	**18** **18**
	● An Emotion-Aware Personalized Music Recommendation System Using a Convolutional Neural Networks Approach	**20**
	● DEEP LEARNING IN MUSIC	
	● Review on Facial Expression Based Music Player	
	• SMART MUSIC PLAYER BASED ON FAIAL EXPRESSION	
3	**SOFTWARE RQURIMENTS**	**21**
	• EXISTINGSYSTEM	**22**
	• PROPOSEDSYSTEM	**22**
	• ArchitectureOverview	22
	• SYSTEMREQUIREMENTS	23
4	**SYSTEM ANALYSIS**	**24**
	• PURPOSE	**24**
	• SCOPE	**24**
	• EXISTING SYSTEM	**24**

	• PROPSED SYSTEM	**25**

5	**SYSTEM DESGIN** • INPUT DESGIN • OUTPUT DESGIN • DATAFLOW • UML DIAGREMS	26 26 26 26 26 27
6	**MODULES** • Data Collection Module • Emotion Extraction Module • Audio Extraction Module • Emotion - Audio Integration Module	32 32 32
7	**SYSTEM IMPLEMENTATION** • SYSTEM ARTICTURE	33
8	**SYSTEM TESTING** • TEST OF PLAN • VERIFICATION • VALIDATION • WHITE BOX TESTING • BLACK BOX TESTING • TYPES OF TESTING • REQURIMENT ANALYSIS • FUNCTIONAL ANALYSIS • NON FUNCTIONAL ANALYSIS	35 35 36 40
9	CONCLUSION	41
	REFERENCES	42
	PLAGRISM REPORT	43

LIST OF FIGURES

1 USE CASE DIAGRAM 31

2 CLASS DIAGRAM 31

3 SEQUENCE DIAGRAM 32

4 ALOGRITHM DIAGRAM 6

5 MODEL DIAGRAM 29

REAL TIME FACIAL EXPRESSION BASED SMART MUSIC PLAYER

ABSTRACT

The face is an important aspect in predictinghuman emotions and mood.Usuallythe human emotions are extracted with the use of camera. There are many applications getting developed based on detection of human emotions. Few applications of emotion detection are business notification recommendation, e-learning, mental disorder and depression detection, criminal behaviour detection etc. In this proposed system, we develop a prototype in recommendation of dynamic music recommendation system based on human emotions. Based on each human listening pattern, the songs for each emotions are trained. Integration of feature extraction and machine learning techniques, from the real face the emotion are detected and once the mood is derived from the input image, respective songs for the specific mood would be played to hold the users.In this approach, the application gets connected with human feelings thus giving a personal touch to the users.Therefore our projected system concentrate on identifying the human feelings for developing emotion based music player using computervision and machine learning techniques. For experimental results, we use openCV for emotion detection and music recommendation.

CHAPTER 1

INTRODUCTION

People tend to express their emotions, mainly by their facial expressions. Music has always been known to alter the mood of an individual. Capturing and recognizing the emotion being voiced by a person and displaying appropriate songs matching the one's mood and can increasingly calm the mind of a user and overall end up giving a pleasing effect. The project aims to capture the emotion expressed by a person through facial expressions. A music player is designed to capture human emotion through the web camera interface available on computing systems. The software captures the image of the user and then with the help of image segmentation and image processing techniques extracts features from the face of a target human being and tries to detect the emotion that the person is trying to express. The project aims to lighten the mood of the user, by playing songs that match the requirements of the user by capturing the image of the user. Since ancient times the best form of expression analysis known to humankind is facial expression recognition. The best possible way in which people tend to analyze or conclude the emotion or the feeling or the thoughts that another person is trying to express is by facial expression. In some cases, mood alteration may also help in overcoming situations like depression and sadness. With the aid of expression analysis, many health risks can be avoided, and also there can be steps taken that help brings the mood of a user to a better stage.

1.1 PROPOSED ALGORITHMS

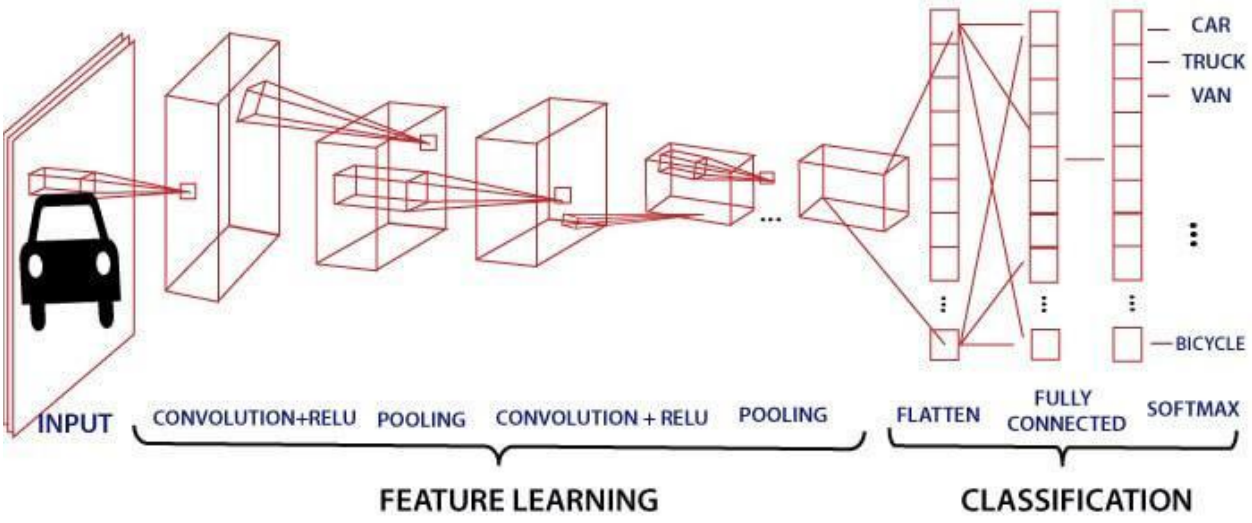
CNN ALGORITHM

Convolutional Neural Network is one of the main categories to do image classification and image recognition in neural networks. Scene labeling, objects detections, and face recognition, etc., are some of the areas where convolutional neural networks are widely used.

CNN takes an image as input, which is classified and process under a certain category such as dog, cat, lion, tiger, etc. The computer sees an image as an array of pixels and depends on the resolution of the image. Based on image resolution, it will see as **h * w * d**, where h= height w= width and d= dimension. For example,

An RGB image is **6 * 6 * 3** array of the matrix, and the grayscale image is **4 * 4 * 1** array of the matrix.

In CNN, each input image will pass through a sequence of convolution layers along with pooling, fully connected layers, filters (Also known as kernels). After that, we will apply the Soft-max function to classify an object with probabilistic values 0 and 1.



Convolution Layer

Convolution layer is the first layer to extract features from an input image. By learning image features using a small square of input data, the convolutional layer preserves the relationship between pixels. It is a mathematical operation which takes two inputs such as image matrix and a kernel or filter.

- The dimension of the image matrix is $\mathbf{h \times w \times d}$.
- The dimension of the filter is $\mathbf{f_h \times f_w \times d}$.
- The dimension of the output is $\mathbf{(h-f_h+1) \times (w-f_w+1) \times 1}$.

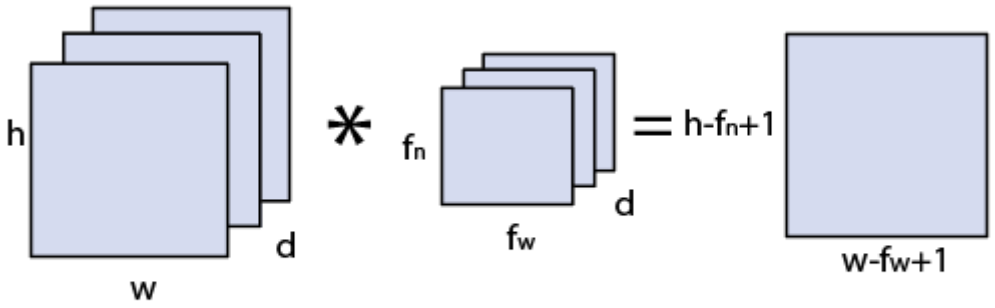


Image matrix multiplies kernl or filter matrix

Let's start with consideration a 5*5 image whose pixel values are 0, 1, and filter matrix 3*3 as:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

5×5 – Image Matrix 3×3 – Filter Matrix

The convolution of 5*5 image matrix multiplies with 3*3 filter matrix is called "**Features Map**" and show as an output.

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

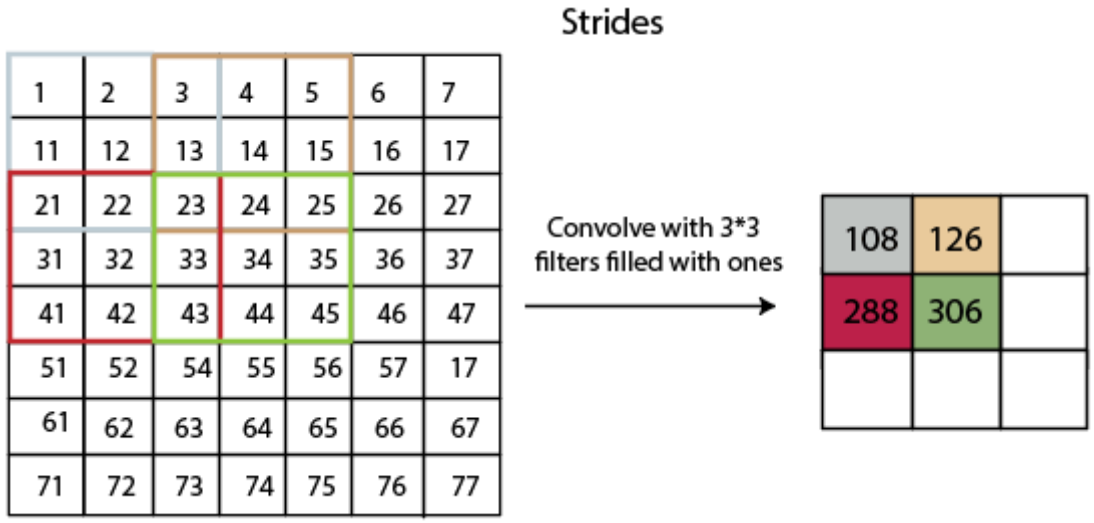
Convolved Feature

Convolution of an image with different filters can perform an operation such as blur, sharpen, and edge detection by applying filters.

Strides

Stride is the number of pixels which are shift over the input matrix. When the stride is equaled to 1, then we move the filters to 1 pixel at a time and similarly, if

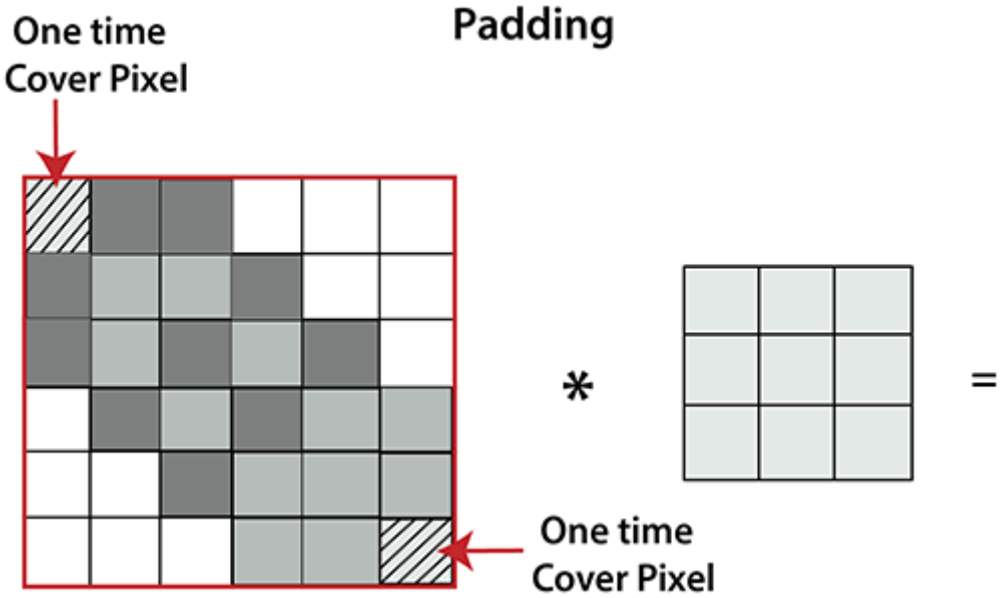
the stride is equaled to 2, then we move the filters to 2 pixels at a time. The following figure shows that the convolution would work with a stride of 2.



Padding

Padding plays a crucial role in building the convolutional neural network. If the image will get shrink and if we will take a neural network with 100's of layers on it, it will give us a small image after filtered in the end.

If we take a three by three filter on top of a grayscale image and do the convolving then what will happen?



It is clear from the above picture that the pixel in the corner will only get covers one time, but the middle pixel will get covered more than once. It means that we have more information on that middle pixel, so there are two downsides:

- Shrinking outputs
- Losing information on the corner of the image.

To overcome this, we have introduced padding to an image. **"Padding is an additional layer which can add to the border of an image."**

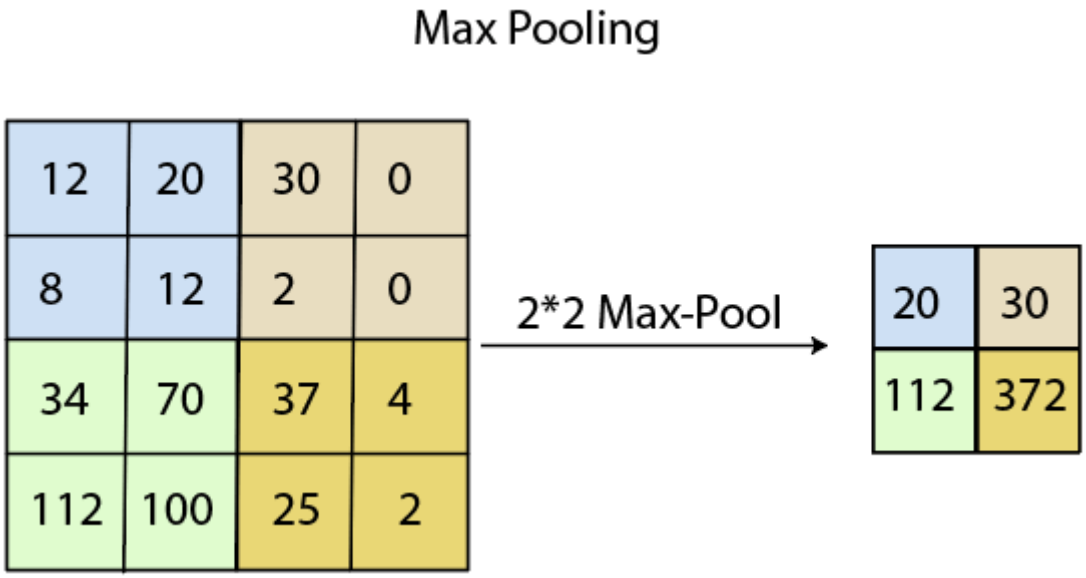
Pooling Layer

Pooling layer plays an important role in pre-processing of an image. Pooling layer reduces the number of parameters when the images are too large. Pooling is "**downscaling**" of the image obtained from the previous layers. It can be compared to shrinking an image to reduce its pixel density. Spatial pooling is also called downsampling or subsampling, which reduces the dimensionality of each map but retains the important information. There are the following types of spatial pooling:

Max Pooling

Max pooling is a **sample-based discretization process**. Its main objective is to downscale an input representation, reducing its dimensionality and allowing for the assumption to be made about features contained in the sub-region binned.

Max pooling is done by applying a max filter to non-overlapping sub-regions of the initial representation.



Average Pooling

Down-scaling will perform through average pooling by dividing the input into rectangular pooling regions and computing the average values of each region.

Syntax

layer = averagePooling2dLayer(poolSize)

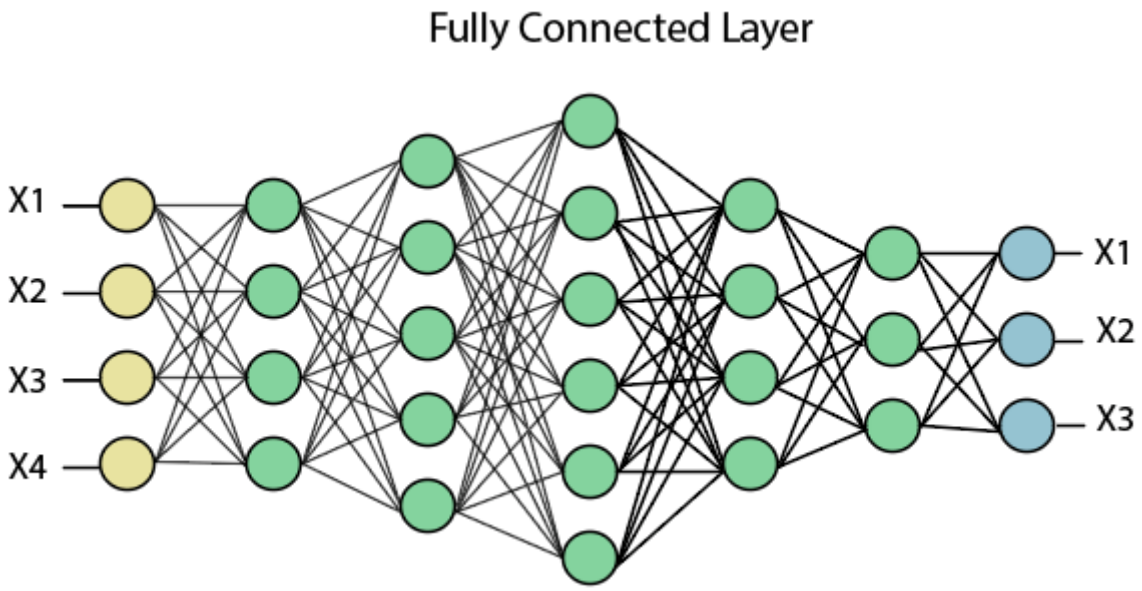
layer = averagePooling2dLayer(poolSize,Name,Value)

Sum Pooling

The sub-region for **sum pooling** or **mean pooling** are set exactly the same as for **max-pooling** but instead of using the max function we use sum or mean.

Fully Connected Layer

The fully connected layer is a layer in which the input from the other layers will be flattened into a vector and sent. It will transform the output into the desired number of classes by the network.



In the above diagram, the feature map matrix will be converted into the vector such as **x1, x2, x3... xn** with the help of fully connected layers. We will combine features to create a model and apply the activation function such as **softmax** or **sigmoid** to classify the outputs as a car, dog, truck, etc.

CHAPTER 2
LITERATURE SURVEY

Smart Music Player Integrating Facial Emotion Recognition

Songs, as a medium, have always been a popular choice to depict human emotions. We validate our models by creating a real-time vision system which accomplishes the tasks of face detection and emotion classification simultaneously in one blended step using our proposed CNN architecture. Reliable emotion-based classification systems can go a long way in facilitating emotions. However, research in the field of emotionbased music classification has not yielded optimal results. In this paper, we present an affective cross-platform music player, EMP, which recommends music based on the real-time mood of the user. EMP provides smart mood-based music recommendation by incorporating the capabilities of emotion context reasoning within our adaptive music recommendation system. Our music player contains three modules: Emotion Module, Music Module and Integrating Module. The Emotion Module takes an image of the user as an input and makes use of deep learning algorithms to identify the mood of the user with an

accuracy of 90.23%.

Mood based Music Recommendation System

A user's emotion or mood can be detected by his/her facial expressions. These expressions can be derived from the live feed via the system's camera. A lot of research is being conducted in the field of Computer Vision and Machine Learning (ML), where machines are trained to identify various human emotions or moods. Machine Learning provides various techniques through which human emotions can be detected. One such technique is to use MobileNet model with Keras, which generates a small size trained model and makes Android-ML integration easier.Music is a great connector. It unites us across markets, ages, backgrounds, languages, preferences, political leanings and income levels. Music players and other streaming apps have a high demand as these apps can be used anytime, anywhere and can be combined with daily activities, travelling, sports, etc. With the rapid development of mobile networks and digital multimedia technologies, digital music has become the mainstream consumer content sought by many young people.People often use music as a means of mood regulation, specifically to change a bad mood, increase energy level or reduce tension. Also, listening to the right kind of music at the right time may improve mental health. Thus, human emotions have a strong relationship with music.In our proposed system, a mood-based music player is created which performs real time mood detection and suggests songs as per detected mood. This becomes an additional feature to the traditional music player apps that come pre-installed in our mobile phones. An important benefit of incorporating mood detection is customer satisfaction. The objective of this system is to analyse the users image, predict the expression of the user and suggest songs suitable to the detected mood.

An Emotion-Aware Personalized Music Recommendation System Using a Convolutional Neural Networks Approach

Recommending music based on a user's music preference is a way to improve user listening experience. Finding the correlation between the user data (e.g., location, time of the day, music listening history, emotion, etc.) and the music is a challenging task. In this paper, we propose an emotion-aware personalized music

recommendation system (EPMRS) to extract the correlation between the user data and the music. To achieve this correlation, we combine the outputs of two approaches: the deep convolutional neural networks (DCNN) approach and the weighted feature extraction (WFE) approach. The DCNN approach is used to extract the latent features from music data (e.g., audio signals and corresponding metadata) for classification. In the WFE approach, we generate the implicit user rating for music to extract the correlation between the user data and the music data. In the WFE approach, we use the term-frequency and inverse document frequency (TF-IDF) approach to generate the implicit user ratings for the music. Later, the EPMRS recommends songs to the user based on calculated implicit user rating for the music. We use the million songs dataset (MSD) to train the EPMRS. For performance comparison, we take the content similarity music recommendation system (CSMRS) as well as the personalized music recommendation system based on electroencephalography feedback (PMRSE) as the baseline systems. Experimental results show that the EPMRS produces better accuracy of music recommendations than the CSMRS and the PMRSE. Moreover, we build the Android and iOS APPs to get realistic data of user experience on the EPMRS. The collected feedback from anonymous users also show that the EPMRS sufficiently reflect their preference on music.

An efficient realtime emotion detection using camera and facial landmarks

Emotion recognition has many useful applications in daily lives. In this paper, we present a potential approach to detect human emotion in real time. For any face detected in camera, we extract the corresponding facial landmarks and examine different kinds of features and models for predicting human emotion. The experiments show that our proposed system can naturally detect human emotion in real time and achieve an average accuracy about 70.65%.

Deep Learning in Music Recommendation Systems

Like in many other research areas, deep learning (DL) is increasingly adopted in music recommendation systems (MRS). Deep neural networks are used in this

domain particularly for extracting latent factors of music items from audio signals or metadata and for learning sequential patterns of music items (tracks or artists) from music playlists or listening sessions. Latent item factors are commonly integrated into content-based filtering and hybrid MRS, whereas sequence models of music items are used for sequential music recommendation, e.g., automatic playlist continuation. This review article explains particularities of the music domain in RS research. It gives an overview of the state of the art that employs deep learning for music recommendation. The discussion is structured according to the dimensions of neural network type, input data, recommendation approach (content-based filtering, collaborative filtering, or both), and task (standard or sequential music recommendation). In addition, we discuss major challenges faced in MRS, in particular in the context of the current research on deep learning.

Review on Facial Expression Based Music Player

Human often use nonverbal cues such as hand gestures, facial expressions, and tone of the voice to express feelings in interpersonal communications. The face of the human is an important organ of an individuals body and it plays an important role in extraction of an individualsbehavior and emotional state. Facial expression provides current mind state of person. It is very time consuming and difficult to create and manage large playlists and to select songs from these playlists. Thus, it would be very helpful if the music player itself selects a song according to the current mood of the user. Manually segregating the list of songs associated, generating acceptable playlist supported an individuals emotions could be a terribly tedious, time overwhelming, intensive and upheld task Thus, an application can be developed to minimize these efforts of managing playlists. However the proposed existing algorithms in use are computationally slow and less accurate. This proposed system based on facial expression extracted will

generate a playlist automatically thereby reducing the effort and time involved in rendering the process manually. Facial expressions are given using inbuilt camera. The image is captured using camera and that image is passed under different stages to detect the mood or emotion of the user. We will study about how to automatically detect the mood of the user and present him a playlist of songs which is suitable for his current mood. Proposed paper has used Viola-Jones algorithm and multiclass SVM (Support Vector Machine) for face detection and emotion detection respectively.

Smart Music Player Based on Emotion Recognition from Facial Expression

The magical power of music is scientifically proven. People always like to hear the music depending on their emotional feelings. Music is considered to be a tool for stress relief. Many psychological states can be very well controlled by listening to music. . We focus on developing an emotion based music system. The image of the face is captured in a camera and the emotions are classified. The classification is done using CNN classifier. Theneural network model is trained and used to find the emotion from the image of the face captured .Depending on the mood of the user a playlist is formed in the music player implemented using PyQt5.

An emotion based music player for Android

Music plays a very important role in human's daily life and in the modern advanced technologies. Usually, the user has to face the task of manually browsing through the playlist of songs to select. Here we are proposing an efficient and accurate model, that would generate a playlist based on current emotional state and behavior of the user. Existing methods for automating the playlist generation process are computationally slow, less accurate and sometimes even require use of additional hardware like EEG or sensors. Speech is the most ancient and natural way of expressing feelings, emotions and mood and its and its processing requires high computational, time, and cost. This proposed system based on real-time extraction of facial expressions as well as extracting audio features from songs to classify into a specific emotion that will generate a playlist automatically such that

the computation cost is relatively low.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

System : Pentium i3 Processor

Hard Disk : 500 GB.

Monitor : 15'' LED

Input Devices : Keyboard, Mouse

Ram : 2 GB

3.2 SOFTWARE REQUIREMENTS

Operating system : Windows 10

Coding Language : Python

3.3 LANGUAGE SPECIFICATION

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU

General Public License (GPL). This **tutorial** gives enough understanding on **Python programming** language.

3.4. HISTORY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

3.5. APPLICATION OF PYTHON

Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

Easy-to-read − Python code is more clearly defined and visible to the eyes.

Easy-to-maintain − Python's source code is fairly easy-to-maintain.

A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

Databases − Python provides interfaces to all major commercial databases.

GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Scalable − Python provides a better structure and support for large programs than shell scripting.

3.6 FEATURES OF PYTHON

It supports functional and structured programming methods as well as OOP.

It can be used as a scripting language or can be compiled to byte-code for building large applications.

It provides very high-level dynamic data types and supports dynamic type checking.

It supports automatic garbage collection.

It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

3.7 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

3.7.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures mustjustified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.7.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.7.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

CHAPTER 4

SYSTEM ANALYSIS

4.1 PURPOSE

The purpose of this document is real time facial expression based music recommender system using machine learning algorithms. In detail, this document will provide a general description of our project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project - such as interface, functional requirements and performance requirements.

4.2 SCOPE

The scope of this SRSdocument persists for the entire life cycle of the project. This document defines the final state of the software requirements agreed upon by the customers and designers. Finally at the end of the project execution all the functionalities may be traceable from the SRSto the product. The document describes the functionality, performance, constraints, interface and reliability for the entire life cycle of the project.

4.3 EXISTING SYSTEM

Nikhil et al. determines the mindset of the user by using facial expression Humans often express their feeling by their expressions, hand gestures, and by raising the voice of tone but mostly humans express their feelings by their face. Emotion-based music player reduces the time complexity of the user. Generally, people have a large number of songs on their playlist. Playing songs randomly does not satisfy the mood of the user. This system helps user to play songs automatically according to their mood. The image of the user is captured by the web camera, and the images are saved. The images are first converted from RGB to binary format. This process of representing the data is called a feature-point detection method. This process can also be done by using Haar Cascade technology provided by Open CV. The music player is developed by using a java program. It manages the database and plays the song according to the mood of the user.

4.4 PROPOSED SYSTEM

The proposed system can detect the facial expressions of the user and based on his/her facial expressions extract the facial landmarks, which would then be classified to get a particular emotion of the user. Once the emotion has been classified the songs matching the user's emotions would be shown to the user. In this proposed system, we develop a prototype in recommendation of dynamic music recommendation system based on human emotions. Based on each human listening pattern, the songs for each emotions are trained. Integration of feature extraction and machine learning techniques, from the real face the emotion are detected and once the mood is derived from the input image, respective songs for the specific mood would be played to hold the users. In this approach, the application gets connected with human feelings thus giving a personal touch to the

users. Therefore our projected system concentrate on identifying the human feelings for developing emotion based music player using computer vision and machine learning techniques. For experimental results, we use openCV for emotion detection and music recommendation.

CHAPTER 5

SYSTEM DESIGN

5.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.

- Methods for preparing input validations and steps to follow when error occur.

5.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action

5.3 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

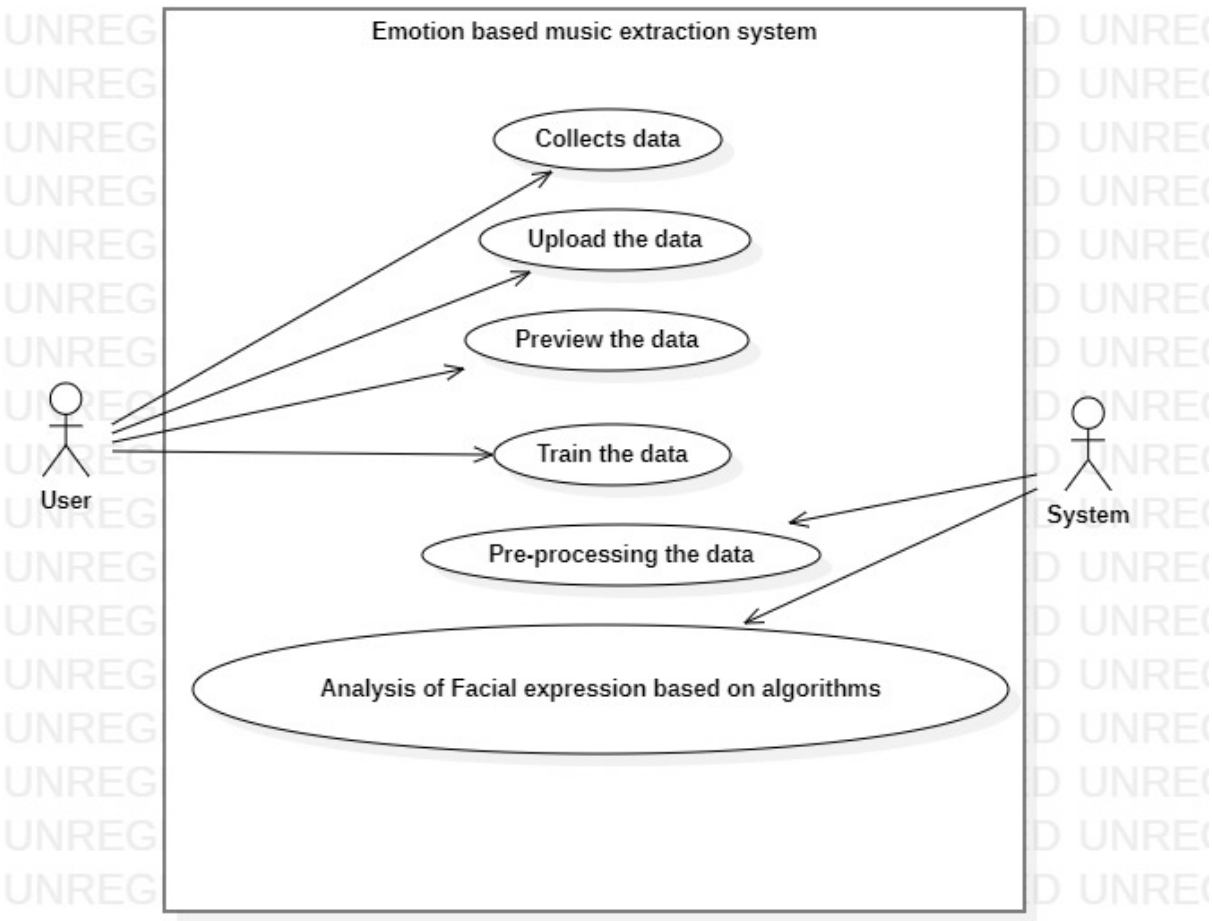
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.

5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

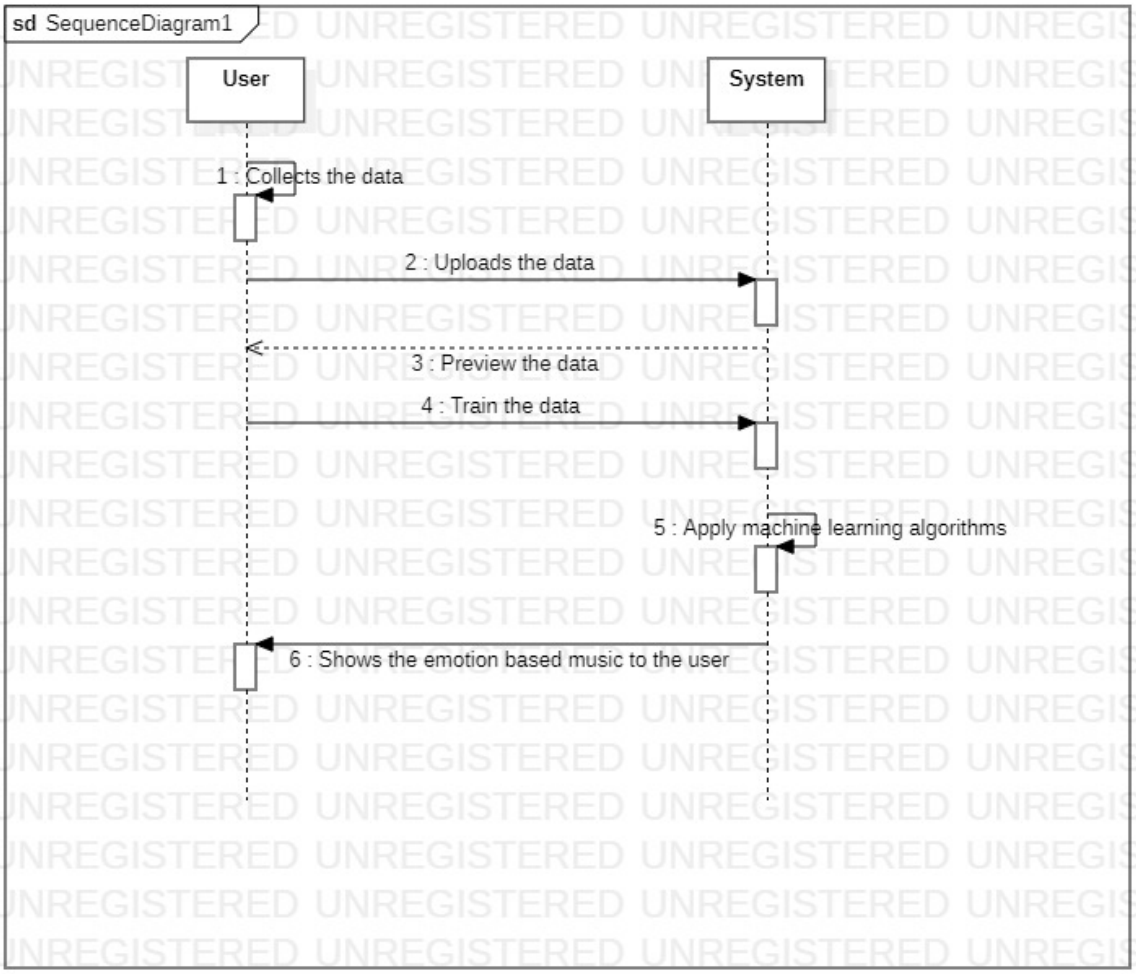
USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



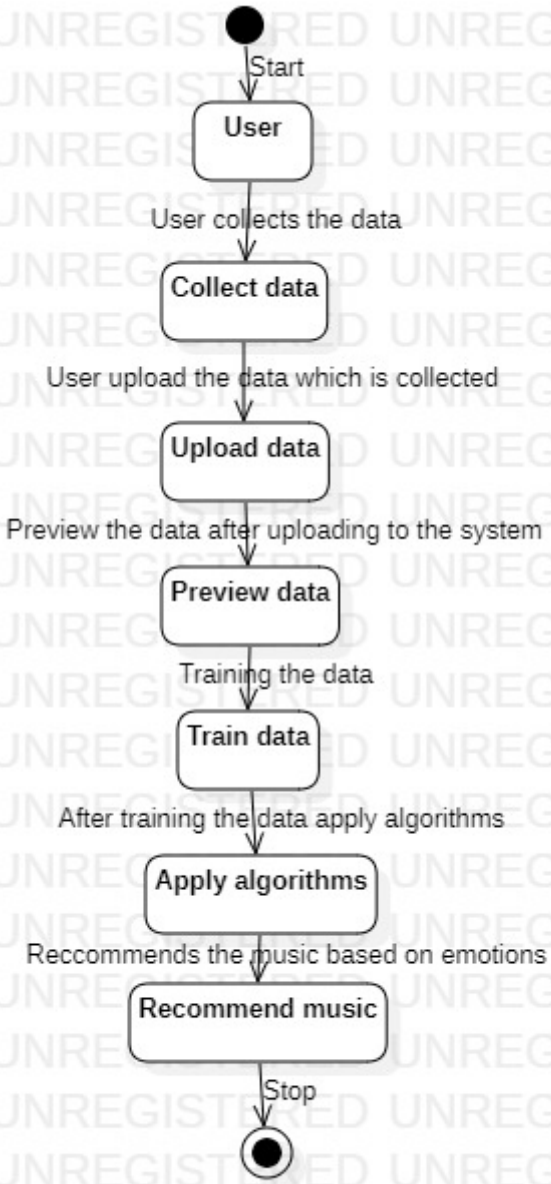
SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.


Start
User
User collects the data
Collect data
User upload the data which is collected
Upload data
Preview the data after uploading to the system
Preview data
Training the data
Train data
After training the data apply algorithms
Apply algorithms
Reccommends the music based on emotions
Recommend music
Stop

CHAPTER 6
MODULES

MODULES

Data Collection Module

Emotion Extraction Module

Audio Extraction Module

Emotion - Audio Integration Module

MODULE DESCRIPTION

Data Collection Module

A survey was collected from users based on 3 parameters which are, 1. What type of songs would they want to listen to when they are happy? 2. What type of songs would they want to listen to when they are sad? 3. What type of songs would they want to listen to when they are angry.

Emotion Extraction Module

The image of the user is captured with the help of a camera/webcam. Once the picture captured, the frame of the captured image from webcam feed is converted to a grayscale image to improve the performance of the classifier, which is used to identify the face present in the picture. Once the conversion is complete, the image is sent to the classifier algorithm which, with the help of feature extraction techniques can extract the face from the frame of the web camera feed. From the extracted face, individual features are obtained and are sent to the trained network to detect the emotion expressed by the user. These images will be used to train the classifier so that when a completely new and unknown set of images is presented to the classifier, it is able to extract the position of facial landmarks from those images based on the knowledge that it had already acquired from the training set and return the coordinates of the new facial landmarks that it detected. The network is trained with the help of extensive data set. This is used to identify the emotion being voiced by the user.

Audio Extraction Module

After the emotion of the user is extracted the music/audio based on the emotion voiced by the user is displayed to the user, a list of songs based on the emotion is displayed, and the user can listen to any song he/she would like to. Based on the regularity that the user would listen to the songs are displayed in that order.

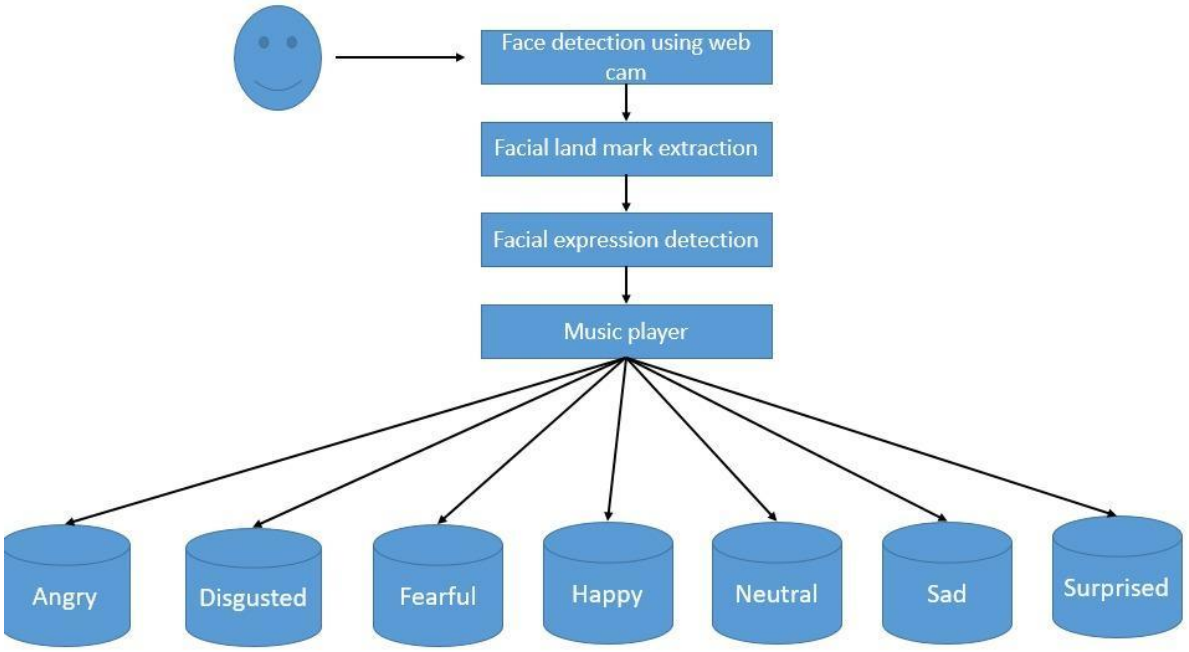
Emotion - Audio Integration Module

The emotions which are extracted for the songs are stored, and the songs based on the emotion are displayed on the web page. For example, if the emotion or the facial feature is categorized under happy, then songs from the happy database are displayed to the user.

CHAPTER 7
SYSTEM IMPLEMENTATION

7.1 SYSTEM ARCHITECTURE

Describing the overall features of the software is concerned with defining the requirements and establishing the high level of the system. During architectural design, the various web pages and their interconnections are identified and designed. The major software components are identified and decomposed into processing modules and conceptual data structures and the interconnections among the modules are identified. The following modules are identified in the proposed system.



CHAPTER 8

SYSTEM TESTING

8.1 Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

8.2 Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

8.3 Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

8.4 Basics of software testing

There are two basics of software testing: black box testing and white box testing.

8.5 Black box Testing

Black box testing is a testing technique that ignores the internal mechanism

of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

8.6 White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

8.7 Types of testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

8.7.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

8.7.2 Integration Testing

Integration testing is testing in which a group of components are combined

to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

8.7.3 Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

8.7.4 System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

8.7.5 Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

8.7.6 Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

8.7.7 Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how

to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

8.7.8 Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

8.7.9 Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing

REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired

response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that is has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one’s personal information.

NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

CHAPTER 9
CONCLUSION

In this paper, we have discussed that how our proposed systemrecommend music based on facial expression using machine learning algorithms. The proposed system is also scalable for recommending music based on facial expressions by using techniques after collecting data. The system is not having complex process to recommend music that the data like the existing system. Proposed system gives genuine and fast result than existing system. Here in this system we use machine learning algorithms to recommend music based on real time facial expression.

REFERENCES

[1] Emanuel I. Andelin and Alina S. Rusu,"Investigation of facial microexpressions of emotions in psychopathy - a case study of an individual in detention", 2015, Published by Elsevier Ltd.

[2] Paul Ekman, Wallace V Friesen, and Phoebe Ellsworth. Emotion in the human face: Guidelines for research and an integration of findings. Elsevier 2013.

[3] F. De la Torre and J. F. Cohn, "Facial expression analysis," Vis. Anal. Hum., pp. 377–410, 2011.

[4] Bavkar, Sandeep, Rangole, Jyoti, Deshmukh," Geometric Approach for Human Emotion Recognition using Facial Expression", International Journal of Computer Applications, 2015.

[5] Zhang, Z. Feature-based facial expression recognition: Sensitivity analysis and experiments with a multilayer perceptron. International Journal of Patten Recognition and Artificial Intelligence.

[6] Remi Delbouys, Romain ´ Hennequin, Francesco Piccoli, Jimena RoyoLetelier, Manuel Moussallam. "Music mood detection based on audio.

[7] nd lyrics with Deep Neural Net", 19th International Society for Music Information Retrieval Conference, Paris, France, 2018.

[8] KrittrinChankuptarat, etal, “Emotion Based Music Player”, IEEE 2019 conference.

[9] Kim, Y.: Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 2014 Conference on EMNLP, pp. 1746–1751 (2014).

[10] Tripathi, S., Beigi, H.: Multi-Modal Emotion recognition on IEMOCAP Dataset using Deep Learning. In: arXiv:1804.05788 (2018).

[11] Teng et al.,”Recognition of Emotion with SVMs”, Lecture Notes in Computer Science, August 2006.

[12] B.T. Nguyen, M.H. Trinh, T.V. Phan, H.D. NguyenAn efficient realtime emotion detection using camera and facial landmarks , 2017 seventh international conference on information science and technology (ICIST) (2017)

SAMPLE CODE

```
from keras.preprocessing.image import img_to_array
import imutils
import cv2
from keras.models import load_model
import numpy as np
import tensorflow as tf
from tensorflow.python.keras.layers import  Input, Embedding, Dot, Reshape,
Dense
from tensorflow.python.keras.models import Model
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","sad","happy","scared","surprised","neutral"]


#feelings_faces = []
#for index, emotion in enumerate(EMOTIONS):
  # feelings_faces.append(cv2.imread('emojis/' + emotion + '.png', -1))

# starting video streaming
cv2.namedWindow('your_face')
camera = cv2.VideoCapture(0)
#while True:
for i in range(0,200):
    frame = camera.read()[1]
    #reading the frame
    frame = imutils.resize(frame,width=300)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces =
face_detection.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSi
ze=(30,30),flags=cv2.CASCADE_SCALE_IMAGE)

    canvas = np.zeros((250, 300, 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)


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


#    try:
        for (i, (emotion, prob)) in enumerate(zip(EMOTIONS, preds)):
            # construct the label text

            if prob > 0.3:
                print(emotion)
                text = "{}: {:.2f}%".format(emotion, prob * 100)
                print(text)
                # 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, 0, 255), -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, 0, 255),
2)
                cv2.rectangle(frameClone, (fX, fY), (fX + fW, fY + fH),
                              (0, 0, 255), 2)

                cv2.imshow('your_face', frameClone)
                cv2.imshow("Probabilities", canvas)
                if cv2.waitKey(1) & 0xFF == ord('q'):
                    break



                if emotion == 'happy':


                    playsound(r'C:/Users/Kumara Swamy/Documents/facial emotion
final/songs/happy/muthumazhyay.mp3')
                    playsound(r'C:/Users/Kumara Swamy/Documents/facial emotion
final/songs/happy/koodemele.mp3')

                    camera.release()
                    cv2.destroyAllWindows()
                    time.sleep(20)
                    break

                if emotion == 'sad':
```

```
                playsound(r'C:/Users/Kumara Swamy/Documents/facial emotion
final/songs/sad/enn-kathalle.mp3')
                camera.release()
                cv2.destroyAllWindows()
                time.sleep(20)
                break
            if emotion == 'neutral':


                playsound(r'C:/Users/Kumara Swamy/Documents/facial emotion
final/songs/neutral/kanna-nee-thoogada.mp3')
                camera.release()
                cv2.destroyAllWindows()
                time.sleep(20)
                break

            if emotion == 'scared':


                playsound(r'C:/Users/Kumara Swamy/Documents/facial emotion
final/songs/scared/Chandramukhi.mp3')
                camera.release()
                cv2.destroyAllWindows()
                time.sleep(20)
                break

            if emotion == 'surprised':


                playsound(r'C:/Users/Kumara Swamy/Documents/facial emotion
final/songs/surprised/poona-usiru.mp3')
                camera.release()
                cv2.destroyAllWindows()
                time.sleep(20)
                break

            if emotion == 'angry':


                playsound(r'C:/Users/Kumara Swamy/Documents/facial emotion
final/songs/angry/Kalippu-Premam.mp3')
                camera.release()
                cv2.destroyAllWindows()
                time.sleep(20)
                break

            if emotion == 'fear':


                playsound('SONGS/fear/agayam.mp3')
                camera.release()
                cv2.destroyAllWindows()
                time.sleep(20)
                break
```

```
#    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)
#    except :
#        er = 'error'
```