**SPOTIFY GENRE RECOMMENDATION BASED ON USER EMOTION USING DEEP LEARNING**

**A PROJECT REPORT SUBMITTED TO**

**SRM INSTITUTE OF SCIENCE & TECHNOLOGY**

**IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE**

**AWARD OF THE DEGREE OF**

**MASTER OF SCIENCE IN APPLIED DATA SCIENCE**

**BY**

**SUHASINI V**

**REG NO. RA2232014010101**

**UNDER THE GUIDANCE OF**

**Dr. SELVAM.L, MCA., M.Phil., Ph.D.,**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**FACULTY OF SCIENCE AND HUMANITIES**

**SRM INSTITUTE OF SCIENCE & TECHNOLOGY**

**Kattankulathur – 603 203**

**Chennai, Tamil Nadu**

**April-2024**

**BONAFIDE CERTIFICATE**

This is to certify that the project report titled **“SPOTIFY GENRE RECOMMENDATION BASED ON USER EMOTION USING DEEP LEARNING”** is a bonafide work carried out by **SUHASINI V (RA2232014010101)** under my supervision for the award of the Degree of Master of science in Applied Data Science. To my knowledge the work reported herein is the original work done by this student.

**Dr. Selvam. L Dr. S. Albert Antony Raj**

Assistant Professor, Deputy Dean & Head,

Department of Computer Applications Department of Computer Applications

(GUIDE)

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**DECLARATION OF ASSOCIATION OF RESEARCH PROJECT WITH SUSTAINABLE DEVELOPMENT GOALS**

This is to certify that the research project entitled **“SPOTIFY GENRE RECOMMENDATION BASED ON USER EMOTION USING DEEP LEARNING”** carried out by **Ms. SUHASINI V** under the supervision of **Dr. Selvam. L** in partial fulfilment of the requirement for the award of post-graduation program has been significantly or potentially associated with SDG Goal No **16 (SIXTEEN)** titled **PEACE, JUSTICE AND STRONG INSTITUTION**

This study has clearly shown the extent to which its goals and objectives have been met in terms of filling the research gaps, identifying needs, resolving problems, and developing innovative solutions locally for achieving the above-mentioned SDG on a National and/or on an international level.

**SIGNATURE OF THE STUDENT**  **GUIDE**

**HEAD OF THE DEPARTMENT**

**ACKNOWLEDGEMENT**

With profound gratitude to the ALMIGHTY, I take this chance to thank people who helped me to complete this project.

I take this as a right opportunity to say THANKS to my parents who are there to stand with me always with the words “YOU CAN”.

I wish to express my sincere gratitude to **Dr.T.R.Paarivendhar**, Chancellor, SRM Institute of Science & Technology who gave us the platform to establish me to reach greater heights.

I am thankful to **Prof. A.Vinay Kumar**, Pro Vice-Chancellor (FSH, Management, Law) for his unwavering support throughout my project.

I earnestly thank **Dr. A.Duraisamy**, Dean, Faculty of Science and Humanities, SRM Institute of Science & Technology who always encourage me to do novel things.

I express my sincere thanks to **Dr. S. Albert Antony Raj,** Deputy Dean and Head, Department of Computer Applications, Faculty of Science and Humanities for his valuable guidance and support to execute all incline in learning.

It is my delight to thank my project guide **Dr. Selvam. L,** Assistant Professor, Department of Computer Applications for her help, support, encouragement, suggestions, and guidance throughout the development phases of the project.

I convey my gratitude to all the faculty members of the department who extended their support through valuable comments and suggestions during the reviews.

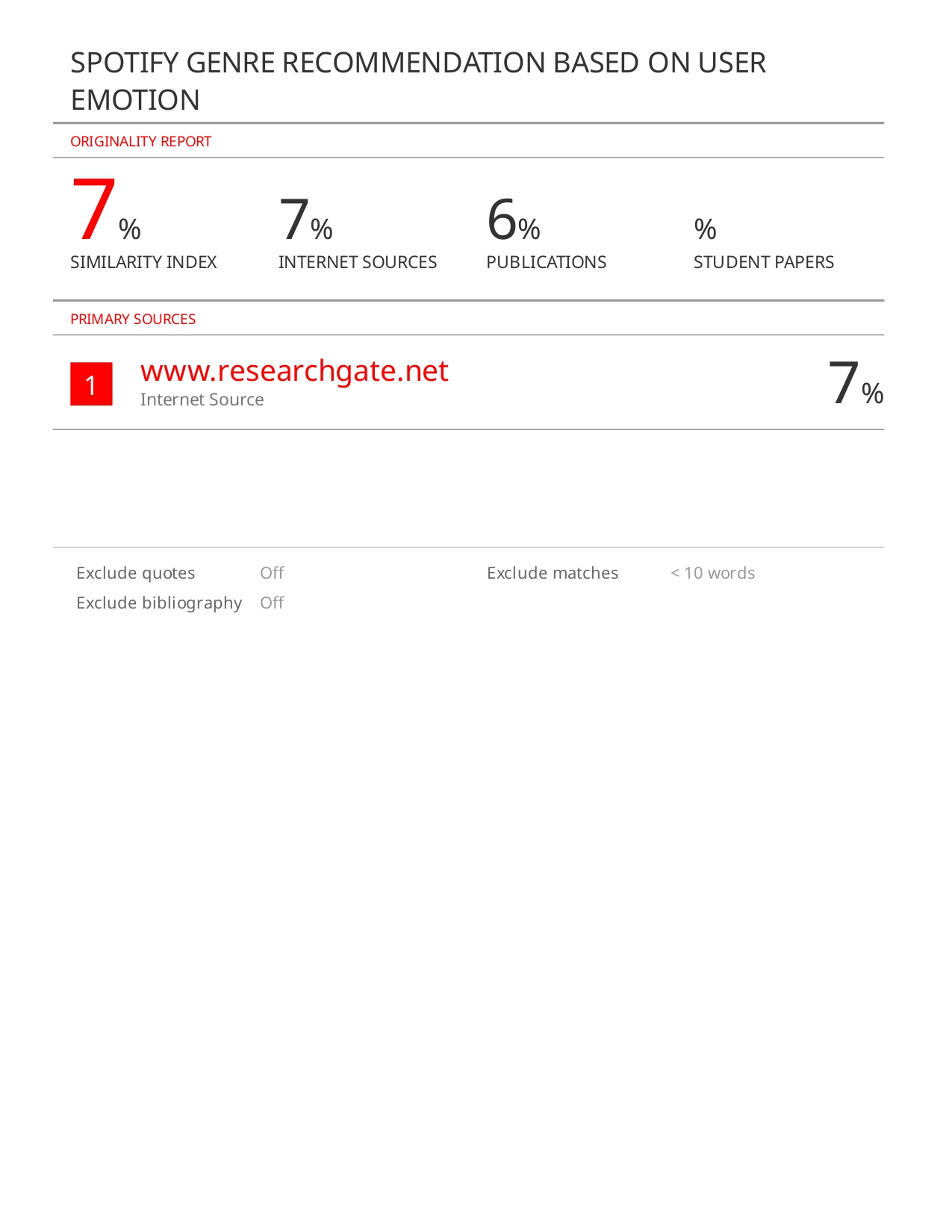
A great note of gratitude to friends and people who are known and unknown to me who helped in carrying out this project work a successful one.

**SUHASINI V**

**COMPANY LETTER**

****

**PLAGIARISM REPORT**

****

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | **3** |
| **01** | **INTRODUCTION**   * 1. OVERVIEW   2. IMAGE PROCESSING | **4** |
| **02** | **LITERATURE SURVEY** | **11** |
| **03** | **SYSTEM ANALYSIS**  3.1 EXISTING SYSTEM  -DRAWBACKS  3.2 PROPOSED SYSTEM  -ADVANTAGE | **17** |
| **04** | **SYSTEM DESGIN**  4.1 SYSTEM ARCHITECTURE  4.2UML DIAGRAM  4.3 USE CASE DIAGRAM  4.4 CLASS DIAGRAM  4.5 SEQUENCE DIAGRAM  4.6 DATAFLOW DIAGRAM | **19** |
| **05** | **SYSTEM IMPLEMENTATAION**  5.1MODULE LIST  5.2MODULE DESCRIPTION | **21** |
| **06** | **ALGORTHIM**  6.1 INTRODUCTION  6.2 WORKING PROCESS | **26** |
| **07** | **SYSTEM TESTING**  7.1 TYPES OF TESTS  7.2 UNIT TESTING  7.3 INTEGRATRATED TESTING  7.4 ACCEPTANCE TESTING | **28** |
| **08** | **SYSTEM REQUIREMENT**  8.1 PYTHON  8.2 IMPLEMENTATION  8.3 OTHER IMPLEMENTATION | **32** |
| **9** | **APPENDICES**  10.1 CODING  10.2 SCREENSHOTS | **37** |
| **10** | **CONCLUSION** | **46** |
| **11** | **FUTURE WORK** | **47** |
| **12** | **REFERENCES** | **48** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **TITLE** | **PAGE.NO** |
| **01** | **SYSTEM ARCHITECTURE** | **19** |
| **02** | **UML DIAGRAM** | **19** |
| **03** | **USE CASE DIAGRAM** | **20** |
| **04** | **CLASS DIAGRAM** | **21** |
| **05** | **SEQUENCE DIAGRAM** | **20** |
| **06** | **DATAFLOW DIAGRAM** | **21** |

**SPOTIFY GENRE RECOMMENDATION BASED ON USER EMOTION USING DEEP LEARNING**

**ABSTRACT**

This paper introduces an innovative machine learning approach for musical therapy that harnesses facial expression recognition technology to personalize musical experiences. The system analyzes real-time facial expressions to determine an individual's emotional state and dynamically adjusts musical compositions accordingly. By leveraging this technology, the system can generate or modify melodies in response to the user's mood, crafting uplifting tunes for moments of joy or relaxation, and soothing melodies for times of melancholy or stress. This personalized approach enhances the therapeutic impact of music by closely aligning it with the individual's emotions, fostering relaxation, emotional release, and motivation. The music serves as a reflective mirror of the user's feelings, offering a unique and deeply engaging musical therapy experience tailored to their emotional needs. This novel application of facial expression recognition in musical therapy has the potential to revolutionize how music is used for emotional well-being and psychological support.

**CHAPTER 1**

**1.INTRODUCTION**

In recent years, the intersection of machine learning and healthcare has led to remarkable advancements in personalized therapeutic interventions. One particularly intriguing application lies in the realm of musical therapy, where the fusion of facial expression recognition technology with Convolutional Neural Network (CNN) promises to revolutionize the way music is utilized for emotional well-being. By employing CNN algorithms, this innovative approach seeks to dynamically tailor musical compositions to individuals' emotional states in real-time, based on the analysis of their facial expressions. This introduction outlines the potential of leveraging CNNs in musical therapy, offering a nuanced understanding of how this integration could enhance the therapeutic impact of music by creating personalized and deeply engaging experiences that resonate with individuals' emotions. The utilization of CNN algorithms in conjunction with facial expression recognition presents a paradigm shift in the field of musical therapy. Traditional approaches often rely on predetermined playlists or static compositions, lacking the adaptability to address the dynamic nature of human emotions. However, by integrating CNNs, which are adept at generating realistic and contextually relevant content, this approach enables the creation of music that evolves in response to the user's emotional cues. This introduction sets the stage for exploring the potential of CNN-based facial expression recognition in musical therapy, highlighting its capacity to unlock new dimensions of personalization and effectiveness in emotional support through music.

**1.1 OVERVIEW**

Ever since computers were invented, people have wanted to build artificially intelligent (AI) systems that are mentally and/or physically equivalent to humans. In the past decades, the increase of generally available computational power provided a helping hand for developing fast learning machines, whereas the Internet supplied an enormous amount of data for training. Among a lot of advanced machine learning techniques that have been developed so far, deep learning is widely considered as one of the most promising techniques to make AI machines approaching human-level intelligence.



Facial expression recognition is the process of identifying human emotion based on facial expressions for musical recommend. Humans are naturally capable of recognizing emotions. In fact, children, which are only 36 hours old, can interpret some very basic emotions from faces. In older humans, this ability is considered one of the most important social skills. There is a universality in facial expressions of humans in expressing certain emotions. Human develop similar muscular movements belonging to a certain mental state, despite their place of birth, race, education, etcetera. Therefore, if properly being modelled, this universality can be a convenient feature in human-machine interaction: a well-trained system can understand emotions, independent of who the subject is.

Automated facial expression recognition has numerous practical applications such as psychological analysis, medical diagnosis, forensics (lie-detection), studying effectiveness of advertisement and so on. The ability to read facial expressions and then recognize human emotions provides a new dimension to human-machine interactions, for instance, smile detector in commercial digital cameras or interactive advertisements. Robots can also benefit from automated facial expression recognition. If robots can predict human emotions, they can react upon this and have appropriate behaviors. This paper introduces a cutting-edge approach to musical therapy that leverages facial expression recognition technology integrated with Convolutional Neural Network (CNN). By analyzing real-time facial expressions, the system dynamically adjusts musical compositions to match individuals' emotional states. This innovative fusion of CNN algorithms with facial recognition holds the potential to revolutionize musical therapy, offering personalized and deeply engaging experiences tailored to individuals' emotional needs. Through this integration, music becomes a responsive medium, mirroring and addressing users' emotions in real-time, thereby enhancing the therapeutic impact of musical interventions.

**1.2 IMAGE PROCESSING**

**Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.**

* **Image processing basically includes the following three steps:**
* **Importing the image via image acquisition tools;**
* **Analysing and manipulating the image;**
* **Output in which result can be altered image or report that is based on image analysis.**

**There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction.**

**In this lecture we will talk about a few fundamental definitions such as image, digital image, and digital image processing. Different sources of digital images will be discussed and examples for each source will be provided. The continuum from image processing to computer vision will be covered in this lecture. Finally we will talk about image acquisition and different types of image sensors.**

* **Image Acquisition:** The process starts with acquiring an image using devices like digital cameras, scanners, or sensors. The quality and resolution of the acquired image are crucial.
* **Preprocessing:** This stage involves cleaning up the image, removing noise, correcting distortions, and enhancing its quality. Common preprocessing techniques include image denoising, contrast adjustment, and image resizing.
* **Image Enhancement:** Enhancement techniques aim to improve the visual quality of an image. These methods can sharpen edges, adjust brightness and contrast, and highlight certain features within the image.
* **Image Restoration:** Restoration techniques are used to recover or improve the original image from a degraded or damaged version. This can be useful in scenarios such as restoring old photographs or removing scratches and stains.
* **Image Segmentation:** Segmentation involves dividing an image into meaningful regions or objects. It is commonly used in object detection, medical image analysis, and more.
* **Feature Extraction:** In this step, relevant information is extracted from the image. This may include extracting specific patterns, shapes, or features for further analysis.
* **Object Recognition:** Object recognition techniques are used to identify and classify objects within an image. This is fundamental in applications like facial recognition, object tracking, and autonomous vehicles.
* **Pattern Matching:** Image processing can be used to find patterns or templates within an image, which can be useful in various fields such as character recognition or fingerprint analysis.

**IMAGE TYPES**

* Binary image
* Grayscale image
* Indexed image
* True color or RGB image

**Binary Image:**

A binary image is the simplest form of image representation, consisting of only two colors: black and white (or 0 and 1). Each pixel in the image is either completely black (0) or completely white (1). Binary images are commonly used in applications like image segmentation and object detection.

**Grayscale Image:**

In a grayscale image, each pixel is represented by a single intensity value ranging from 0 (black) to 255 (white) in an 8-bit depth. The shades of gray are used to represent variations in intensity. Grayscale images are commonly used in applications where color information is not necessary, such as medical imaging and certain types of image processing.

**Indexed Image:**

In an indexed image, the color information is stored in a separate color map or palette. Instead of representing each pixel with RGB values, each pixel contains an index that corresponds to a color in the palette. This reduces the amount of data required to represent the image. Indexed images are often used in applications where memory efficiency is crucial, such as in GIF images.

**True Color or RGB Image:**

True color images, also known as RGB (Red, Green, Blue) images, represent each pixel with three color channels: red, green, and blue. Each channel can have intensity values ranging from 0 to 255, resulting in a wide spectrum of colors. True color images are commonly used in photography, digital displays, and most applications where detailed color information is essential.

**APPLICATION**

**Photo Enhancement App:**

Develop an application that automatically enhances the quality of photos by adjusting brightness, contrast, sharpness, and color balance.

**Facial Recognition System:**

Create a facial recognition system that can identify and authenticate individuals based on facial features. This can be used for security, access control, or social media tagging.

**Object Detection and Recognition:**

Build an application that detects and recognizes objects in images. This can be applied to assist visually impaired individuals, inventory management, or security surveillance**.**

**Medical Image Analysis:**

Develop a system for analyzing medical images like X-rays or MRIs. This can aid in the early detection of diseases and assist healthcare professionals in diagnosis.

**Document Scanner:**

Create an application that turns a smartphone into a document scanner. It can automatically detect the document edges, correct perspective, and enhance the document for better readability.

**Augmented Reality Filters:**

Design an app that applies augmented reality filters to live camera feeds. This is popular in social media applications for adding fun and creative effects to photos and videos.

**Image Compression:**

Develop an efficient image compression algorithm to reduce file sizes without significant loss of quality. This can be useful for optimizing storage and bandwidth in various applications.

**Art Style Transfer:**

Create an application that applies artistic styles to photos using neural style transfer algorithms. Users can transform their photos into artworks inspired by famous painters.

**Barcode and QR Code Reader:**

Build an app that can scan and interpret barcodes and QR codes. This is useful for inventory management, product tracking, and information retrieval.

**Gesture Recognition:**

Develop a system that recognizes hand gestures from images or video streams. This can be applied in gaming, sign language interpretation, or human-computer interaction.

**CHAPTER 2**

**2.LITERATURE SURVEY**

**2.1 TITLE :** A Survey on Human Face Expression Recognition Techniques

**AUTHOR :** IM Revina, WRS Emmanuel

**YEAR** 2021

**DESCRIPTION:**

This paper describes the survey of Face Expression Recognition (FER) techniques which include the three major stages such as preprocessing, feature extraction and classification. This survey explains the various types of FER techniques with its major contributions. The performance of various FER techniques is compared based on the number of expressions recognized and complexity of algorithms. Databases like JAFFE, CK, and some other variety of facial expression databases are discussed in this survey. The study on classifiers gather from recent papers reveals a more powerful and reliable understanding of the peculiar characteristics of classifiers for research fellows.

**2**.**2 TITLE :** Deep Residual Neural Network for Child’s Spontaneous Facial Expressions Recognition

**AUTHOR :** A Qayyum, I Razzak

**YEAR** 2021

**DESCRIPTION:**

In this paper, we present progressive light residual learning to classify spontaneous emotion recognition in children. Unlike earlier residual neural network, we reduce the skip connection at the earlier part of the network and increase gradually as the network go deeper. The progressive light residual network can explore more feature space due to limiting the skip connection locally, which makes the network more vulnerable to perturbations which help to deal with overfitting problem for smaller data. Experimental results on benchmark children emotions dataset show that the proposed approach showed a considerable gain in performance compared to the state of art methods.

**2.3 TITLE :** Facial expressions recognition and discrimination in Parkinson’s disease

**AUTHOR :** G Mattavelli, E Barvas, C Longo

**YEAR** 2021

**DESCRIPTION:**

This study aims at assessing emotion recognition and discrimination in PD. Recognition of six facial expressions was studied in order to clarify its relationship with motor, cognitive and neuropsychiatric symptoms. Sensitivity in discriminating happy and fearful faces was investigated to address controversial findings on impairment in early stages of emotion processing. To do so, seventy PD patients were tested with the Ekman 60 Faces test and compared with 46 neurologically unimpaired participants. Patients’ performances were correlated with clinical scales and neuropsychological tests. A subsample of 25 PD patients and 25 control participants were also tested with a backward masking paradigm for sensitivity in happiness and fear discrimination. Results showed that PD patients were impaired in facial emotion recognition, especially for fearful expressions. The performance correlated with perceptual, executive and general cognitive abilities, but facial expression recognition deficits were present even in cognitively unimpaired patients. In contrast, patients’ sensitivity in backward masking tasks was not reduced as compared to controls.

**2.4 TITLE :** Facial Expressions as a Vulnerability in Face Recognition

**AUTHOR :** A Peña, A Morales, I Serna, J Fierrez

**YEAR** 2021

**DESCRIPTION:**

This work explores facial expression bias as a security vulnerability of face recognition systems. Despite the great performance achieved by state-of-the-art face recognition systems, the algorithms are still sensitive to a large range of covariates. We present a comprehensive analysis of how facial expression bias impacts the performance of face recognition technologies. Our study analyzes: i) facial expression biases in the most popular face recognition databases; and ii) the impact of facial expression in face recognition performances. Our experimental framework includes two face detectors, three face recognition models, and three different databases. Our results demonstrate a huge facial expression bias in the most widely used databases, as well as a related impact of face expression in the performance of state-of-the-art algorithms. This work opens the door to new research lines focused on mitigating the observed vulnerability.

**2.5 TITLE :** The Customized Convolutional Neural Network of Face Emotion Expression Classification

**AUTHOR :** MJ Awan, A Raza, A Yasin

**YEAR** 2021

**DESCRIPTION:**

In this paper, Convolutional Neural network (CNN) are used to understand the seven different human face expressions. The seven classes are fear, angry, disgust, sad, happy, surprise and neutral. The dataset consisted is consisted upon almost 36,000 gray scale images. Our customized proposed CNN model of 4 convolutional and 2 fully connected layer is achieved 64.3 % accuracy of test data.

**2.6 TITLE :** The impact of face masks on emotion recognition performance and perception of threat

**AUTHOR :** M Grahlow, CI Rupp, B Derntl

**YEAR** 2022

**DESCRIPTION:**

In the current study, we investigated whether emotion recognition, assessed by a validated emotion recognition task, is impaired for faces wearing a mask compared to uncovered faces, in a sample of 790 participants between 18 and 89 years (condition mask vs. original). In two more samples of 395 and 388 participants between 18 and 70 years, we assessed emotion recognition performance for faces that are occluded by something other than a mask, i.e., a bubble as well as only showing the upper part of the faces (condition half vs. bubble). Additionally, perception of threat for faces with and without occlusion was assessed. We found impaired emotion recognition for faces wearing a mask compared to faces without mask, for all emotions tested (anger, fear, happiness, sadness, disgust, neutral). Further, we observed that perception of threat was altered for faces wearing a mask. Upon comparison of the different types of occlusion, we found that, for most emotions and especially for disgust, there seems to be an effect that can be ascribed to the face mask specifically, both for emotion recognition performance and perception of threat.

**2.7 TITLE :** A Study on Human Face Recognition Techniques

**AUTHOR :** S Ranjani

**YEAR** 2022

**DESCRIPTION:**

In this facial recognition system uses biometrics to capture the facial features from the photograph or video. It compares the information from the known database to find a match. There are many facial recognition techniques used nowadays. Among the biometrics, facial recognition plays a major role. The facial recognition uses machine learning to discover, match and recognize the person, and it is widely used in a variety of ways. The facial recognition system is used to differentiate among the users, and it produces an accurate result. In this work, facial recognition and its techniques will be briefly described.

**2.8 TITLE :** Face Expression Recognition Using a Combination of Local Binary Patterns and Local Phase Quantization

**AUTHOR :** A Durmuşoğlu, Y Kahraman

**YEAR** 2022

**DESCRIPTION:**

This channel helps us to understand people thoughts. The accuracy of the knowledge that have obtained by this channel benefits us in decision-making stages. Therefore, many studies in this domain are conducted for the accurate face expression recognition (FER) systems. In order to establish a robust and reliable FER system, it is a requirement to have the informative facial image features. In this study, it is used Local Binary Patterns (LBP) and Local Phase Quantization (LPQ) methods to establish a reliable FER approach. The main goal is to determine whether combining these two methods makes a contribution or not. Experiments show that combining these two methods increases the classification success rate from 75% to 88%.

**2.9 TITLE :** Beyond Face Value: Evidence for the Universality of Bodily Expressions of Emotion

**AUTHOR :** Z Witkower, AK Hill, J Koster, JL Tracy

**YEAR** 2022

**DESCRIPTION:**

In this study have found that the body is reliably used to express and recognize anger, fear, and sadness, by individuals in several industrialized populations. Here, we provide the first evidence that bodily expressions of these three emotions are reliably recognized by members of an isolated small-scale traditional society: the Mayangna of Nicaragua. Specifically, we found that recognition rates for sadness and anger bodily expressions were high, and recognition rates for a fear bodily expression were lower but still significantly greater than chance. Given that the Mayangna are unlikely to have learned these bodily expressions through cross-cultural transmission, their ability to recognize these displays provides strong evidence for the universality of each expression.

**2.10 TITLE :** Light Field-Based Face and Expression Recognition in the Wild Using Capsule Routing

**AUTHOR :** A Sepas-Moghaddam, A Etemad

**YEAR** 2022

**DESCRIPTION:**

In this context, this paper proposes a new deep face and expression recognition solution, called Caps Field, based on a convolutional neural network and an additional capsule network that utilizes dynamic routing to learn hierarchical relations between capsules. Caps Field extracts the spatial features from facial images and learns the angular part-whole relations for a selected set of 2D sub-aperture images rendered from each LF image. To analyze the performance of the proposed solution in the wild, the first in the wild LF face dataset, along with a new complementary constrained face dataset captured from the same subjects recorded earlier have been captured and are made available. A subset of the in the wild dataset contains facial images with different expressions, annotated for usage in the context of face expression recognition tests. An extensive performance assessment study using the new datasets has been conducted for the proposed and relevant prior solutions, showing that the CapsField proposed solution achieves superior performance for both face and expression recognition tasks when compared to the state-of-the-art.

**CHAPTER 3**

**3.SYSTEM ANALYSIS**

* 1. **EXISTING SYSTEM**

In existing system manual system has been implemented, based on the computer vision research, Haar wavelet is used for image feature detection for object recognition. Human is counted manually for attendance systems. Counting the humans results in an inaccurate result and there will be no database proof Machine learning algorithm has been implemented in existing system Different deep learning algorithm, CNN feature extraction algorithm has been implemented. Based on the computer vision research, Haar wavelet is used for image feature detection for object recognition. The success of the real-time face recognition systems are limited by the varying quality of images due to unreliable environment conditions. Human is counted manually for attendance systems. Counting the humans results in an inaccurate result and there will be no database proof.

**DISADVANTAGE**

* Inaccurate counting result.
* Time consuming process.
* It automatically identifies the expression/emotion of the human correctly.
* It results in high accuracy even though lighting illumination problem occurs.
  1. **PROPOSED SYSTEM**

The proposed system represents a pioneering fusion of facial expression recognition technology with the therapeutic power of music. By integrating machine learning algorithms, particularly deep learning-based techniques like Convolutional Neural Network (CNN), the system is designed to analyze real-time facial expressions and translate them into corresponding musical compositions. This innovative approach enables the system to capture the nuanced emotions displayed on an individual's face and dynamically generate or modify music to synchronize with their emotional state. Leveraging CNNs for facial expression recognition allows for a more robust and nuanced understanding of human emotions, ensuring that the generated musical compositions closely reflect the user's feelings. Furthermore, the personalized nature of this approach holds significant promise for enhancing the effectiveness of musical therapy interventions. By tailoring the musical experience to match the individual's detected emotions, the system aims to create a deeply engaging and therapeutic environment. Whether it's fostering relaxation, facilitating emotional release, or inspiring motivation, the proposed system seeks to harness the power of music as a therapeutic tool in a highly personalized manner. Through the integration of advanced machine learning techniques, this system represents a novel and potentially transformative approach to musical therapy, offering individuals a unique and tailored experience that resonates with their emotional needs.

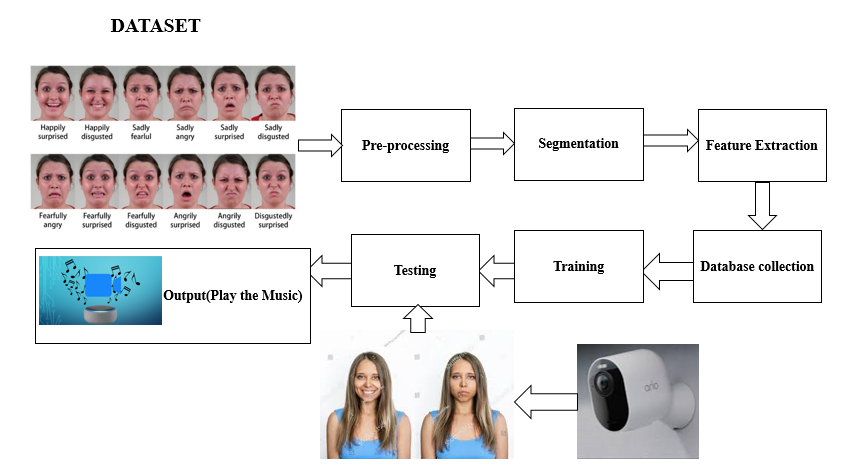
**ADVANTAGE**

* Personalized music recommendations tailored to user emotions.
* Improved user experience with emotionally resonant music suggestions.
* Relevance of recommendations enhanced by considering current emotional state.
* Automated emotion detection from audio inputs via CNN model.
* Scalability and efficiency for processing large volumes of data.
* Insights into user behavior and preferences for informed decision-making.

**CHAPTER – 4**

**4.SYSTEM DESGIN**

**4.1 SYSTEM ARCHITECTURE**

****

**4.2 UML DIAGRAM**

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 object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

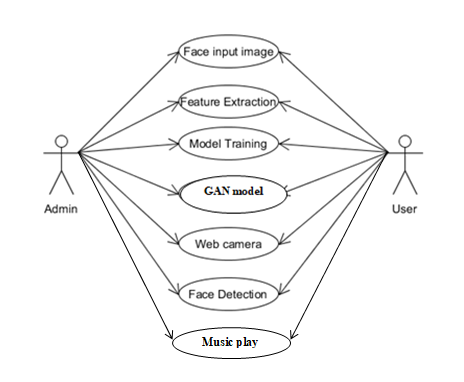
**GOALS:**

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

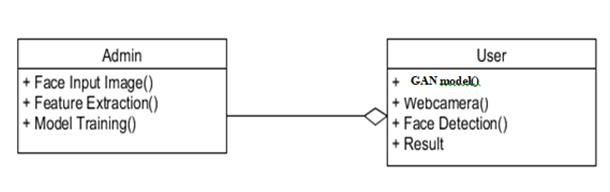
1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a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

**4.3 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.

****

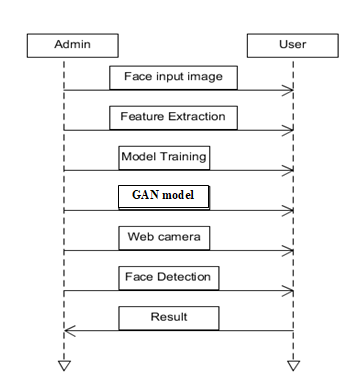
**CNN**

* 1. **CLASS DIAGRAM**

**CNN**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

* 1. **SEQUENCE DIAGRAM**

****

**CNN**

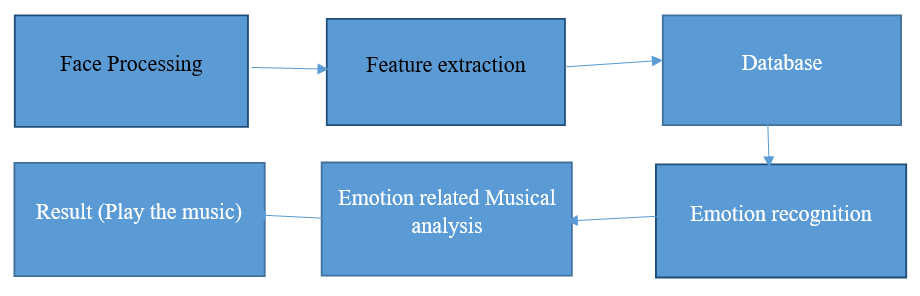
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

**DEPLOYMENT:**

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components.

**4.6 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.



**CHAPTER 5**

**5.SYSTEM IMPLEMENTATION**

**5.1 MODULE LIST**

* Dataset collection
* Preprocessing
* Feature extraction
* Testing with result and analysis

**5.2 MODULE DESCRIPTION**

**DATA COLLECTION:**

This module involves gathering facial expression data from individuals, typically through video are image recordings. The data collected may include various facial expressions such as joy, sadness, anger, and surprise, among others. The quality and diversity of the collected data are crucial for training accurate facial expression recognition models.

**PREPROCESSING:**

The preprocessing module focuses on preparing the collected facial expression data for analysis. This may involve tasks such as face detection and alignment, normalization of facial features, noise reduction, and data augmentation to enhance the robustness and quality of the dataset. Preprocessing ensures that the data is in a suitable format for further analysis and feature extraction.

**FEATURE EXTRACTION:**

Feature extraction involves identifying relevant facial features or descriptors that capture key characteristics of different facial expressions. This module utilizes techniques such as facial landmark detection, texture analysis, and geometric feature extraction to transform raw facial data into meaningful feature vectors. These features serve as input for the facial expression recognition model.

**DATABASE:**

The database module manages the storage and organization of both raw and preprocessed facial expression data. It provides efficient access to the data for training and testing purposes, ensuring that the system can effectively retrieve and manipulate the required information during runtime. Additionally, the database may include metadata associated with each data sample, facilitating easy retrieval and annotation.

**MUSIC GENERATION MODULE:**

The music generation module is responsible for dynamically generating or modifying musical compositions based on the recognized facial expressions. This module utilizes the output of the facial expression recognition model to select or adapt musical elements such as tempo, melody, harmony, and instrumentation. The generated music aims to reflect the emotional state of the user, providing a personalized and engaging musical experience.

**TESTING WITH RESULT AND ANALYSIS:**

The testing module evaluates the performance of the facial expression recognition system and the effectiveness of the generated musical therapy interventions. It involves conducting experiments using test datasets or live user interactions to assess the accuracy of emotion detection and the perceived impact of the generated music on users' emotional well-being. Analysis of the results helps identify strengths, weaknesses, and areas for improvement in the system, guiding future research and development efforts.

**CHAPTER-6**

**6.ALGORITHM**

**Convolutional Neural Network (CNN)**

**6.1 INTRODUCTION**

In this project, we propose an innovative approach to enhance music recommendation systems by incorporating user emotions using deep learning techniques, specifically Convolutional Neural Networks (CNNs). Leveraging the power of CNNs in extracting meaningful features from audio data, we aim to predict user emotions from audio inputs, allowing for more personalized music recommendations aligned with the user's current mood or emotional state. The project entails collecting a sizable dataset of audio samples labeled with corresponding emotional categories and preprocessing them to extract spectrograms or similar representations capturing temporal and frequency information. With a carefully designed CNN architecture, comprising convolutional and max-pooling layers followed by fully connected layers, the model is trained to map input spectrograms to emotional categories. Through rigorous evaluation on a separate validation set, the model's performance in accurately predicting user emotions is assessed using various metrics. Once validated, the trained CNN model is seamlessly integrated into the music recommendation system, where it predicts user emotions based on audio inputs and suggests music genres tailored to the user's emotional context. This novel approach aims to enrich the user experience by offering personalized music recommendations that resonate with the user's emotions, thereby enhancing engagement and satisfaction with the platform.

**6.2 Working process**

In the proposed CNN model for emotion-based music recommendation, the process begins with the collection and preprocessing of a diverse dataset of labeled audio samples, each associated with specific emotional categories. These audio samples undergo preprocessing to extract spectrograms or similar representations capturing essential temporal and frequency information. With a meticulously designed CNN architecture, featuring convolutional layers followed by max-pooling layers, the model is adept at capturing spatial features from the spectrograms. Through an iterative training process, the CNN model learns to map these input spectrograms to corresponding emotional categories, continually adjusting its parameters to minimize the loss function. The model's performance is meticulously evaluated on a separate test set, assessing its ability to accurately predict user emotions from previously unseen audio samples. Once validated, the trained CNN model seamlessly integrates into the music recommendation system, where it dynamically predicts user emotions based on incoming audio inputs. Leveraging these predictions, the recommendation algorithm suggests music genres aligned with the user's emotional context, ultimately enhancing the overall user experience by delivering personalized music recommendations tailored to the user's current emotional state.

**CHAPTER 7**

### **7.SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

**7.1 TYPES OF TESTS**

**7.1.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**7.1.2 Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**7.1.3 Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**7.1.4 System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**7.1.5 White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**7.1.6 Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box.you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**7.2 Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**7.2.1 Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**7.2.2 Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**7.2.3 Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# **7.3 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**7.4 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**CHAPTER 8**

**8.SYSTEM REQUIREMENT**

**H/W SYSTEM CONFIGURATION:-**

* Processor - Pentium – IV
* RAM - 4 GB (min)
* Hard Disk - 20 GB

**S/W SYSTEM CONFIGURATION:-**

* Operating System : Windows 7 or 8
* Software : python Idle

**8.1 PYTHON**

Python is an interpreter, high-level, general-purpose programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it. With Python 2's end-of-life, only Python 3.5.x and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

**8.1.1 Libraries**

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited too many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

* Graphical user interfaces
* +Web frameworks
* Multimedia
* Databases
* Networking
* Test frameworks
* Automation
* Web scraping
* Documentation
* System administration
* Scientific computing
* Text processing
* Image processing
* Machine learning
* Data analytics

**8.1.2Development environments**

Most Python implementations (including CPython) include a read–eval–print loop (REPL), permitting them to function as a command line interpreter for which the user enters statements sequentially and receives results immediately.

Other shells, including IDLE and IPython, add further abilities such as auto-completion, session state retention and syntax highlighting.

As well as standard desktop integrated development environments, there are Web browser-based IDEs; SageMath (intended for developing science and math-related Python programs); PythonAnywhere, a browser-based IDE and hosting environment; and Canopy IDE, a commercial Python IDE emphasizing scientific computing

**8.2 Implementations**

**8.2.1 Reference implementation**

CPython is the reference implementation of Python. It is written in C, meeting the C89 standard with several select C99 features. It compiles Python programs into an intermediate byte code which is then executed by its virtual machine. CPython is distributed with a large standard library written in a mixture of C and native Python. It is available for many platforms, including Windows and most modern Unix-like systems. Platform portability was one of its earliest priorities.

**8.3 Other implementations**

PyPy is a fast, compliant interpreter of Python 2.7 and 3.5. Its just-in-time compiler brings a significant speed improvement over CPython but several libraries written in C cannot be used with it.

Stackless Python is a significant fork of CPython that implements microthreads; it does not use the C memory stack, thus allowing massively concurrent programs. PyPy also has a Stackless version.

MicroPython and CircuitPython are Python 3 variants optimized for microcontrollers. This includes Lego Mindstorms EV3.

RustPython is a Python 3 interpreter written in Rust.

**8.3.1 Unsupported implementations**

Other just-in-time Python compilers have been developed, but are now unsupported:

Google began a project named Unladen Swallow in 2009, with the aim of speeding up the Python interpreter five-fold by using the LLVM, and of improving its multithreading ability to scale to thousands of cores, while ordinary implementations suffer from the global interpreter lock.

Psyco is a just-in-time specializing compiler that integrates with CPython and transforms bytecode to machine code at runtime. The emitted code is specialized for certain data types and is faster than standard Python code.

In 2005, Nokia released a Python interpreter for the Series 60 mobile phones named PyS60. It includes many of the modules from the CPython implementations and some additional modules to integrate with the Symbian operating system. The project has been kept up-to-date to run on all variants of the S60 platform, and several third-party modules are available. The Nokia N900 also supports Python with GTK widget libraries, enabling programs to be written and run on the target device.

**8.3.2 Cross-compilers to other languages**

There are several compilers to high-level [object languages](https://en.wikipedia.org/wiki/Object_language), with either unrestricted Python, a restricted subset of Python, or a language similar to Python as the source language:

* [Jython](https://en.wikipedia.org/wiki/Jython) enables the use of the Java class library from a Python program.
* [IronPython](https://en.wikipedia.org/wiki/IronPython) follows a similar approach in order to run Python programs on the .NET [Common Language Runtime](https://en.wikipedia.org/wiki/Common_Language_Runtime).
* The [RPython](https://en.wikipedia.org/wiki/RPython) language can be compiled to [C](https://en.wikipedia.org/wiki/C_(programming_language)), and is used to build the PyPy interpreter of Python.
* [Pyjs](https://en.wikipedia.org/wiki/Pyjs) compiles Python to [JavaScript](https://en.wikipedia.org/wiki/JavaScript).
* [Cython](https://en.wikipedia.org/wiki/Cython) compiles Python to [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B).
* [Numba](https://en.wikipedia.org/wiki/Numba) uses [LLVM](https://en.wikipedia.org/wiki/LLVM) to compile Python to machine code.
* Pythran compiles Python to [C++](https://en.wikipedia.org/wiki/C%2B%2B).
* Somewhat dated [Pyrex](https://en.wikipedia.org/wiki/Pyrex_(programming_language)) (latest release in 2010) and [Shed Skin](https://en.wikipedia.org/wiki/Shed_Skin) (latest release in 2013) compile to C and C++ respectively.
* Google's Grumpy compiles Python to [Go](https://en.wikipedia.org/wiki/Go_(programming_language)).
* [MyHDL](https://en.wikipedia.org/wiki/MyHDL) compiles Python to [VHDL](https://en.wikipedia.org/wiki/VHDL).

[Nuitka](https://en.wikipedia.org/wiki/Nuitka) compiles Python into C++.

**9.APPENDICES**

**CODING:**

**Face Dectection:**

import flask

from flask import Flask,request,render\_template,redirect,url\_for

import tensorflow as tf

from tensorflow import keras

import numpy as np

import cv2

from pygame import mixer

import time

def detect():

mixer.init()

sound = mixer.Sound('Angry.mp3')

sound1 = mixer.Sound('Disgust.mp3')

sound2 = mixer.Sound('Fear.mp3')

sound3 = mixer.Sound('Happy.mp3')

sound4 = mixer.Sound('Sad.mp3')

sound5 = mixer.Sound('Surprise.mp3')

sound6 = mixer.Sound('Neutral.mp3')

emotion = ['Anger', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral']

msg = ''

model = keras.models.load\_model("model\_trained.h5")

font = cv2.FONT\_HERSHEY\_SIMPLEX

cam = cv2.VideoCapture(0)

face\_cas = cv2.CascadeClassifier('./cascades/haarcascade\_frontalface\_default.xml')

while True:

ret, frame = cam.read()

if ret == True:

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

# gray = cv2.flip(gray,1)

faces = face\_cas.detectMultiScale(gray, 1.3, 5)

for (x, y, w, h) in faces:

face\_component = gray[y:y + h, x:x + w]

fc = cv2.resize(face\_component, (48, 48))

inp = np.reshape(fc, (1, 48, 48, 1)).astype(np.float32)

inp = inp / 255.

msg = ''

ans = ''

a = ''

prediction = model.predict(inp)

index = np.argmax(prediction)

time.sleep(13)

if index == 0:

a = 'anger is '

sound.play()

#time.sleep(4)

elif index == 1:

a = 'disgust is'

sound1.play()

elif index == 2:

a = 'fear is '

sound2.play()

elif index == 3:

a = 'happy is'

sound3.play()

elif index == 4:

a = 'sad is'

sound4.play()

elif index == 5:

a = 'suprise is '

sound5.play()

elif index == 6:

a = 'neutral is'

sound6.play()

em = emotion[index]

time.sleep(4)

print(em)

ans = em

score = np.max(prediction)

print(score \* 100)

msg1 = score \* 100

msg = 'Stress level : ' + str(msg1)

cv2.putText(frame, em + " " + str(score \* 100) + '%', (x, y), font, 1, (0, 255, 0), 2)

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)

cv2.imshow("image", frame)

if cv2.waitKey(1) == 27:

break

else:

print('Error')

cam.release()

cv2.destroyAllWindows()

if \_name\_ == '\_main\_':

detect()

**Facial Expression Code:**

import tensorflow as tf

from tensorflow import keras

#from tensorflow.keras.models import Sequential

#from tensorflow.keras.layers import Dense, Conv2D, MaxPooling2D, Dropout, Flatten

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

#-------------------------------------------------------------------------------------------

def generate\_dataset():

"""generate dataset from csv"""

df = pd.read\_csv("./fer2013/fer2013.csv")

train\_samples = df[df['Usage']=="Training"]

validation\_samples = df[df["Usage"]=="PublicTest"]

test\_samples = df[df["Usage"]=="PrivateTest"]

y\_train = train\_samples.emotion.astype(np.int32).values

y\_valid = validation\_samples.emotion.astype(np.int32).values

y\_test = test\_samples.emotion.astype(np.int32).values

X\_train =np.array([ np.fromstring(image, np.uint8, sep=" ").reshape((48,48)) for image in train\_samples.pixels])

X\_valid =np.array([ np.fromstring(image, np.uint8, sep=" ").reshape((48,48)) for image in validation\_samples.pixels])

X\_test =np.array([ np.fromstring(image, np.uint8, sep=" ").reshape((48,48)) for image in test\_samples.pixels])

return X\_train, y\_train, X\_valid, y\_valid, X\_test, y\_test

#------------------------------------------------------------------------------------------------------------------

def generate\_model(lr=0.001):

"""training model"""

with tf.device('/gpu:0'):

model = keras.models.Sequential()

model.add(keras.layers.Conv2D(64,(3,3), input\_shape=(48,48, 1), padding="same"))

model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Activation('relu'))

model.add(keras.layers.MaxPooling2D())

model.add(keras.layers.Dropout(0.20))

model.add(keras.layers.Conv2D(128,(5,5), padding='same'))

model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Activation('relu'))

model.add(keras.layers.MaxPooling2D())

model.add(keras.layers.Dropout(0.20))

model.add(keras.layers.Conv2D(512,(3,3), padding="same"))

model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Activation('relu'))

model.add(keras.layers.MaxPooling2D())

model.add(keras.layers.Dropout(0.20))

model.add(keras.layers.Conv2D(512,(3,3)))

model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Activation('relu'))

model.add(keras.layers.MaxPooling2D())

model.add(keras.layers.Dropout(0.25))

model.add(keras.layers.Conv2D(256,(3,3), activation='relu'))

model.add(keras.layers.Conv2D(128,(3,3), padding='same', activation='relu'))

model.add(keras.layers.MaxPooling2D())

model.add(keras.layers.Dropout(0.25))

#model.add(keras.layers.GlobalAveragePooling2D())

model.add(keras.layers.Flatten())

model.add(keras.layers.Dense(256))

model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Activation('relu'))

model.add(keras.layers.Dropout(0.5))

model.add(keras.layers.Dense(512, activation='relu'))

model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Activation('relu'))

model.add(keras.layers.Dropout(0.5))

model.add(keras.layers.Dense(7,activation='softmax'))

model.compile(loss="sparse\_categorical\_crossentropy", optimizer=keras.optimizers.Adam(lr=lr) , metrics=['accuracy'])

return model

#-------------------------------------------------------------------------------------------------------------------

if \_name=="main\_":

#df = pd.read\_csv("./fer2013/fer2013.csv")

X\_train, y\_train, X\_valid, y\_valid, X\_test, y\_test = generate\_dataset()

X\_train = X\_train.reshape((-1,48,48,1)).astype(np.float32)

X\_valid = X\_valid.reshape((-1,48,48,1)).astype(np.float32)

X\_test = X\_test.reshape((-1,48,48,1)).astype(np.float32)

X\_train\_std = X\_train/255.

X\_valid\_std = X\_valid/255.

X\_test\_std = X\_test/255.

model = generate\_model(0.01)

with tf.device("/gpu:0"):

history = model.fit(X\_train\_std, y\_train,batch\_size=128,epochs=35, validation\_data=(X\_valid\_std, y\_valid), shuffle=True)

model.save("model\_trained.h5")

**OUTPUT SCREENSHOT:**

Output 1:



Output 2:

****

**10.CONCLUSION**

In conclusion, the integration of Convolutional Neural Network (CNN) into the realm of facial expression recognition for musical therapy holds immense promise for revolutionizing the field. By leveraging the capabilities of CNN algorithms to accurately interpret and respond to real-time facial expressions, this innovative approach enables the creation of deeply personalized and dynamically adaptive musical experiences. Through the seamless synchronization of music with the individual's emotional state, this system has the potential to significantly enhance the therapeutic impact of music, fostering relaxation, emotional release, and motivation in users. Furthermore, the utilization of advanced machine learning techniques, such as CNNs, underscores the importance of innovation in addressing complex human needs. As research and development in this area continue to progress, the fusion of facial expression recognition with musical therapy powered by CNNs offers a promising avenue for enhancing emotional well-being and providing tailored support to individuals in need.

**11.FUTURE WORK:**

Future work for the Spotify recommendation system involves optimizing the CNN model for better performance in emotion prediction and exploring multi-modal fusion techniques to incorporate additional data sources, such as text or physiological signals. Long-term user modeling can be developed to adapt recommendations as users' emotional preferences evolve, while enhanced user interaction mechanisms and cross-cultural adaptation can further refine the system's accuracy and relevance. Incorporating explainability and transparency into the recommendation process will build user trust, and real-world deployment and evaluation will validate the system's effectiveness and user satisfaction, guiding further refinement and improvement efforts.

**12.REFERENCE**

[1] T.-H. Tsai, C.-C. Huang, C.-H. Chang, and M. A. Hussain, ‘‘Design of wireless vision sensor network for smart home,’’ IEEE Access, vol. 8,pp. 60455–60467, 2020, doi: 10.1109/ACCESS.2020.2982438.

[2] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, ‘‘Convolutional neural networks based fire detection in surveillancevideos,’’ IEEE Access, vol. 6, pp. 18174–18183, 2018, doi:10.1109/ACCESS.2018.2812835.

[3] M. T. Bhatti, M. G. Khan, M. Aslam, and M. J. Fiaz, ‘‘Weapon detectionin real-time CCTV videos using deep learning,’’ IEEE Access, vol. 9,pp. 34366–34382, 2021, doi: 10.1109/ACCESS.2021.3059170.

[4] Y. Li, Y. Wang and D. Li, "Privacy-preserving lightweight face recognition", Neurocomputing, 2019.

[5]J. Zheng, R. Ranjan, C. Chen, J. Chen, C. D. Castillo and R. Chellappa, "An Automatic System for Unconstrained Video-Based Face Recognition", IEEE Transactions on Biometrics Behavior and Identity Science, vol. 2, no. 3, pp. 194-209, July 2020.

[6]G. T. S. Draughon, P. Sun and J. P. Lynch, "Implementation of a Computer Vision Framework for Tracking and Visualizing Face Mask Usage in Urban Environments", 2020 IEEE International Smart Cities Conference (ISC2), pp. 1-8, 2020.

[7]A. Oumina, N. El Makhfi and M. Hamdi, "Control The COVID-19 Pandemic: Face Mask Detection Using Transfer Learning", 2020 IEEE 2nd International Conference on Electronics Control Optimization and Computer Science (ICECOCS), pp. 1-5, 2020.

[8]R. Anand, T. Shanthi, M. Nithish and S. Lakshman, "Face recognition and classification using googlenet architecture" in Soft Computing for Problem Solving., Springer, pp. 261-269, 2020.

[9]O. Rudenko, O. Bezsonov and O. Romanyk, "Neural network time series prediction based on multilayer perceptron", Development Management, vol. 17, no. 1, pp. 23-34, 2019.

[10]W. Chen, H. Huang, S. Peng et al., "YOLO-face: a real-time face detector", Vis Comput, vol. 37, pp. 805-813, 2021.