**SONG RECOMMENDATION BY FACIAL RECOGNITION**

**A Project Work**

*Submitted in the partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE, SPECIALISATION IN AIML**

**Submitted by**

**AKSHAT PARETA (20BCS6567)**

**RIBHAV SHARMA (20BCS6558)**

**ROHAN MISHRA (20BCS4456)**

**SIMRANJEET SINGH (20BCS6575)**

**Under the Supervision of**

**Mr. Himanshu Mishra**

**CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,**

**PUNJAB**

**May,2023**



**ABSTRACT**

Nowadays, music platforms provide quick access to large amounts of music. They are working continuously to enhance music organization and search management thereby addressing the matter of choice and simplify exploring new music pieces. Recommendation systems have gained huge popularity and help people to pick appropriate music for all occasions. However, there's still a niche in personalization and emotions driven recommendations. Music features a great influence on humans and is widely used for relaxing, mood regulation, destruction from stress and diseases, to take care of mental and physical work. There is a good range of clinical settings and practices in music therapy for wellbeing support. This paper will show the planning of the personalized music recommendation system, driven by listener feelings, emotions and activity contexts. With a mixture of AI technologies and generalized music therapy approaches, a recommendation system is targeted to assist people with music selection for various life situations and maintain their mental and physical conditions.

# **INTRODUCTION**

#### **Problem Definition**

A lot of research has been through with reference to music-driven influence on the physiological and spirit of a person's . Humans perceive a variety of feelings from different types of music and from ancient times considered music influence in the formation of a personal character and ability to treat diseases. Music listening has a significant impact on human feelings, thoughts and as a result, it influences mental and physical health, and the topic of music wellbeing support is gaining popularity. A lot of measurements and research has been conducted to understand the impact of music on brain activity. Music therapy is taken into account as an efficient enhancement to plain care within the treatment of depression.

Music recommendation are often applied in several areas like support of intellectual and physical work, studying, sports, relaxing, stress and tiredness destruction, music therapy and lots of others

* 1. **Project Overview/Specifications**

In this project, we present the design of the personalized emotion-driven music recommendation system. Main purposes of the recommender are: addressing the choice problem, exploring new music pieces, support mental and physical wellbeing and support in improving working processes. The design involves a mixture of AI techniques and generalized, music recommendation and therapy approaches. This project describes approach of applying emotion-driven personalization while music recommendation process. The following section tells application cases of the recommendation system and the problem domain clarification.

* + 1. **Software Used:**

1. Python
2. Machine Learning Libraries

**LITERATURE REVIEW**

1. **EXISTING SYSTEMS:**

**Face Detection and Facial Expression Recognition System**

Anagha S. Dhavalikar et al proposed Automatic Facial Expression recognition system. In This system there are three phases: 1. Face detection 2. Feature Extraction and 3. Expression recognition. The First Phase Face Detection are done by RGB Color model, lighting compensation for getting facial and morphological operations for retaining required face i.e eyes and mouth of the face. This System is also used as AAM i.e Active Appearance Model Method for facial feature extraction. In this technique the point on the face like eye, eye brows and mouth are located and it creates a data file which gives information about the model points detected and detects the face, an expression is given as input AAM Model changes according to expression.

**Emotional Recognition from Facial Expression Analysis using Bezier Curve Fitting**

Yong-Hwan Lee, Woori Han and Youngseop Kim proposed system dependent on Bezier curve fitting. This system uses two steps for facial expression and emotion. first one is detection and analysis of facial area from input original image and next phase is verification of facial emotion of characteristics feature in the region of interest. The first phase for face detection uses color still image based on skin color pixel by initialized spatial filtering, dependent on the result of lighting compassion than to estimate face position and facial location of eyes and mouth. It uses feature maps. After extracting regions of interest this system extract points of the feature map to apply the Bezier curve on the eye and mouth for the understanding of emotion. This system uses training and measuring the difference of Hausdorff distance with the Bezier curve between entered face image and image from the database.

**Using Animated Mood Pictures in Music Recommendation**

Arto Lehtiniemi and Jukka Holm et al proposed system on animated mood picture in music recommendation. On this system the user interacts with the collection of images to get music recommendation with respect to genre of images. This music recommendation system is made by Nokia researched center. This system utilizes textual meta tags for describing the genre and audio signal processing.

**Human-computer interaction using emotion recognition from facial expression.**

F. Abdat, C. Maaoui et al and A. Pruski et al. They proposed a system fully automatic facial expression and recognition system based on three step face detection, facial characteristics extraction and facial expression classification. This system suggested anthropometric model to detect the face feature point combined to shi and Thomasi method. In this method the variation of 21 distances which describe the facial feature from neutral face and the classification base on SVM (Support Vector Machine).

**An Accurate Algorithm for Generating a Music Playlist based on Facial Expressions**

Anukriti Dureha et al. In this he proposed Manual segregation of a playlist and annotation of songs, in accordance with the present spirit of a user, is labor intensive and time consuming. Numerous algorithms are proposed to automate this process. However, the prevailing algorithms are slow, increase the general cost of the system by using additional hardware (e.g., EEG systems and sensors) and have less accuracy. This paper presents an algorithm that automates the method of generating an audio playlist, supported the facial expressions of a user, for rendering salvage of your time and labor, invested in performing the process manually. The algorithm proposed during this paper aspires to scale back the general computational time and therefore the cost of the designed system. It also focusses at increasing the accuracy of the designed system. The face expression recognition module of the given algorithm is validated by testing the systems against user dependent and user independent dataset.

1. **PROPOSED SYSTEM:**

The human face plays an important role in knowing an individual's mood. Camera is used to get the required input from the human face. One of the applications of this input can be for extracting the information to deduce the mood of an individual. The “emotion” derived from the input provided earlier are used to get a list of songs. This tedious task of manually Segregating or grouping songs into different lists are reduced and helps in generating an appropriate playlist based on an individual's emotional features. Facial Expression Based Music Player aims at scanning and interpreting the data and accordingly creating a playlist based the parameters provided. Thus, our proposed system focuses on detecting human emotions for developing emotion-based music player, which are the approaches used by available music players to detect emotions, which approach our music player follows to detect human emotions and how it is better to use our system for emotion detection.

**Something About Keras :**

Keras is a deep learning API written in Python, running on top of the machine learning platform [TensorFlow](https://github.com/tensorflow/tensorflow). It was developed with a focus on enabling fast experimentation. *Being able to go from idea to result as fast as possible is key to doing good research.*

Keras is:

* **Simple** -- but not simplistic. Keras reduces developer *cognitive load* to free you to focus on the parts of the problem that really matter.
* **Flexible** -- Keras adopts the principle of *progressive disclosure of complexity*: simple workflows should be quick and easy, while arbitrarily advanced workflows should be *possible* via a clear path that builds upon what you've already learned.
* **Powerful** -- Keras provides industry-strength performance and scalability: it is used by organizations and companies including NASA, YouTube, or Waymo.

**More About Tensorflow and Keras:**

Machine learning is a complex discipline. But implementing machine learning models is far less daunting and difficult than it used to be, thanks to machine learning frameworks—such as [Google’s TensorFlow](https://www.tensorflow.org/)—that ease the process of acquiring data, training models, serving predictions, and refining future results.

Created by the Google Brain team, TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.

**About OpenCV :**

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding [18 million](https://sourceforge.net/projects/opencvlibrary/files/stats/timeline?dates=2001-09-20+to+2019-01-30). The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV’s deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, [Android](https://opencv.org/opencv/android/) and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured [CUDA](https://opencv.org/opencv/cuda/)and [OpenCL](https://opencv.org/opencv/opencl/) interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

How does TensorFlow works :

TensorFlow allows developers to create *dataflow graphs*—structures that describe how data moves through a [graph](https://www.infoworld.com/article/3263764/database/what-is-a-graph-database-a-better-way-to-store-connected-data.html), or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or *tensor*.

TensorFlow provides all of this for the programmer by way of the Python language. Python is easy to learn and work with, and provides convenient ways to express how high-level abstractions can be coupled together. Nodes and tensors in TensorFlow are Python objects, and TensorFlow applications are themselves Python applications.

The actual math operations, however, are not performed in Python. The libraries of transformations that are available through TensorFlow are written as high-performance C++ binaries. Python just directs traffic between the pieces, and provides high-level programming abstractions to hook them together.

TensorFlow applications can be run on most any target that’s convenient: a local machine, a cluster in the cloud, iOS and Android devices, CPUs or GPUs. If you use Google’s own cloud, you can run TensorFlow on Google’s custom [TensorFlow Processing Unit](https://www.infoworld.com/article/3197331/artificial-intelligence/googles-new-tpus-are-here-to-accelerate-ai-training.html) (TPU) silicon for further acceleration. The resulting models created by TensorFlow, though, can be deployed on most any device where they will be used to serve predictions.

More About the Data used :

* The data we have used here is from <https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks>.
* The file contains more than 160,000 songs collected from Spotify Web API, and also you can find data grouped by artist, year, or genre in the data section.
* It has features like acousticness, energy, loudness and danceability which make the clustering algorithm work more effectively.

More Libraries :

The SpotiPy :

*Spotipy* is a lightweight Python library for the [Spotify Web API](https://developer.spotify.com/web-api/). With *Spotipy* you get full access to all of the music data provided by the Spotify platform.

Pandas :

Pandas is built on top of two core Python libraries—[matplotlib](https://mode.com/python-tutorial/libraries/matplotlib) for data visualization and [NumPy](https://mode.com/python-tutorial/libraries/numpy) for mathematical operations. Pandas acts as a wrapper over these libraries, allowing you to access many of matplotlib's and NumPy's methods with less code. For instance, pandas' .plot() combines multiple matplotlib methods into a single method, enabling you to plot a chart in a few lines.

Before pandas, most analysts used Python for data munging and preparation, and then switched to a more domain specific language like R for the rest of their workflow. Pandas introduced two new types of [objects for storing data](https://mode.com/python-tutorial/python-basics" \l "basic-python-objects) that make analytical tasks easier and eliminate the need to switch tools: Series, which have a list-like structure, and DataFrames, which have a tabular structure.

MatPlot :

**Matplotlib** is a [plotting](https://en.wikipedia.org/wiki/Plotter) [library](https://en.wikipedia.org/wiki/Library_(computer_science)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language and its numerical mathematics extension [NumPy](https://en.wikipedia.org/wiki/NumPy). It provides an [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [API](https://en.wikipedia.org/wiki/API) for embedding plots into applications using general-purpose [GUI toolkits](https://en.wikipedia.org/wiki/GUI_toolkit) like [Tkinter](https://en.wikipedia.org/wiki/Tkinter), [wxPython](https://en.wikipedia.org/wiki/WxPython), [Qt](https://en.wikipedia.org/wiki/Qt_(software)), or [GTK](https://en.wikipedia.org/wiki/GTK). There is also a [procedural](https://en.wikipedia.org/wiki/Procedural_programming) "pylab" interface based on a [state machine](https://en.wikipedia.org/wiki/State_machine) (like [OpenGL](https://en.wikipedia.org/wiki/OpenGL)), designed to closely resemble that of [MATLAB](https://en.wikipedia.org/wiki/MATLAB), though its use is discouraged. [SciPy](https://en.wikipedia.org/wiki/SciPy) makes use of Matplotlib.

Matplotlib was originally written by [John D. Hunter](https://en.wikipedia.org/wiki/John_D._Hunter). Since then it has an active development community and is distributed under a [BSD-style license](https://en.wikipedia.org/wiki/BSD_licenses). Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and was further joined by Thomas Caswell. Matplotlib is a [NumFOCUS](https://en.wikipedia.org/w/index.php?title=NumFOCUS&action=edit&redlink=1) fiscally sponsored project.

Cufflinks :

Plotly is an open-source and charting library that provides the facility of interactive plotting. The library is available for several programming languages such as Python, R, MATLAB, Arduino, and REST, among others.

Cufflink is also a python library that connects plotly with pandas so that we can create charts directly on data frames. It basically acts as a plugin.

Plotly charts are dynamic in nature, it allows us to hover over the values, zoom in zoom out of graphs, identifies outliers in the dataset. On the other hand, matplotlib and seaborn charts are static, we can’t zoom in or zoom-out, it does not provide detail of every value on the chart. The most important feature of plotly is that it allows us to build dynamic charts for the web directly from python which is not possible in the case of matplotlib. With plotly, we can also create animations and interactive graphs on geographical, scientific, statistical, and financial data.

Seaborn :

Seaborn is a library for making statistical graphics in Python. It builds on top of [matplotlib](https://matplotlib.org/) and integrates closely with [pandas](https://pandas.pydata.org/) data structures.

Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

Data Visualization is the graphic representation of data. It converts a huge dataset into small graphs, thus aiding in data analysis and predictions. It is an indispensable element of data science that makes complex data more understandable and accessible. Matplotlib and Seaborn act as the backbone of data visualization through Python.

[Matplotlib:](https://www.geeksforgeeks.org/python-introduction-matplotlib/) It is a Python library used for plotting graphs with the help of other libraries like Numpy and Pandas. It is a powerful tool for visualizing data in Python. It is used for creating statical interferences and plotting 2D graphs of arrays. It was first introduced by John D. Hunter in 2002. It uses Pyplot for providing MATLAB like interface free and open-source. It is capable of dealing with various operating systems and their graphical backends.

[Seaborn](https://www.geeksforgeeks.org/plotting-graph-using-seaborn-python/)**:** It is also a Python library used for plotting graphs with the help of Matplotlib, Pandas, and Numpy. It is built on the roof of Matplotlib and is considered as a superset of the Matplotlib library. It helps in visualizing univariate and bivariate data. It uses beautiful themes for decorating Matplotlib graphics. It acts as an important tool in picturing Linear Regression Models. It serves in making graphs of statical Time-Series data. It eliminates the overlapping of graphs and also aids in their beautification.

Data is an entity that is omnipresent in every aspect of life. Whether it be in spreadsheets, your sales pipeline, social media platforms and more, it can be termed a collection of facts, numbers, words, measurements, observations that have been translated into a form that computers and systems can process. Every piece of information collected or recorded is treated as data that is stored and recorded by systems or humans, individually or combined. Data can be used to measure and understand a wide range of internal and external activities in every ecosystem. With the growth of the world wide web and smartphones becoming technologically superior over the past decade, it has led to a surge in digital data creation. Data churned these days now also include text, audio and video information, and web log and web activity records, which can be categorized as unstructured data. The analysis of such data to gain knowledge of what is happening around and to generate insights that might be useful for decision making is known as Data Analysis. The sole purpose of Data Analysis is to extract useful information from data which can become the pivotal factor for decision making based upon the data analysis. It is being said that “data is the new oil”, the more data one has, the more it will aid the analysis, which might aid in discovering groundbreaking facts.

## **Trinktr GUI library :**

## Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

## Python Pillow Tutorial :

As technology emerges day by day, digital images can be an indispensable source of receiving data. We encounter a lot of digital images in daily life. However, these pictures can be related to anything.

In the programming world, we can process digital images using various libraries or tools. In this article, we will learn about one of the popular tools of Python is **Pillow**. However, [Python](https://www.javatpoint.com/python-tutorial) provides many other useful libraries such as [OpenCV](https://www.javatpoint.com/opencv), Python Image Library (PIL), and **Scikit-image**. This article completely focuses on the **Python Pillow module (PIL)**.

Before dive deep into this topic, let's understand image processing.

Python Pillow module is built on top of PIL (Python Image Library). It is the essential modules for image processing in Python. But it is not supported by Python 3. But, we can use this module with the Python 3.x versions as PIL. It supports the variability of images such as [jpeg](https://www.javatpoint.com/jpg-full-form), [png](https://www.javatpoint.com/png-full-form), bmp, [gif](https://www.javatpoint.com/gif-full-form), ppm, and tiff.

We can do anything on the digital images using the pillow module. In the upcoming section, we will learn various operations on the images such as filtering images, Creating thumbnail, merging images, cropping images, blur an image, resizing an image, creating a water mark and many other operations.

More Projects already existing :

An application which detects our emotions at real time using webcam feed and smartly classifies your playlist into genres, at last playing a song that suits the mood specified by the facial analysis. The application provides a pre-trained model for emotion or mood recognition which has been trained on Kaggle's 'Fer2013' dataset. It also provides a pre-trained model for music classification which has been trained on GTZAN Genre collection.

For running emotion recognition only the file "emotions.py" can be run which takes input from the webcam feed. The source can be changed in the code.

For running the music genre classification the file "audioAnalysis.p can be run with a full command like:- 'python audioAnalysis.py classifyFolder -i --model <model\_name> --classifier --details'

i

Expression Based Music Player :

A web-based music player which detects the user's mood and recommends a set of songs, while changing its background theme accordingly. Created for TOHacks 2021, May 8-9. Made using HTML/CSS/Javascript for the frontend and Django/Python for the backend. The backend integrates both the Haar Cascade machine learning model and a CNN classifier through OpenCV. The classifier was trained using Colaboratory and Tensorflow. Essentially, the Haar Cascade algorithm utilizes the user's webcam to locate the user's face, which is then passed onto the classifier to detect what emotion the person is displaying. The classifier is trained on 4 emotions: Angry, Happy, Calm, and Sad. Depending on what emotion it detects, it will change the song selection and background of the music player accordingly.

Project based on Html and css :

The Flow goes like :: run the capture.py file it will trigger html file which will show you CSS-HTML based music player(webpage) -> Want to play any music just click on play-button shown on song or have plus sign to add it to queue. ->another option of based on emotion will be shown on right uper side, select it. JS will trigger python function. ->camera will start and record your image in backend and go for 10 successful image which contains any face. ->generate emotion prediction on those images and get aggregate result of those 10 result and choose appropriate emotion and forward it to JS script. ->JS chose random song of that genere to play. ->Whenever song will go to end It will repeat the same process in backend such that user can't not know that

# **PROBLEM FORMULATION**

Music influence on human wellbeing can be used across various areas; the use cases presented in the previous section demonstrate the possibility of the practical application of music influence on humans. To address these and other possible cases there is a need of having a generalized system with reflect personal features of each particular user. This project targets design the solution that is aimed to recognize personal aspects of the physical and emotional influence of music-related features in various contexts and combine them with well-known generalized approaches.

# **OBJECTIVES**

The proposed project is aimed to carry out work leading to the development of emotion-based music recommendation system. The proposed aim will be achieved by dividing the work into following objectives:

1. To provide an interface between the music system.
2. To provide a very good entertainment for the users.
3. To implement the ideas of machine learning.
4. To provide a new age platform for music lovers.
5. To bridge gap between growing technologies and music techniques.

# **METHODOLOGY**

The outcome of the project is split into 2 phases:

1. Develop a software to recognize user emotion based on facial expression using Python.

2. Integrate the python code into the web service and play the music based on the facial expression.

* We first process the image of the user taken as an input with the help of a python library for Computer Vision called 'OpenCV'. This captured image is then made available for the CNN in combination with DNN to make a prediction whether the current mood of the user is 'Happy' or 'Sad'.
* The second part is the usage of Unsupervised Machine Learning techniques for clustering songs.The songs are clustered as either of the two classes-'VERY ENTERTAINING'(class 0) and 'RELAXED'(class 1) using the popular K-means algorithm. Then the recommendation is made in order of the current popularity of the respective songs.
* We have an unique story in the way we recommend the songs for each mood, for example when other sites recommend sad songs when a person is sad or feeling bad, we recommend users with songs which will cheer them up('VERY ENTERTAINING') and 'RELAXING' songs when they are 'HAPPY'.
* The code to train the neural network can be found in the 'Emotion\_detector\_version2' iPython notebook. If anyone wants to modify the network to suit their particular needs or feel it is necessary to tweak the network they can do so by making changes to the code present there. The model created is saved as 'final\_model.h5'

The following methodology will be followed to achieve the objectives defined for proposed project:

* 1. Face Recognition: The main objective of face detection technique is to identify the face so as to detect the emotions of the user. To achieve this, the face of the user is detected in the web cam.
  2. Emotion Classification: When the face is successfully detected, then we proceed to classify the emotions of the user
  3. Music Recommendation: The songs are recommended based on the emotion detected. The emotions are assigned to every song. When a certain emotion is detected, the respective song list will be recommended.

1. **RESULTS AND DISCUSSION**

The music player based on facial recognition system is highly essential for all the person in modern day life ecology. This system is further enhanced with benefit able features for upgrading in future. The methodology of enhancement in the playing of songs are done by detection of the facial expression. The facial expression is detected by programming interface with the camera. An alternative method, based on additional emotions which is excluded in our system as disgust and fear. On this emotion included to support the playing of music automatically.

In this project, we presented a model to recommend a music based on the emotion detected from the facial expression. Music is the one that has the power to heal any stress or any kind of emotions. Recent development promises a wide scope in developing emotion-based music recommendation system. Thus, the proposed system presented Face based emotion recognition system to detect the emotions and play music from the emotion detected.

# **REFERENCES**

1. HI James, JJA Arnold, JMM Ruban, M Tamilarasan, Emotion based music recommendation system, academia.edu.
2. S Lekamge, A Marasinghe, Identifying the associations between music and emotion in developing an emotion-based music recommendation system, reasearchgate.net.
3. B Han, S Rho, S Jun, E Hwang, Music emotion classification and context-based music recommendation, psu.edu.
4. MT Quasim, EH Alkhammash, MA Khan, M Hadjouni, Emotion based music recommendation and classification using machine learning with IoT framework, Springer.
5. Y Song, S Dixon, M Pearce, A survey of music recommendation systems and future perspectives, psu.edu.
6. X Zhu, YY Shi, HG Kim, KW Eom, An integrated music recommendation system, academia.edu.
7. Anagha S. Dhavalikar and Dr. R. K. Kulkarni, “Face Detection and Facial Expression Recognition System” 2014 International Conference on Electronics and Communication System (ICECS -2014).
8. Yong-Hwan Lee, Woori Han and Youngseop Kim, “Emotional Recognition from Facial Expression Analysis using Bezier Curve Fitting” 2013 16th International Conference on Network-Based Information Systems.
9. Arto Lehtiniemi and Jukka Holm, “Using Animated Mood Pictures in Music Recommendation”, 2012 16th International Conference on Information Visualisation.
10. F. Abdat, C. Maaoui and A. Pruski, “Human computer interaction using emotion recognition from facial expression”, 2011 UK Sim 5th European Symposium on Computer
11. T.-H. Wang and J.-J.J. Lien, “Facial Expression Recognition System Based on Rigid and Non-Rigid Motion Separation and 3D Pose Estimation,” J. Pattern Recognition, vol. 42, no. 5, pp. 962-977, 2009.
12. Renuka R. Londhe, Dr. Vrushshen P. Pawar, “Analysis of Facial Expression and Recognition Based on Statistical Approach”, International Journal of Soft Computing and Engineering (IJSCE) Volume-2, May 2012.
13. Anukriti Dureha “An Accurate Algorithm for Generating a Music Playlist based on Facial Expressions”: IJCA 2014.
14. Bruce Ferwerda and Markus Schedl “Enhancing Music Recommender Systems with Personality Information and Emotional States”: A Proposal: 2014.