International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I6P129

Volume 72 Issue 6, 315-327, June 2024

Original Article

Facial Emotion-Based Song Recommender System Using CNN

Ashish Tripathi[1], Abhijat Mishra[2], Rajnesh Singh[3], Bhoopendra Dwivedy[4], Amit Kumar[5], Kuldeep Singh[6]

[1,3,4,5]*SCSE, Galgotias University, Greater Noida, Uttar Pradesh, India.*
[2]*Infosys Limited, Pune, Maharashtra, India.*
[6]*CSE, G. L. Bajaj Institute of Technology and Management, Greater Noida, Uttar Pradesh, India.*

[1]*Corresponding Author : ashish.mnnit44@gmail.com*

Received: 02 March 2024 Revised: 20 May 2024 Accepted: 05 June 2024 Published: 29 June 2024

Abstract *- It is observed that many times, people are not able to recognize what kind of song they really want to listen to on the basis of their current mood. Sometimes, people end up searching for the perfect song according to their mood, and it eventually leads to a waste of time in finding the exact requirements of the song. In the era of technology and research, specifically in the world of Artificial Intelligence (AI), implementing these technologies in the advancement of song recommender systems will help people recognize the exact requirements and recommend songs accordingly. It will be the perfect combination of technology and the requirements of the user. This research basically focuses on the recommendation of a song to the person based on his/her current mood. According to the mood of an individual, the song is recommended. The process goes like this: the user first takes a photo of the user with the help of a webcam on the laptop, with the user's permission. After that, the number of photos is matched with the data stored, and when the particular emotion is identified with the help of a CNN (convolutional neural network), it is then redirected to YouTube. According to the mood of the user, the song is played. Hence, the basic requirement of the song recommender system is completed.*

Keywords *- Facial emotion, Convolutional Neural Network (CNN), Song recommendations, Emotion recognition.*

1. Introduction

A facial expression is composed of one or more motions or positions of the facial muscles. One controversial hypothesis contends that these movements make a person's emotional state visible to onlookers. Facial expressions are another kind of nonverbal communication. Most other mammals and a few other animal species use them as a primary form of social communication, in addition to humans. Facial recognition systems often employ Convolutional Neural Networks (CNN) because they excel at image processing, especially in regard to analyzing facial features and expressions. As a form of deep learning algorithm, CNN can automatically learn features from images. CNN is used in facial recognition and can be trained to identify particular facial features, such as the mouth, nose, and eyes, to detect exclusive patterns for distinct individuals. These patterns can then be utilized to recognize and categorize individuals. According to a research paper [7], the facial emotion recognition system utilizes the Convolutional Neural Network (CNN) to normalize images from a dataset. Subsequently, the normalized images are reduced, and with the aid of the CNN, emotions are extracted and presented to the user. Only analysis of the emotion of a person is limited by the use of emotion detection. However, music plays a major role in lightening the mood of the person because emotion detection systems can let an individual identify the external emotions of the person. However, inner emotions can only be treated with the help of music, which makes the user feel more comfortable and stress-free. Songs, and music in general, can play a notable role in depression, with the potential to either worsen symptoms or provide comfort. Listening to music may relieve some individuals experiencing depression, while for others, it may cause negative emotions to surface and worsen their condition. The finding in the research paper [15] suggests that music is accepted as a medicine to deal with depression, and depressed people feel better if they can listen to songs based on mood.

So to better analyze emotion, a song recommendation system becomes important. A facial emotion-based song recommendation system is a deep learning technique that targets the emotions of individuals to identify and recommend the song according to their mood. The sample model of the song recommendation system is shown in Figure 1 for understanding how it works. Currently, the existing recommendation systems do not fully account for the listener's mood, thereby missing out on the deeper emotional connection that could provide more tailored song recommendations.

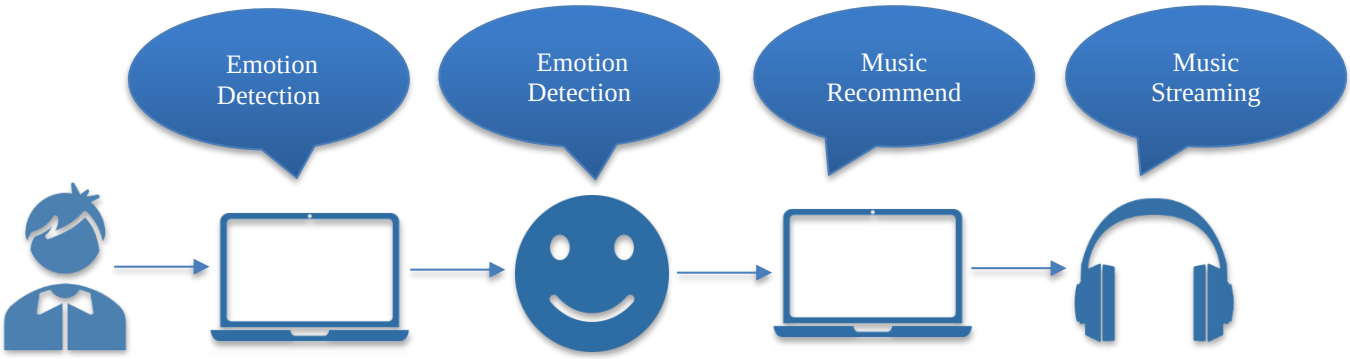

Fig. 1 Song recommendation system

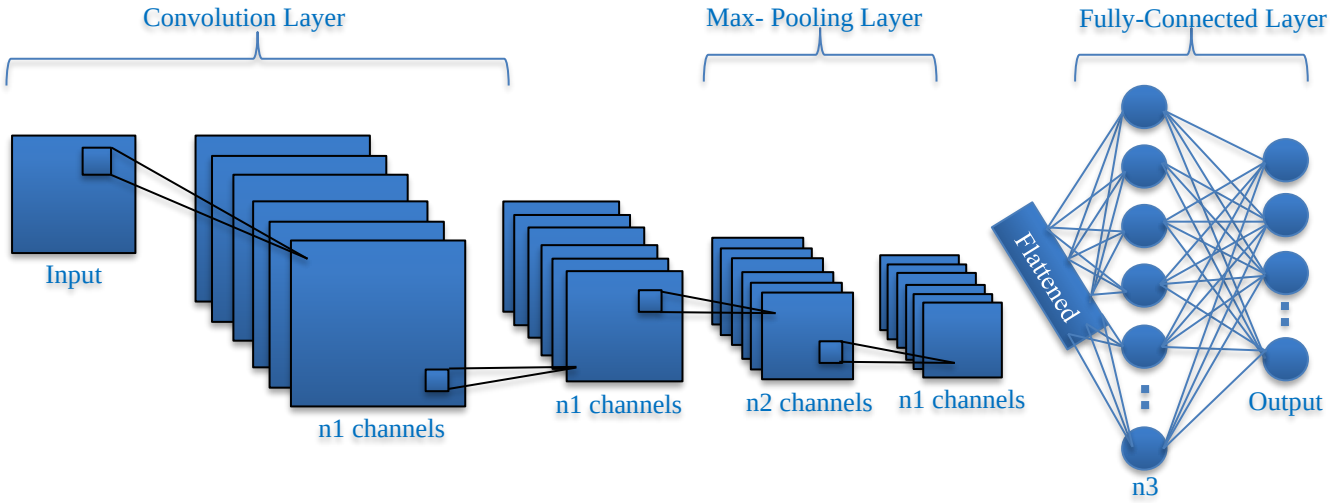

Fig. 2 CNN architecture

Furthermore, the systems cannot categorize the music into the appropriate emotional states based on their inherent properties, instead relying on content-based filtering for suggestions.

To overcome the limitations of the current song recommendation system, the proposed system aims to improve the music listening experience by providing songs that are not only enjoyable but also emotionally connected to a person. It provides responsive and empathic intelligence by incorporating emotional intelligence into music recommendations, resulting in a more personalized and emotionally engaging user experience.

The suggested system contributes to novelty by introducing a real-time, flexible, and responsive song recommender system to the individuals. This system shows optimal accuracy in matching the individual's emotions with the right song.

In this paper, the original dataset is used for training the model. It is a prototype for a brand-new product that has two primary modules: song suggestion and facial expression recognition. By recognizing the user's facial expressions, the system can determine whether the user is feeling happy, sad, excited, or any other emotion. On the basis of testing, evaluating, and experimenting, the overall outcome of the proposed work is accomplished.

The research is primarily focused on developing an effective facial recognition system for song selection that can identify a user's mood. The adopted algorithm would prove to be more effective than the systems now in use. In a particular system [1], a framework for an emotion-based song recommendation system is used that can identify a user's mood based on the data from wearable physiological sensors. Also, this allows for the larger-scale recovery of the time and labour used to complete the procedure manually. The overall idea of the system is to propose songs and recognize facial emotions effectively. The proposed system considers both time and cost efficiency. This system works in the following ways: When the user opens the terminal, a command is initiated that leads to a page where the user can specify their favourite singer and language to listen to the song. Then, a picture is taken using the webcam and matched with the stored training image. After the image is matched, the system

recognizes the emotion and redirects the user to YouTube, where a list of songs that match the recognized emotion is displayed. This system helps the user to find the perfect song that matches their emotions at a particular moment and provides a relaxing and stress-free environment.

2. Background Details

2.1. Convolutional Neural Networks

Shaha et al. have, in their research, stated that CNN is famous for feature and information extraction [16]. CNN is a neural network architecture based on deep learning that has many practical implementations in interpreting images and visualizing data with the help of artificial intelligence. CNN is an upgraded form of artificial neural network that provides more detailed image properties for better classification. In CNN, every image (which is provided as input) is treated as a matrix. Then mathematical operations are performed over the different matrices (input image) to obtain a resultant matrix (output image) from which the required information is extracted.

Figure 2 shows a Convolutional Neural Network (ConvNet) that is composed of five layers, namely the Input Layer, Convolutional Layer, Pooling layer, fully connected layer and output layer.

2.1.1. Input Layer

The input layer takes input as an image from the user converts it into a matrix, and then outputs the converted matrix to the convolutional layer.

2.1.2. Convolutional Layer

The convolutional layer is the layer present after the input layer of CNN. This is the first layer where mathematical operations are performed to extract features of the input images provided. The convolution operation is performed over the input matrix, which includes matrix multiplication of the input image and the filter (which is the MxN matrix often known as the kernel, which is used to extract properties). The result is stored in a feature map, that is developed after doing multiplication of input and kernel. Once the feature map is obtained, there are two important terminologies: padding and stride.

2.1.3. Padding

In order to preserve spatial dimensions throughout the convolution process, padding is the process of adding extra pixels surrounding the input image or feature map. It is essential to the architecture and functionality of convolutional neural networks and aids in preventing information loss at the edges.

2.1.4. Stride

While reading the image, the computer has to decide how far the filter needs to move from one position to the next across the image by stride. The filter moves across the image from the top left corner to the bottom right corner. Here, stride ensures the number of pixels (i.e., squares) that need to be skipped by the filter to read the image.

The size of the stride determines the number of features learned by the filter. The smaller size of the stride tells that more features are learned due to large data extraction. While on the other hand, more size of the stride means less features are learned due to less data extraction.

Size of feature map= (N-M+1)*(N-M+1)
Where,
N= number of rows in the input matrix
M= number of columns in the input matrix
NxN= size of the input matrix
MxM= size of kernel/filter

2.1.5. Pooling Layer

The pooling layer is used to compress the size of the matrix to make mathematical calculations easy, making the cost of computation less, and another is to increase the stability of ConvNet. The pooling layer contributes to a reduction in training parameters, which speeds up computation. Generally, there are three types of pooling operations i.e., max, min, and average pooling.

Max pooling chooses the maximum feature values in the selected region of the image to summarize that region. Min pooling selects the minimal feature values in the selected region of the image to summarize that region. Average pooling determines the summed value of features in a region based on its average value.

2.1.6. Fully Connected Layer

This layer multiplies the input and weight matrix and adds a bias vector to it. This layer performs the function of connecting neurons of the previous and fully connected layer. The formula for calculating a fully connected layer is shown in Equation 1.

$$y_{jk}(x) = f\left(\sum_{i=1}^{n_H} w_{jk}x_i + w_{j0}\right) \quad (1)$$

Here,

W = weight matrix
Wo = bias vector
X= input matrix
Y= output matrix

2.1.7. Output Layer

It is the last layer of the Convolutional Neural Network, whose work is to predict the final answer by mapping the features learned from input images. The output from this layer classifies the emotion of a user, which can be any one of the emotions for which the machine is trained.

3. Literature Review

Table 1 shows the literature review of the works done by the researchers in recent years.

Table 1. Literature review

Author	Data Set	Algorithm/ Model	Work	Result (%)
Zhang et al. [1]	LFW Dataset	CNN	This work presents a CNN-based system for face image recognition and age detection with an average recognition rate.	Recognition Rate: 88.56
Han et al. [2]	Local Dataset	K-Means Clustering	This work presents content analysis using the mel frequency cepstral coefficient (MFCC). It recommends music even when the data is missing.	Accuracy: 80
Sasaki et al. [3]	RWC Dataset	Valence/arousal plane	The valence-arousal plane-based affective music recommendation system suggests music to the user using input visuals and music without needing textual information.	NA
Chang et al. [4]	Local Dataset	Support Vector Machine (SVM)	The suggested work uses a correlation coefficient to identify the qualities of music that elicit a certain emotion. Using this method, one can categorize music based on his/her emotions.	Accuracy: 68.21
Kim et al. [5]	Self-Dataset	K-Means Clustering	This work provides a user-specific personalized music service by analyzing properties and users' preferences for the music.	Music (rock (34) + classical (40)) Recommendation: 74
K.M Aswin et al. [6]	Kohn Canade Dataset	SVM	Using real-time video and audio sources, this system concurrently identifies emotion. After breaking up the video stream into individual images, the system automatically recognizes faces from frames and determines the emotion associated with each one. Average accuracy on Cohn Kanade Dataset: 84.68 Average accuracy on Berlin Emotional Dataset: 80.68	Average accuracy in Real Time: 81.58
Jaiswal et al. [7]	JAFEE+ FERC 2013	CNN	The suggested model detects emotion through facial expressions. Two datasets, i.e., FERC-2013 (Accuracy: 70.14%) and JAFFE (Accuracy: 98.65%), have been used to check the model's performance.	Accuracy: Average of (70.14, 98.65): 84.39
Ayata et al. [8]	Multimodal Deep Emotion Dataset	GSR and PPG	Provided an emotion-based framework for music recommendations that uses inputs from wearable physiological sensors to determine a user's emotion.	Accuracy: GSR: 71.53, 71.04 PPG: 70.92, 70.76
Gilda et al. [9]	Kaggle Face Expression Recognition Challenge	CNN	This work introduces EMP, an effective cross-platform music player that makes music recommendations based on the user's current mood. EMP incorporates emotion context reasoning capabilities into an adaptive music recommendation system to deliver intelligent mood-based music recommendations.	Accuracy: 90.23

Krupa et al. [10]	FERC 2013	Two level CNN	The system at the center of this work proposes songs to users based on their emotional state. Through chatbot conversations and facial expressions, machine vision components are used in this system to assess the user's emotions and select music accordingly.	Accuracy: 88
Chiang et al. [11]	Local Dataset	KBCS, NWFE and SVM	To identify music emotions, this study suggests a method for doing so. In order to identify four different types of music emotions—happy, tense, sad, and peaceful—a total of 35 features related to dynamic, rhythm, pitch, and timbre are created from audio recordings of each music sample.	Accuracy: 86.94 and 92.33
Fessahye et al. [12]	Spotify Recys Challenge	T-RECSYS	This work proposes an enhanced music recommendation system; however, the suggested method can be used for a wide range of platforms and industries, such as videos on YouTube, movies on Netflix, shopping on Amazon, etc.	Precision: 88
Moswedi et al. [13]	Local Dataset	Fourier transform magnitude spectrum.	The authors provided a gene-based classification system for music. They made use of a portion of the music signal's properties. It has been discovered that characteristics like volume, pitch, and pace can be useful in identifying the genre of a piece of music. In this work, by data exploration and analysis, the features that might be utilized for classification are identified. Then, an information ranking classifier is applied to determine the best features.	Accuracy: 85.87
R.L. Rosa et al. [14]	Brazalian Music database	Enhanced Sentiment Metric(eSM)	This study provides an enhanced Sentiment Metric (eSM)-based music recommendation system that uses a sentiment intensity metric. It does this by combining a user's profile-based correction factor with a lexicon-based sentiment metric.	User Satisfaction: 91
Zhu et al. [15]	Self Collected from various sources	AdaBoost Algorithm	This work presents a new tempo characteristic called log-scale modulation frequency coefficients. When paired with timbre features, the suggested tempo feature enhances the ability to classify emotions and musical genres.	Accuracy: 90.5

4. Proposed Work

4.1. Dataset

The dataset used for training the model is self-generated with the help of code whose algorithm is given below. The data set shown in Table 2 consists of five emotions, such as Happy, Sad, Anger, Neutral, and Rock, with each emotion trained with 100 real-time web-captured images of the user. As it is trained on real-time data, the accuracy of the application increases, and it does not depend on static data collected from external sources, which eventually leads to more accurate outcomes.

Table 2. Emotions details

Emotions	Total Images Captured
Happy	100
Sad	100
Neutral	100
Angry	100
Rock	100

4.2. Steps for Data Collection

Step 1: Importing Required Libraries.

- The media pipe library is imported to utilize its functionalities for holistic and hand landmark detection.
- The NumPy library is imported to work with arrays and numerical data.
- The cv2 module is imported from OpenCV for video capturing and image processing.

Step 2: Video Capture Initialization

- Initializes the video capture object to capture frames from the default camera.

Step 3: Input for Data Name

- The user is prompted to input a name for the data.

Step 4: Initializing Variables and Objects

- The holistic and hands modules from Mediapipe are imported for holistic and hand landmark detection, respectively.
- An instance of the Holistic class is created and assigned to the variable holis.
- The drawing _utils module is imported for drawing landmarks on the frames.
- Two empty lists, X and lst, are initialized to store the extracted landmark data.
- data_size is initialized to zero to keep track of the number of captured frames.

Step 5: Frame Processing Loop

- Loop continuously captures frames, processes them, and performs landmark detection using the Holistic class.
- An empty list lst is initialized to store the landmark data for each frame.
- The cap.read () function captures a frame from the video capture object cap, and the returned frame is assigned to frame “frm”.
- The frame “frm” is flipped horizontally using cv2.flip () to correct the mirror effect.
- The frame is then converted from BGR to RGB color space using cv2.cvtColor () before being processed by the Holistic class's process () method. The result is stored in the res variable.

Step 6: Facial Landmark Extraction

- If facial landmarks are detected in the current frame (res.face_landmarks is not None), the loop iterates over each facial landmark (res.face_landmarks.landmark).
- The x and y differences between each landmark's coordinates and the coordinates of the second landmark are computed and appended to the lst list.

Step 7: Left Hand Landmark Extraction

- If left-hand landmarks are detected in the current frame (res.left_hand_landmarks is not None), the loop iterates over each left-hand landmark (res.left_hand_landmarks.landmark).
- The x and y differences between each landmark's coordinates and the coordinates of the eighth landmark are computed and appended to the lst list.
- If no left-hand landmarks are detected, the loop iterates 42 times (the number of landmarks) and appends zeros to the lst list.

Step 8: Right Hand Landmark Extraction

- If right-hand landmarks are detected in the current frame (res.right_hand_landmarks is not None), the loop iterates over each right-hand landmark (res.right_hand_landmarks.landmark).
- The x and y differences between each landmark's coordinates and the coordinates of the eighth landmark are computed and appended to the lst list.
- If no right-hand landmarks are detected, the loop iterates 42 times (the number of landmarks) and appends zeros to the lst list.

Step 9: Data Storage and Visualization

- The extracted landmark data lst for each frame is appended to the X list.
- The data_size variable is incremented by 1 to keep track of the number of captured
- frames.
- Landmarks are visualized on the frame using the draw_landmarks () function from
- drawing_utils.
- The current data_size is displayed on the frame using cv2.putText ().
- The processed frame is displayed in a window using cv2.imshow ().
- If the "Esc" key (key code 27) is pressed or the data_size exceeds 99, the loop breaks, and the video capture is released, and windows are closed.

Step 10: Data Saving and Printing Shape

- The extracted landmark data stored in the X list is converted to a NumPy array using np.array(X).
- The array is saved as a binary file with the given name using np.save().
- The shape of the array is printed using np.array(X).shape.

Figure 3 depicts the screenshot of the data collection with the help of the above-discussed algorithm for data collection for the rocking emotion, which is one among the five emotions; the 61 in the left above corner informs that it is the 61st image for the rocking emotion.

Below is the pseudocode for the data collection used in the proposed model. Notations and their descriptions used in the algorithm for data collection are depicted in Table 3.

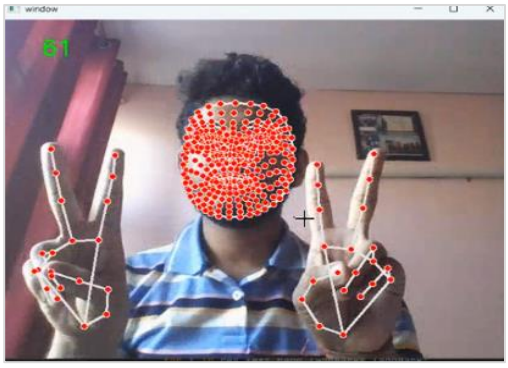

Fig. 3 Data collection

Table 3. Notation in data collection

S. No.	Notation	Description
1	mp	mediapipe
2	np	Numphy
3	cv	cv2
4	lst	list
5	ds	data_size
6	res.fsl	res.face_landmarks.landmark
7	res.lh	res.left_hand_landmarks
8	res.rh	res.right_hand_landmarks
9	frm	frame
10	i	iterator

4.3. Data Collection Algorithm

initialize mp, np and cv
initialize cap for video capture
input name of emotion as name
initialize holistic and hands models
initialize holis as Holistic object
initialize drawing
X ← []
ds ← 0 //initialize data_size as 0
while True:
lst← [] //initialize empty list
read from cap and store in frm
flip the frame to avoid mirror image
process the frm using holis.process() and store the result in res
if res.fsl! ← None:
for i in res.fsl:
append i.x - res.fsl to lst
append i.y - res.fsl to lst
if res.lh!=None:
for i in res.lh:
append i.x - res to lst
append i.y - res.lh to lst
else:
for i in range(42):
append 0.0 to lst
if res.rh!=None:
for i in res.rh:
append i.x - res.rh to lst
append i.y - res.rh to lst
else:
for i in range(42):
append 0.0 to lst
append lst to X
ds← ds+1// data size increment
draw face and hand landmarks on frm using drawing.draw_landmarks ()
put text ds on frm using cv2.putText ()
show the frm using cv2.imshow ()
if key == ESC || ds > 99:
cv2.destroyAllWindows() // exit frame
break
save name.npy
print(np.array(X).shape)

4.4. Simulation Setup

Table 4 shows the specifications of the system and the software used to build the song recommendation system. Hardware requirements include RAM of 4GB with an Intel i5 (64-bit) processor, the operating system used in the development of the application is Windows 10 and codes related to the application are written in PyCharm.

4.5. Methodology

The proposed model consists of three sub-modules, namely, data collection, data training, and the prediction of the input data. The data set for training purposes is collected in real-time, like ChatGpt and it is not collected from any external source. Once the data is collected in real-time then the training is performed. The below-given flowchart (Figure 4) is the overall workflow of the suggested work. The proposed work starts with the command initiated in a terminal. After that, a window appears that leads to a webpage that includes a column for the favourite singer and language in which the user wants to listen to the song. The user performs the required action by entering the name of the singer and song. After performing the required action, it leads to capturing the live emotion of the user using a webcam which appears on the screen of the user. There are two conditions applied in the above situation: if the webcam is able to capture the live emotion and the mood of emotion is directed, it leads to the display of the emotion; otherwise, it reopens the webcam in order to reperform the action of image capture and emotion detection and the comparison of live emotion from the training data set is performed.

Table 4. System specification

Name	Specification
Processor	Intel Core i3 10th Gen
RAM	4 GB
System Type	Python 3.7(64 bit)
Setup	PyCharm
Operating System	Windows 11

After the successful recognition of the image from the training data set, it redirects to YouTube, where the list of songs is displayed; the user can choose any song according to their preference and can relax and release stress by listening to the song. If the user wants to change the song, the webcam can recapture the emotion and will allow a user to choose the perfect song based on the current emotion.

4.6. Proposed Steps

1. Import libraries required
   - streamlit
   - streamlit_webrtc
   - av
   - cv2
   - numpy
   - mediapipe
   - keras
   - webbrowser
   - PIL

2. Load files
   - Load the trained model file "model.h5".
   - Load the emotion labels from "labels.npy".
   - Load the image for the logo.

3. Create an instance of Holistic and Hands from mediapipe.

4. Define a class EmotionProcessor
   - Define a recv function:
   - Convert the video frame to an ndarray format.
   - Flip the frame horizontally.
   - Use Holistic to process the frame and get the facial and hand landmarks.
   - Extract features from the landmarks and predict the emotion using the loaded model.
   - Save the predicted emotion in a file.
   - Draw the facial and hand landmarks on the frame.
   - Return the frame in av.VideoFrame format.

5. Create a streamlit app
   - Display the logo using st.image().
   - Display the header using st.header().
   - Check if the "run" key is not in session_state, and if not, add it.
   - Try to load the previous emotion from "emotion.npy", and if not available, set it to an empty string.
   - Check if the emotion is empty or not, and set the "run" key accordingly.
   - Create two input boxes for language and singer name using st.text_input().
   - Check if the language, singer name, and "run" key are not false, and if so, create a webrtc streamer using webrtc_streamer().
   - Create a button to recommend songs using st.button().
   - If the button is clicked:
   - Check if the emotion is empty or not, and if so, show a warning message and set the "run" key to true.
   - Otherwise, open a web browser with a search query for the given language, emotion, and singer name.
   - Set the emotion to an empty string and the "run" key to false.
   - Save the emotion in "emotion.npy".

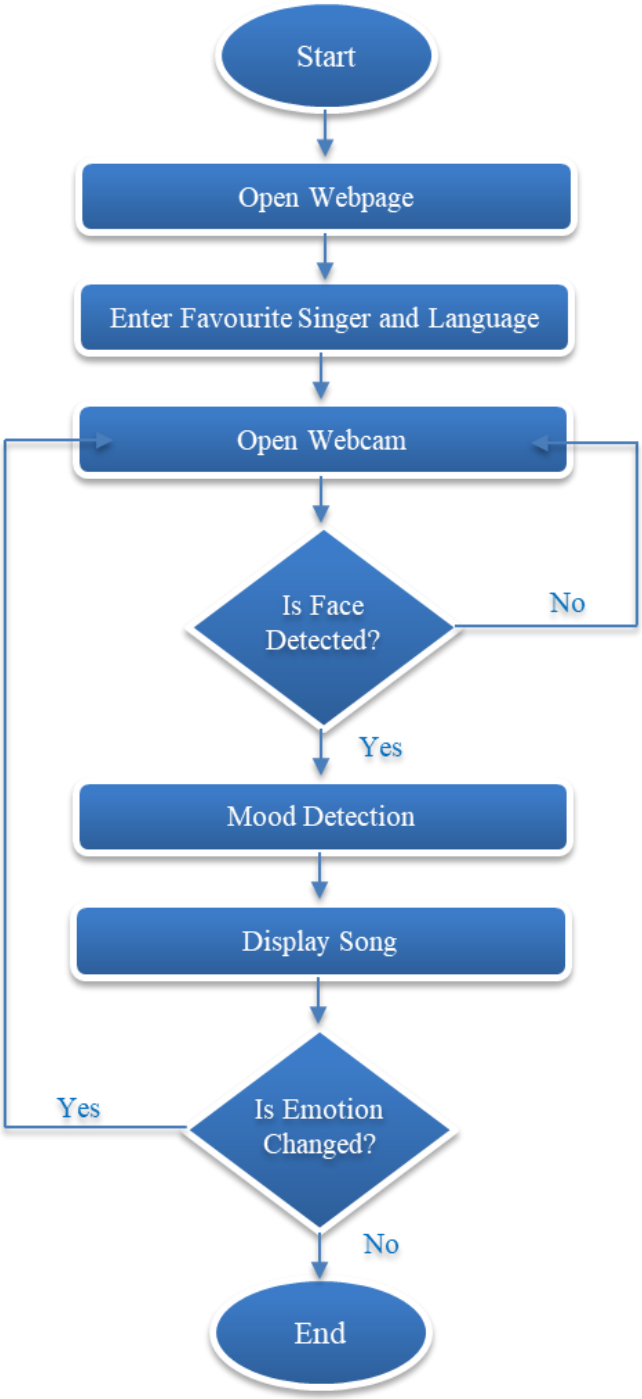

Fig. 4 Detailed working of the proposed model

4.7. Pseudocode of the Proposed Methodology

Below given is the pseudocode of the proposed model. Notations and their descriptions used in the algorithm are depicted in Table 5.

```
import libraries
logo ← Image.open(input image url)
st.image(logo, use_column_width ← True)
model ← load_model("model.h5")
label ← np.load("labels.npy")
holistic ← mp.solutions.holistic
hands ←mp.solutions.hands
holis ← holistic.Holistic()
drawing ← mp.solutions.drawing_utils
if run != st.session_state
   st.session_state[run] ← true
try
   emotion = np.load(emotion.npy)[0]
except
   emotion = ""
if != emotion
   st.session_state[run] ← true
else
   st.session_state[run] ← false
class EmotionProcessor
   def recv(self, frm)
      frm ← frame.to_ndarray(format="bgr24")
      frm ← cv2.flip(frm, 1)
      res←holis.process(cv2.cvtColor(frm,
cv2.COLOR_BGR2RGB))
      lst ← []    // Initialize list
         if res.face_landmarks == 1
         for i ← res.face_landmarks.landmark
            lst.append(i.x - res.face_landmarks.landmark[1].x)
            lst.append(i.y - res.face_landmarks.landmark[1].y)
         if res.left_hand_landmarks==1
            for i <= res.left_hand_landmarks.landmark
               lst.append(i.x -
res.left_hand_landmarks.landmark[8].x)
               lst.append(i.y -
res.left_hand_landmarks.landmark[8].y)
         else
            for i <= range(42)
               lst.append(0.0)
         if res.right_hand_landmarks==1
            for i <= res.right_hand_landmarks.landmark:
               lst.append(i.x -
res.right_hand_landmarks.landmark[8].x)
               lst.append(i.y -
res.right_hand_landmarks.landmark[8].y)
         else
            for i <= range(42)
               lst.append(0.0)
         lst ←  np.array(lst).reshape(1, -1)
         pred ←  label[np.argmax(model.predict(lst))]
         print(pred)
            np.save(emotion.npy, np.array([pred])
      drawing.draw_landmarks(face_marks)
      drawing.draw_landmarks(left_hand_marks)
      drawing.draw_landmarks(right_hand_marks)
      return av.VideoFrame
lang ← st.text_input(Language)
singer ←st.text_input(Singer)
if lang && singer && st.session_state[run] != false
   webrtc_streamer(key←key,
desired_playing_state←True,)
btn ←  st.button()
if btn
   if != emotion
      st.session_state[run] ← true
   else
      Redirect to the streaming platform
      np.save(emotion.npy, np.array([""]))
      st.session_state[run] ←false
```

5. Result Analysis

For the analysis purpose, three individuals tested the suggested system by taking a single photograph representing each of the five emotions throughout the testing phase: happy, neutral, shock, rock and sad. As demonstrated, this research paper discovered that the suggested method reliably recognizes emotions most of the time.

Additional findings of this research paper draw attention to the fact that smiling faces without teeth may be categorised as neutral, a surprised face with smiles may be categorised as happy, and a face without smiles may occasionally be categorised as sad due to the transformation of the mouth.

These findings may help to explain situations in which the accuracy of the teeth's detection is questioned. The testing picture’s real-time nature impacts the accuracy. The accuracy of the proposed model is shown in Table 6. Figure 5 shows a graphical representation of the accuracy of the model for five emotions.

To compare the loss during the training and testing phase of the model, a graph has been plotted, in which the X-axis indicates the Epochs and the Y-axis indicates the loss measure. On comparing after 10 epochs, the loss value is around 0.61 for the training of the model, and the validation loss is about 0.65. On comparing the value of loss after 20 epochs the loss in training and validation has been reduced to 0.45 and 0.49, respectively. After the completion of 30 epochs, the loss in the training of the model reaches 0.40, and for validation, the overall loss reaches 0.42. On completion of 40 epochs, values for loss and validation are 0.36 and 0.43, respectively. At the end of 50 epochs, the overall loss for training and validation of the model is 0.16 and 0.23, respectively. Figure 6 shows the comparison of the training and validation loss of the model. In the figure, the epochs are shown on the X-axis, and the Y-axis represents the loss of the training and testing.

Table 5. Notation for proposed pseudocode

S. No.	Notation	Description
1	lst	list
2	res	chT
3	st	Streamlit library
4	frm	frame
5	av	Audio video library
6	np	Numpy library
7	lang	Language
8	btn	button

Table 6. Accuracy of the model

Person	Mode	Accuracy (%)
Person 1	Happy	95
	Sad	90
	Neutral	80
	Shock	85
	Rock	90
Person 2	Happy	90
	Sad	70
	Neutral	85
	Shock	75
	Rock	95
Person 3	Happy	90
	Sad	80
	Neutral	90
	Shock	85
	Rock	100

To compare the accuracy of the model during the training and testing phase, a graph has been plotted, which is shown in Figure 7, in which the X-axis indicates the Epochs and the Y-axis accuracy measure has been plotted.On comparing after 10 epochs accuracy value is around 0.60 for the training of the model, and for the validation accuracy is about 0.65. On comparing the value of accuracy after 20 epochs the accuracy in training and validation has been updated to 0.65 and 0.72, respectively. After the completion of 30 epochs, the accuracy in the training of the model reaches 0.80, and for validation, the overall accuracy reaches 0.70. On completion of 40 epochs, values of accuracy for training and validation are 0.82 and 0.78, respectively. At the end of 50 epochs, the overall accuracy for training and validation of the model is 0.91 and 0.89, respectively. Below is the graph (Figure 7) to show the comparison of the training and validation of the model.

5.1. Confusion Matrix

The confusion matrices are given for each emotion, in which true positive (TP), true negative (TN), false positive (FP), and false negative (FN) are calculated for five emotions, i.e., Happy, Sad, Angry, Neutral, and Rock. Diagonal elements of the confusion matrix represent the true prediction of a particular class, and the rest of the cells represent wrong predictions for that particular class. Table 7 shows the confusion matrix format. It demonstrates that the predicted images for classes 1 and 0 are represented by vertical columns, while the actual images for classes 1 and 0 are represented by horizontal columns. Also, from Table 8 to Table 12, the prediction percentage for all four quadrants of the confusion matrix is shown on the right side of the confusion matrix for all five emotions.

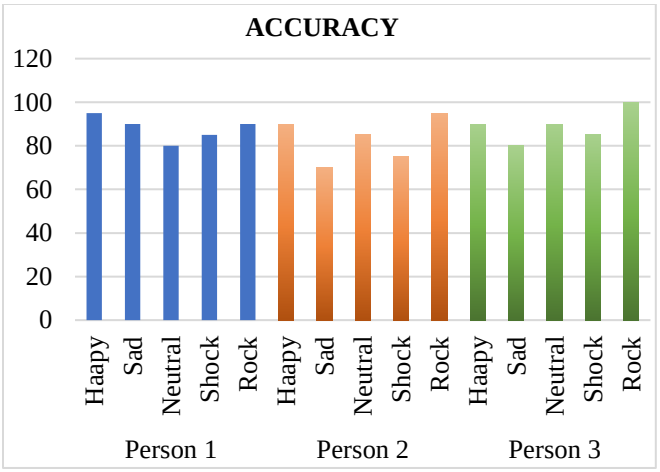

Fig. 5 Accuracy graph

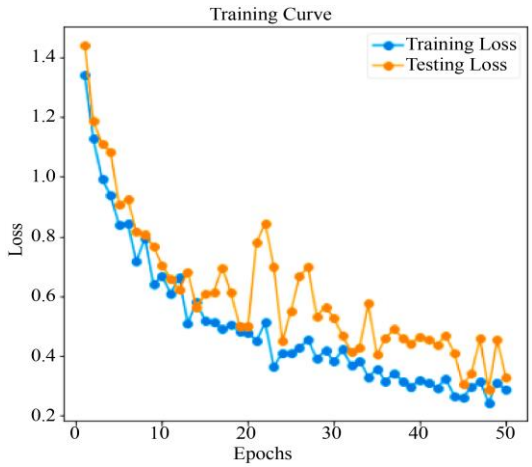

Fig. 6 Model loss

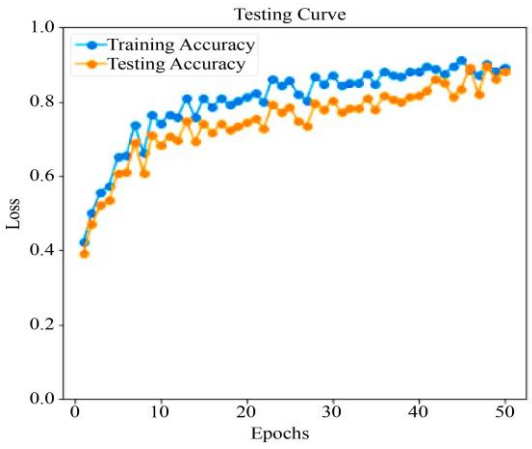

Fig. 7 Model accuracy

Table 7. Confusion matrix format

		Predicted Class	
		Class (1)	Class (0)
Actual Class	Class (1)	TP	FN
	Class (0)	FP	TN

Table 8 shows the confusion matrix for happy emotion, in which 50 images are taken, and matrix predictions present on diagonal elements are classified as correct predictions of the happy class. Out of 50 images, 49 are identified as "happy" and 1 is as "not happy". The 2 provided images are of not happy class but predicted as happy, and 48 images are predicted as not happy, and in actuality, they are also not happy. Table 9 is the confusion matrix for sad emotion, in which 50 images are taken, and matrix predictions present on diagonal elements are classified as correct predictions for the sad class. Out of 50 images, 44 images are truly identified as sad, 4 are classified as not sad, 6 images provided are not sad but predicted as sad, and 46 images are predicted as not sad and, in actuality, they are also not sad.

Table 10 is the confusion matrix for neutral emotion, in which 50 images are taken, and matrix predictions present on diagonal elements are classified as correct predictions of the neutral class. Out of 50 images, 42 images are truly identified as neutral, 6 images are classified as not neutral, 8 images provided are not neutral but predicted as neutral, and 44 images are predicted not neutral; in actuality, they are also not neutral.

Table 8. Confusion matrix for happy class

Predicted Class					
Actual Class		Happy (1)	Not Happy (0)	Prediction (%)	
	Happy (1)	49	2	98	4
	Not Happy (0)	1	48	2	96

Table 9. Confusion matrix for sad class

Predicted Class					
Actual Class		Sad (1)	Not Sad (0)	Prediction (%)	
	Sad (1)	44	4	88	8
	Not Sad (0)	6	46	12	92

Table 10. Confusion matrix for neutral class

Predicted Class					
Actual Class		Neutral	Not Neutral	Prediction (%)	
	Neutral	42	6	84	12
	Not Neutral	8	44	16	88

Table 11. Confusion matrix for rock class

Predicted Class					
Actual Class		Rock (1)	Not Rock (0)	Prediction (%)	
	Rock (1)	48	1	96	2
	Not Rock (0)	2	49	4	98

Table 12. Confusion matrix for angry class

Predicted Class					
Actual Class		Angry (1)	Not Angry (0)	Prediction (%)	
	Angry (1)	46	8	92	16
	Not Angry (0)	4	42	8	84

Table 11 is the confusion matrix for rock emotion, in which 50 images are taken, and matrix predictions present on diagonal elements are classified as correct predictions of rock class. Out of 50 images, 48 images are truly identified as rock, and one image is classified as not rock, two images provided are of not rock class but predicted as rock and 49 images are predicted as not rock, and in actuality, they are also not rock. Table 12 is the confusion matrix for angry emotion, in which 50 images are taken, and matrix predictions present on diagonal elements are classified as correct predictions of the angry class. Out of 50 images, 46 images are truly identified as angry, 8 images are classified as not angry, 4 images provided are of not angry class but predicted as angry, and 42 images are predicted as not angry; in actuality, they are also not angry.

5.2. Classification Report

5.2.1. Precision

It displays the proportion of accurately anticipated situations that really occurred and it is true.

5.2.2. Recall

It shows the proportion of real cases that the model accurately predicts.

5.2.3. F1 – Score

The F1 Score is calculated as the weighted average of Precision and Recall. This score, therefore, takes into account both false positives and false negatives.Table 13 displays the results of the classification analysis for the suggested technique, which includes details on the precision, recall, F1 score, and accuracy for each emotion. The first column lists the names of all five emotions. The precision value for that specific emotion is shown in the second column. The third column displays the recall value for each emotion in its corresponding row. The F1 score value for each emotion is shown in the fourth column. The last column represents the average accuracy for each emotion in the table.

Table 13. Classification report

Emotions	Precision	Recall	F1- Score	Accuracy
Happy	0.96	0.98	0.97	0.97
Sad	0.88	0.92	0.90	0.90
Neutral	0.84	0.88	0.86	0.86
Rock	0.96	0.98	0.97	0.97
Angry	0.92	0.85	0.88	0.88
Average accuracy for all emotions				0.92

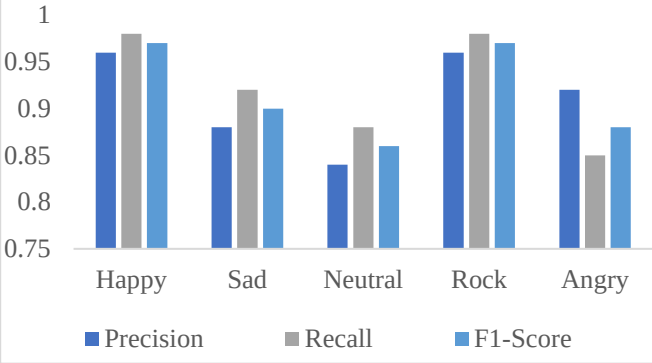

Fig. 8 Classification report

6. Comparative Analysis

Table 14 and Figure 9 compare the proposed model's accuracy with different existing state-of-the-art models. The first model [2] recommends music based on the content analysis. This model shows an accuracy of 80%, which is 12% less efficient than the proposed model. The second model in the table is a CNN-based model that gives 90.23% accuracy in building a computational model to accurately classify emotion into four moods: joyful, sad, angry, and neutral. This model shows 1.77% less accuracy in comparison to the proposed model. In order to identify the characteristics of music that evoke emotion with 68.21% accuracy, the third model is an SVM-based model that uses the correlation coefficient. The proposed model is 23.79% more accurate than this model [4]. Initially, the fourth model [10] in the table got around 63% accuracy on 60 epochs. After that, it was observed that in the most recent epochs, accuracy appeared to rise. After 100 epochs of testing, the accuracy was approximately 88%. When the accuracy of this model is compared, it is found to be 4% less accurate than the proposed one. Based on the categorization of music genre, emotion, and similarity query, this model [15] is put forth as an integrated music recommendation system. For four emotion classifications, they achieved an accuracy of about 90.5%. As compared to this model [15], the proposed model shows a 1.5% improvement in the accuracy. Overall, the proposed model outperformed all the other models in terms of accuracy, demonstrating its effectiveness in emotion classification.

The proposed model shows better results than other state-of-the-art models by addressing the following points (1) novelty in the architectural design that helps to recommend the songs as per the individual's emotions. (2) An improved feature selection and representation process to capture all the required emotions to enhance the model's performance. (3) Data preprocessing includes the data selection process relevant to the emotions (mood) from the dataset and exploratory data analysis for the smooth conduction of training, validation, and testing on the dataset with the allocation of the right proportion of the emotions. (4) Appropriate training and evaluation strategy.Figure 9 is the graph of comparative analysis of the model with the use of different algorithms, including K-Means Clustering, Valence Arousal plane, eSM and CNN.

Table 14. Comparative analysis

Algorithm/Model	Accuracy (%)
K-Means Clustering [2]	80
CNN [9]	90.23
SVM [4]	68.21
Two-level CNN [10]	88
AdaBoost [15]	90.5
Proposed Model (CNN)	92

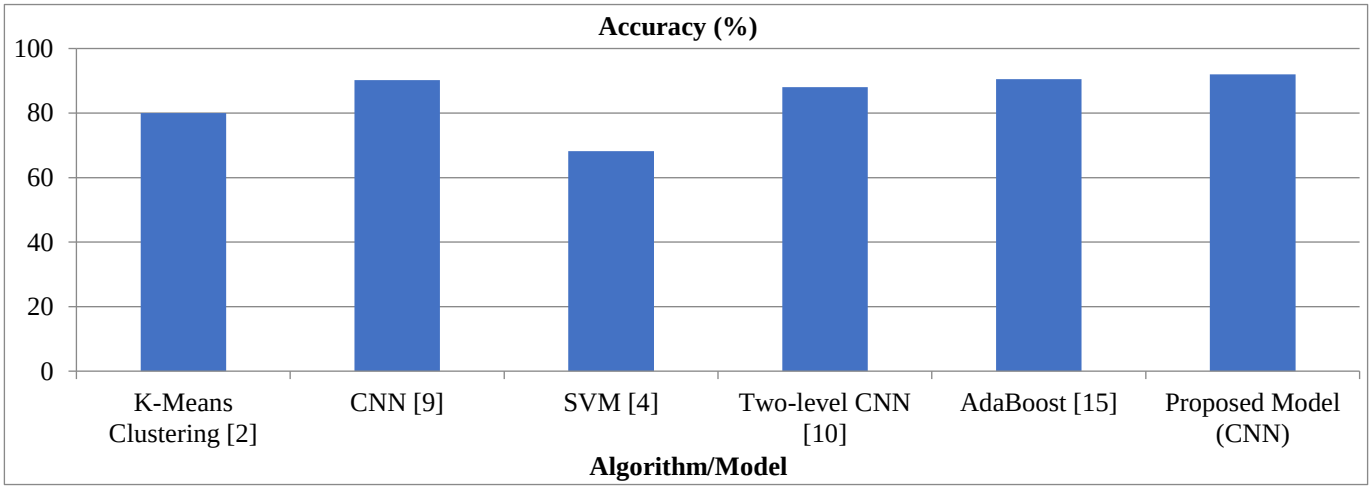

Fig. 9 Accuracy of the models

7. Conclusion

One of the most effective ways to recognize emotions is through facial characteristics. Convolutional Neural Networks (CNN) is a potential approach for emotion recognition. This research paper suggests a music recommendation system based on the user's emotional state and an emotion detection model. In the proposed work, the model is subdivided into two sub-modules: emotion detection and song recommendation. For emotion detection, first data is collected from the data set for the training of the model; after the training validation part of the model is done, and once the model starts working, it is linked to a song recommendation module which takes emotion as input and suggests song based on the mental state of the user. When a user runs the project, it is redirected to a webpage where the user enters information related to the singer and language of the song. After that, the webcam is started, and it captures the emotion of a user, and once the emotion is detected it redirects the user to a song-streaming website on the basis of choice of the user. For future reference, instead of relying on just facial expressions, additional factors like heart rate or body temperature may also be taken into account for reliable identification of the emotions of fear and disgust. Finding appropriate music to play when a sense of dread or disgust is detected is another difficulty. Therefore, it may be viewed as a potential area of focus for this project. Overfitting in trained models can occasionally cause variations in accurate detection. For instance, the anger mood is sometimes mistaken for the disgust emotion since both have comparable visual characteristics (i.e., cheekbones and brows).

References

[1] Hongli Zhang, Alireza Jolfaei, and Mamoun Alazab, "A Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing," *IEEE Access*, vol. 7, pp. 159081-159089, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Huihui Han et al., "Music Recommendation Based on Feature Similarity," *2018 IEEE International Conference of Safety Produce Informatization*, Chongqing, China, pp. 650-654, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Shoto Sasaki et al., "Affective Music Recommendation System Reflecting the Mood of Input Image," *2013 International Conference on Culture and Computing*, Kyoto, Japan, pp. 153-154, 2013. [CrossRef] [Publisher Link]

[4] Chuan-Yu Chang et al., "A Music Recommendation System with Consideration of Personal Emotion," *2010 International Computer Symposium*, Tainan, Taiwan, pp. 18-23, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[5] Dongmoon Kim et al., "A Music Recommendation System with a Dynamic K-Means Clustering Algorithm," *Sixth International Conference on Machine Learning and Applications*, Cincinnati, OH, USA, pp. 399-403, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[6] K.M. Aswin et al., "HERS:Human Emotion Recognition System," *2016 International Conference on Information Science*, Kochi, India, pp. 176-179, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[7] Akriti Jaiswal, A. Krishnama Raju, and Suman Deb, "Facial Emotion Detection Using Deep Learning," *2020 International Conference for Emerging Technology*, Belgaum, India, pp. 1-5, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Deger Ayata, Yusuf Yaslan, and Mustafa E. Kamasak, "Emotion Based Music Recommendation System Using Wearable Physiological Sensors," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 2, pp. 196-203, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[9] Shlok Gilda et al., "Smart Music Player Integrating Facial Emotion Recognition and Music Mood Recommendation," *2017 International Conference on Wireless Communications, Signal Processing and Networking*, Chennai, India, pp. 154-158, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[10] K.S. Krupa et al., "Emotion Aware Smart Music Recommender System Using Two Level CNN," *2020 Third International Conference on Smart Systems and Inventive Technology*, Tirunelveli, India, pp. 1322-1327, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Wei Chun Chiang, Jeen Shing Wang, and Yu Liang Hsu, "A Music Emotion Recognition Algorithm with Hierarchical SVM Based Classifiers," *2014 International Symposium on Computer, Consumer and Control*, Taichung, Taiwan, pp. 1249-1252, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[12] Ferdos Fessahaye et al., "T-RECSYS: A Novel Music Recommendation System Using Deep Learning," *2019 IEEE International Conference on Consumer Electronics*, Las Vegas, NV, USA, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] Davis Moswedi, and Ritesh Ajoodha, "Music Classification Using Fourier Transform and Support Vector Machines," *2022 International Conference on Engineering and Emerging Technologies*, Kuala Lumpur, Malaysia, pp. 1-4, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Renata L. Rosa, Demsteneso Z. Rodriguez, and Graca Bressan, "Music Recommendation System Based on User's Sentiments Extracted from Social Networks," *IEEE Transactions on Consumer Electronics*, vol. 61, no. 3, pp. 359-367, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[15] Xuan Zhu et al., "An Integrated Music Recommendation System," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 3, pp. 917-925, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[16] Manali Shaha, and Meenakshi Pawar, "Transfer Learning for Image Classification," *2018 Second International Conference on Electronics, Communication and Aerospace Technology*, Coimbatore, India, pp. 656-660, 2018. [CrossRef] [Google Scholar] [Publisher Link]