

## TABLE OF CONTENTS

<b>Chapter No.</b>	<b>Chapter Name</b>	<b>Page no.</b>
1.	Introduction	1
2.	Technology	2
3.	Proposed Work	3
4.	Implementation (Code)	4-8
5.	Results/Screenshots	9
6.	Conclusion & Future Scope	10
7.	References	11

## Introduction

MoodTune Pro is like your personal DJ that understands how you feel! Imagine a music player that looks at your face through the computer camera, figures out if you're happy, sad, or anything in between, and then plays music that matches your mood perfectly.

In a world full of music apps, MoodTune Pro is special because it goes beyond just playing songs. It wants to create a musical journey that really connects with your emotions. It captures your facial expressions in real-time, using a camera, and then uses smart technology to pick songs that match how you're feeling.

Think of it as technology that gets you on a deeper level. It's not just about hearing good music; it's about having an experience that feels personal and in tune with your emotions. As we all get used to having technology in our everyday lives, MoodTune Pro shows us a cool way it can make our experiences more special, like having a friend who knows exactly what music you need to hear at the right moment.

Moreover, it's not just about music; it's about making technology more understanding and friendly. MoodTune Pro adds a new chapter to how we interact with our devices, making them not just smart but also emotionally intelligent. It's like having a tech-savvy buddy who can pick up on how you're feeling and make your day a little brighter.

So, in a nutshell, MoodTune Pro is not just a music player; it's a buddy that tunes into your emotions, making your tech experience feel more personal and enjoyable. As we move forward with technology, MoodTune Pro is showing us a sneak peek into a future where our gadgets not only get smarter but also get us on a deeper, more emotional level.

## Technology

MoodTune Pro works like a team of smart helpers using special tools to understand your feelings. Imagine your computer's camera capturing your face in real-time. To make this happen, we use a tool called **OpenCV**. It's like the camera's best friend, helping it see and understand your facial expressions.

Now, to figure out how you're feeling, we have another helper called **DeepFace**. It's like the brain of our team. DeepFace is really good at looking at your face in pictures and figuring out if you're happy, sad, or anything else. It's the one that helps MoodTune Pro accurately detect the main emotion your face is showing.

For the way you interact with MoodTune Pro, we have **Tkinter**. Think of Tkinter as the friendly face that you see on your screen. It creates buttons and labels so you can easily control the music player. It's like the interface that makes everything look nice and helps you click buttons to play, pause, or skip songs.

Now, when it's time to play music, we call on **pygame**. pygame is like the music DJ. It helps play the songs you like, and you can control it with Tkinter. So, OpenCV, DeepFace, Tkinter, and pygame work together like a dream team. OpenCV captures your face, DeepFace understands your feelings, Tkinter gives you buttons to control everything, and pygame plays the music you love. All these helpers team up to make MoodTune Pro a cool and fun experience!

## Proposed Work

Picture MoodTune Pro as your musical best friend, always there to add a soundtrack to your emotions. It's like having a little helper that looks at your expressions through the computer camera and says, "Hey, you look happy today! Let me play some tunes to match that smile." This helper is called DeepFace, and it's super good at understanding if you're excited, relaxed, or feeling something else entirely, just by checking out your face.

Now, imagine opening MoodTune Pro on your computer. What greets you is a friendly screen with buttons – we call this the graphical user interface. It's like a dashboard for your music adventure, where you can press buttons to start the music, take a break, or skip to the next song. And guess what makes MoodTune Pro even more special? It doesn't only play music; it also shares little messages that go hand in hand with your mood. Happy? You might get a cheerful message to brighten your day even more!

But here's the real charm of MoodTune Pro – it transforms listening to music into a journey. It turns your emotions into a melody, creating an interactive experience. You're not just pushing buttons; you're guiding your musical buddy to play the tunes that resonate with your feelings. It's like having a friend who not only understands your emotions but also knows the perfect song for every mood, making your music time a personalized and enjoyable adventure. So, let MoodTune Pro be your sidekick on this musical journey, turning every emotion into a melody that speaks directly to you.

## Implementation (Code)

```
import cv2
from deepface import DeepFace
import tkinter as tk
from tkinter import PhotoImage
from playsound import playsound
import pygame
import os

class EmotionMusicPlayer:
    def __init__(self, root):
        pygame.init()
        pygame.mixer.init()
        self.root = root
        self.root.title("Emotion Music Player")

        self.label = tk.Label(root, text="Capture Image and Play Music Based on Emotion", font=("Helvetica", 16))
        self.label.pack(pady=10)

        self.capture_btn = tk.Button(root, text="Capture Image",
command=self.capture_image)
        self.capture_btn.pack(pady=10)

        self.emotion_label = tk.Label(root, text="", font=("Helvetica", 14))
        self.emotion_label.pack(pady=10)

        self.play_btn = tk.Button(root, text="Play", command=self.play_music)
        self.play_btn.pack(pady=5)
```

```

self.pause_btn = tk.Button(root, text="Pause", command=self.pause_music)
self.pause_btn.pack(pady=5)

self.next_btn = tk.Button(root, text="Next Song", command=self.next_song)
self.next_btn.pack(pady=5)

self.message_label = tk.Label(root, text="", font=("Helvetica", 14))
self.message_label.pack(pady=10)

# Define music mapping for each emotion (update with your actual music files)
self.music_mapping = {
    'happy':
['C:\\Users\\sdgou\\Downloads\\UppongeleGodavari.mp3','C:\\Users\\sdgou\\Downloa
ds\\AppudoIppudo.mp3'],
    'sad': ['path/to/sad_song1.mp3', 'path/to/sad_song2.mp3'],
    'angry': ['path/to/angry_song1.mp3', 'path/to/angry_song2.mp3'],
    'neutral': ['path/to/neutral_song1.mp3', 'path/to/neutral_song2.mp3'],
    'surprise': ['path/to/surprise_song1.mp3', 'path/to/surprise_song2.mp3'],
    'fear': ['path/to/fear_song1.mp3', 'path/to/fear_song2.mp3'],
    'disgust': ['path/to/disgust_song1.mp3', 'path/to/disgust_song2.mp3'],
}

self.current_emotion = None
self.current_song_index = 0
self.paused = False

def detect_emotion(self, img):
    try:
        # Use DeepFace to detect emotion
        result = DeepFace.analyze(img, actions=['emotion'])

```

```

        # Extract the emotion from the result
        dominant_emotion = result[0]['dominant_emotion']

        return dominant_emotion

    except Exception as e:
        print(f"Error during emotion detection: {e}")
        return None

def display_message(self):
    if self.current_emotion:
        messages = {
            'happy': "Great to see you happy! Enjoy the music.",
            'sad': "Cheer up! Music can lift your spirits.",
            'angry': "Take a deep breath. Let the music calm your soul.",
            'neutral': "Feeling neutral? Let's enjoy some tunes!",
            'surprise': "Life is full of surprises! Enjoy the music.",
            'fear': "Don't be afraid. Let the music ease your mind.",
            'disgust': "Feeling disgusted? Music might lighten your mood.",
        }

        message = messages.get(self.current_emotion, "Unknown emotion. Enjoy the music!")

        self.message_label.config(text=message)
        print(f"Displaying message: {message}")

def play_music(self):
    if self.current_emotion:
        if not self.paused:
            self.current_song_index = 0 # Start from the first song
            self.play_next_song()
            self.display_message()

```

```

def pause_music(self):
    if pygame.mixer.music.get_busy():
        pygame.mixer.music.pause()
        print("Music paused.")
    else:
        pygame.mixer.music.unpause()
        print("Music resumed.")

def next_song(self):
    if self.current_emotion:
        self.current_song_index = (self.current_song_index + 1) %
len(self.music_mapping[self.current_emotion])
        self.play_next_song()

def play_next_song(self):
    if self.current_emotion:
        music_files = self.music_mapping[self.current_emotion]
        current_song = music_files[self.current_song_index]

        # Construct the correct path using os.path.join
        music_path = os.path.join('music', self.current_emotion, current_song)

        pygame.mixer.music.load(music_path)
        pygame.mixer.music.play()

        print(f"Playing {self.current_emotion} music: {current_song}")

def capture_image(self):
    # Capture image using OpenCV (requires OpenCV to be installed)
    cap = cv2.VideoCapture(0)

```



```

ret, frame = cap.read()
cap.release()

# Convert OpenCV image to PhotoImage for displaying in Tkinter
img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, (300, 300))
img = PhotoImage(data=cv2.imencode('.png', img)[1].tobytes())

# Display the captured image
image_label = tk.Label(self.root, image=img)
image_label.image = img
image_label.pack(pady=10)

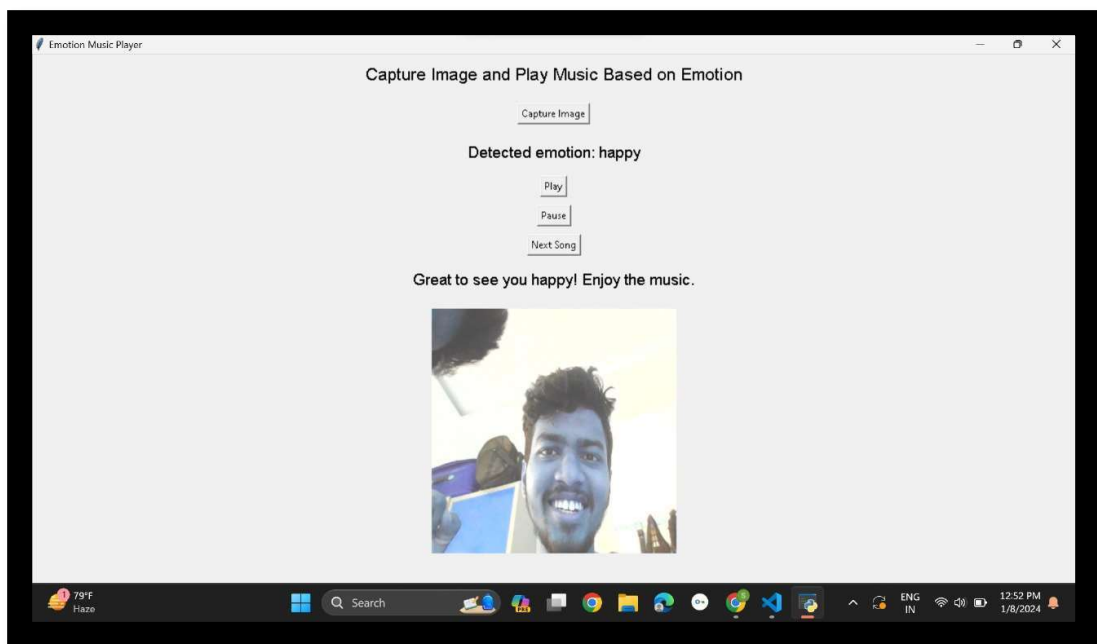
# Detect emotion in the captured image
detected_emotion = self.detect_emotion(frame)

if detected_emotion:
    print(f"Detected emotion: {detected_emotion}")
    self.emotion_label.config(text=f"Detected emotion: {detected_emotion}")
    self.current_emotion = detected_emotion
    self.play_music()
else:
    print("Emotion detection failed.")

if __name__ == "__main__":
    root = tk.Tk()
    app = EmotionMusicPlayer(root)
    root.mainloop()

```

## Results /Screenshot



## **Conclusion & Future Scope**

So, MoodTune Pro has done a great job of using technology to make a special and emotional music experience. It's like having a magic music player that understands how you feel through your face! Now, what's next?

Well, we can make MoodTune Pro even cooler in the future. Imagine having more and more songs to choose from – that would be awesome, right? We could also create a space where you can tell MoodTune Pro what kind of music you really love, making it even more personal. And guess what? We might even connect it to your favorite music apps online, so you can have an endless selection of tunes!

So, the future of MoodTune Pro is bright. We're thinking of adding more songs, making it super personalized for you, and connecting it to the big world of online music. It's like turning your music buddy into a music wizard that knows everything about your taste and feelings. Stay tuned for more exciting updates!

## Reference

- <https://www.geeksforgeeks.org/deep-face-recognition/>
- <https://opencv.org/>
- <https://www.geeksforgeeks.org/python-tkinter-tutorial/>
- <https://www.geeksforgeeks.org/pygame-tutorial/>
- <https://www.geeksforgeeks.org/play-sound-in-python/>
- <https://chat.openai.com/>