

A Fun Virtual Cam Design Project Report

*Design project report submitted in partial fulfilment of the requirements
for the degree of B.Tech. and M.Tech*

by

Sreepathy Jayanand.
(Roll No: CED17I038)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
DESIGN AND MANUFACTURING, KANCHEEPURAM

December 2021

Certificate

I, **Sreepathy Jayanand**, with Roll No: **CED17I038** hereby declare that the material presented in the Design Project Report titled **A Fun Virtual Cam** represents original work carried out by me in the **Department of Computer Science and Engineering** at the **Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram** during the year **2021**. With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: December 11 2021

Student's Signature: Sreepathy Jayanand

In my capacity as supervisor of the above-mentioned work, I certify that the work presented in this Report is carried out under my supervision, and is worthy of consideration for the requirements of Design project during the period Nov 2021 to Dec 2021.

Advisor's Name: Dr. B Sivaselvan

Advisor's Signature:Dr. B Sivaselvan

Abstract

This report documents the work done and the application that has come out of it, a video background changing application based on the emotions detected via his / her speech. The resultant video may be used as a virtual camera, and redirected to popular video conferencing applications to create a more joyful experience.

Acknowledgements

I would like to take this opportunity to express my gratitude to Dr B Sivaselvan, Assistant Professor IIITDM K, for his constant support and guidance.

I would also like to express my sincere thanks to Mrs. Mercy Faustina, for her constant feedback and advice to keep me motivated and for providing a sense of direction.

Contents

Certificate	i
Abstract	ii
Acknowledgements	iii
Contents	iv
List of Figures	vi
List of Tables	vii
Abbreviations	viii
Symbols	ix
1 Introduction	1
1.1 Background	1
1.1.1 Speech recognition	1
1.1.2 Sentiment analysis	2
1.1.3 Multi - threading	2
1.1.4 Computer Vision	2
1.2 Motivation	2
1.3 Objectives of the work	2
2 Methodology	4
2.1 Video processing	4
2.1.1 Video capture	4
2.1.2 Background detection	4
2.1.3 Background removal	5
2.1.4 Frames per second	5
2.2 Audio processing	5
2.2.1 Audio capture	5
2.2.2 Audio to text	5

2.2.3	Sentiment analysis on text	6
2.2.4	Change background given the sentiment	6
3	Work Done	7
3.1	Video Processing	7
3.1.1	Video capture	8
3.1.2	Background detection and removal	8
3.2	Audio processing	8
3.2.1	Audio capture	9
3.2.2	Audio to text	9
3.2.3	Sentiment analysis on text	9
3.2.4	Give sentiment value to Video Processing thread	9
3.3	Code Snippets	10
3.3.1	Imports	10
3.3.2	Audio-Processing Class	10
3.3.3	Video-Processing Class	11
4	Results and Discussions	12
4.1	Outputs	13
5	Extensions possible	15
	Bibliography	16

List of Figures

4.1	Frame without background change	13
4.2	Frames with background change	13
4.3	Frames with and without background	14
4.4	Output - Polarity calculation	14

List of Tables

Abbreviations

CV	Computer Vision
CPU	Central Processing Unit

Symbols

Chapter 1

Introduction

1.1 Background

Covid has hit the world hard, leaving employees and other people no choice but to work from home. Video calls have never been more important. Usually video calls are very monotonous whether it be for the purpose of business or learning. People who are in a bad mood usually stay in the bad mood throughout the meet. The Fun virtual Cam can help alter the monotonicity of the meet, by introducing a variable background for a live video, according to their mood captured via their speech. If they are in a bad mood, we introduce a suitable background for the person which would prompt him to change his speech mannerism, something that would tell him to cheer up a bit more in an indirect way.

Here is a brief description on the major paradigms of computer science used for this particular project.

1.1.1 Speech recognition

Speech recognition is an interdisciplinary subfield of computer science and computational linguistics that develops methodologies and technologies that enable the recognition and translation of spoken language into text by computers[1]

1.1.2 Sentiment analysis

Sentiment analysis is the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information.[\[2\]](#)

1.1.3 Multi - threading

Multithreading is the ability of a central processing unit (CPU) (or a single core in a multi-core processor) to provide multiple threads of execution concurrently, supported by the operating system.[\[3\]](#)

1.1.4 Computer Vision

Computer vision is the field that deals with how computers can help understand high level information given images / videos, so as to capture meaningful information from them.

1.2 Motivation

Video is an essential part of our life. We use it for all sorts of purposes, the main one being video conferencing. During video conferencing using some of the major applications currently in our market, we can do all sorts of tweaks to the video like enabling background bluring, background replacement etc. Usually the background is very subjective, each person has his / her choices, and in case they have to change it they go to the settings of the application to go change it. What if the background automatically changes based on the mood of the person? This is the main thought for creating A Fun Virtual Cam.

1.3 Objectives of the work

The main objectives for the application are:

- Capture the video - From the web camera, we capture frames and display them consecutively as a video.
- Capture the audio - From the microphone, we capture audio.
- Convert audio to text - From the audio captured, we convert it to text.
- Analyse the sentiment from the text - From the text captured from the audio, obtain the sentiments associated in the text.
- Obtain the background images
- Based on the sentiment obtained choose the appropriate background.
- Segment each frame of the video to obtain the foreground and background, and apply the image on the background.

Chapter 2

Methodology

From the logical flow of the application, 2 main components arise.

- Video processing - From the web camera, we capture frames and display them consecutively as a video, along with the corresponding background and the information about the frames per second.
- Audio processing - From the microphone, we capture audio, then convert it into text, analyse the text to recognize the sentiments and give a value to the video processing part to decide on the background

2.1 Video processing

2.1.1 Video capture

The video is obtained by capturing frames from the web camera. Then the frames are displayed one after the other to create the video.

2.1.2 Background detection

The background detection works on the principle of segmentation, where the input image is passed as input ($r \times c \times 3$), and an output ($r \times c \times 1$) segmentation mask is returned

which can be used to detect different segments, i.e foreground and background. The implementation of this feature from the cvzone library is based on the MobileNetV3 model. This is not implemented from scratch within the application but an external library is used for the functionality.

2.1.3 Background removal

Once we get the segmentation mask, i.e once we have information about the background, we can apply another image of the same size on top the base image, only on areas where the detected segment is the background. This creates the illusion that the objects in the image, i.e the foreground is present in another area, i.e the changed background.

2.1.4 Frames per second

The frames per second are calculated based on the number of frame changes happening inside the window per second.

2.2 Audio processing

2.2.1 Audio capture

The audio is captured via the microphone from when the input goes above the minimum threshold defined to when it goes below the threshold.

2.2.2 Audio to text

The audio is converted to text using sequence to sequence models. A third party library is used for it's implementation within the application.

2.2.3 Sentiment analysis on text

Once the text has been obtained from the audio, we analyse the sentiments from the text using BERT and / or other NLP techniques. A third party library is used for it's implementation within the application.

2.2.4 Change background given the sentiment

Once the sentiment is calculated, we can change the background of the video accordingly.

Chapter 3

Work Done

The implementation of the audio processing component takes a relatively high amount of time. If both the video processing component and the audio processing component run the same thread, there would be a drastic drop in frame rates. For the aforementioned reason, the application runs on 2 threads. One for the audio processing and one for the video processing.

3.1 Video Processing

The following packages are used for the video processing.

- cv2
- cvzone
- threading
- time
- os

3.1.1 Video capture

The frames are obtained from capturing the images from the web camera.

```
cam = cv2.VideoCapture(index)
#index -> Index of the camera to be used
img = cam.read()
#img -> The frame captured from the camera
```

Pseudocode for the video

```
while True:
    img = cam.read()
    display(img)
```

3.1.2 Background detection and removal

The background detection and changing it to another image is implemented using the segmentor library of cvzone.

```
imgOutput = segmentor.removeBG(img, ImgWithFlag, threshold)
```

The flag, defines the sentiment values, given by the audio detection thread. Based on the value of flag, an image is chosen for the appropriate sentiment.

The frame rate is calculated using the cvzone library function.

```
fps = cvzone.FPS()
```

3.2 Audio processing

The following packages are used for the audio processing.

- textBlob
- speech-recognition
- threading
- os

3.2.1 Audio capture

The audio is captured via the microphone using the Recognizer from speech-recognition library.

```
receiver = speech_recognition.Recognizer()  
with receiver.Microphone(device_index = 0) as source:  
    audio = receiver.listen(source)
```

3.2.2 Audio to text

The audio is converted to text using the google speech to text recognizer.

```
text = receiver.recognize_google(audio)
```

3.2.3 Sentiment analysis on text

Once the text has been obtained from the audio, we analyse the sentiments with the help of the textBlob library.

```
sentimentObject = TextBlob(text)  
polarityOfText = sentimentObject.sentiment.polarity
```

3.2.4 Give sentiment value to Video Processing thread

Based on the polarity of text, we change the flag value so that the video processing thread can decide which background to keep.

```
if (polarityOfText > 0):  
    flag = 1  
else:  
    flag = 0
```

3.3 Code Snippets

3.3.1 Imports

```
import os
import cv2
import cvzone
from cvzone.SelfiSegmentationModule import SelfiSegmentation
from textblob import TextBlob
import speech_recognition as sr
import threading
import time
from time import perf_counter
```

3.3.2 Audio-Processing Class

```
class Audio_Processing(threading.Thread):
    def __init__(self, name):
        threading.Thread.__init__(self)
        self.name = name
    def run(self):
        global flag
        r1 = sr.Recognizer()
        r1.dynamic_energy_threshold = True
        with sr.Microphone() as source:
            print("Please wait. Calibrating microphone...")
            r1.adjust_for_ambient_noise(source, duration=5)
            r1.energy_threshold = 4000
            while True:
                print("Say something!")
                audio = r1.listen(source, timeout=200)
                try:
                    text = r1.recognize_google(audio)
                    # mic_output = r1.recognize_google(audio)
                    print("I thinks you said '" + text + "'")
                    ff = TextBlob(text)
                    # print(ff.sentiment)
                    if (ff.sentiment.polarity > 0):
                        print("Positive")
                        flag = 2
                    else:
                        print("Negative")
                        flag = 3
                except:
                    pass
```

```
    finally:  
        print("")
```

3.3.3 Video-Processing Class

```
class Image_Processing(threading.Thread):  
    def __init__(self, name):  
        threading.Thread.__init__(self)  
        self.name = name  
    def run(self):  
        global flag  
        cap = cv2.VideoCapture(1)  
        cap.set(3, 640)  
        cap.set(4, 480)  
        segmentor = SelfiSegmentation()  
        fpsReader = cvzone.FPS()  
  
        img_name = "Images/mountains_" + str(flag) + ".jpg"  
        bg1 = cv2.imread(img_name)  
        bg1 = cv2.resize(bg1, (640, 480), interpolation = cv2.INTER_NEAREST)  
        lflag = 2  
        while True:  
  
            if (lflag != flag):  
                img_name = "Images/sentiment" + str(flag) + ".jpg"  
                lflag = flag  
                bg1 = cv2.imread(img_name)  
                bg1 = cv2.resize(bg1, (640, 480), interpolation = cv2.INTER_NEAREST)  
            success, img = cap.read()  
            imgOut = segmentor.removeBG(img, bg1, threshold = 0.8)  
            fps, imgOut = fpsReader.update(imgOut, color = (0, 0, 255))  
            cv2.imshow("Image", imgOut)  
            cv2.waitKey(1)
```

Chapter 4

Results and Discussions

The application - "A Fun Virtual Cam" was made, which can understand the sentiment of the person and change the video background accordingly to create a better experience, while delivering an average frame rate of 43fps on a 1080p web camera.

4.1 Outputs



FIGURE 4.1: Frame without background change



FIGURE 4.2: Frames with background change

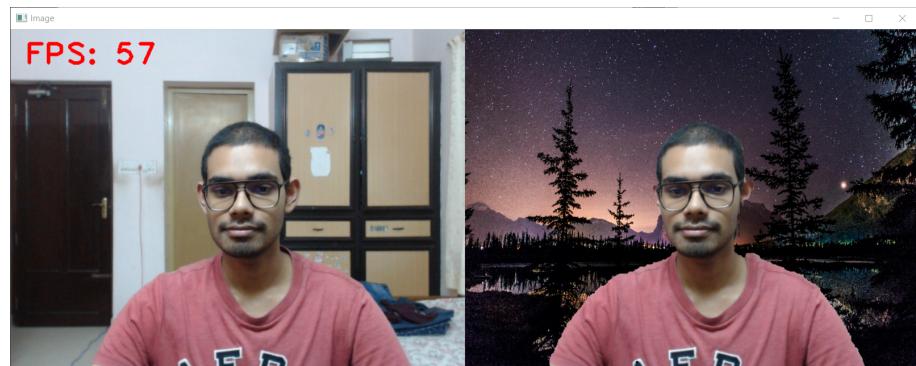


FIGURE 4.3: Frames with and without background

```
(bgrremoval) SJ$python sentiment.py
The movie was bad
Polarity of the text :-0.6999999999999998
The party was excellent.
Polarity of the text :1.0

(bgrremoval) SJ$
```

FIGURE 4.4: Output - Polarity calculation

Chapter 5

Extensions Possible

Currently the application only categorizes the sentiments into positive and negative. We could extend it to cover a bigger use case, for example detecting if a person is angry, sad etc and provide suitable background so that the video

Bibliography

- [1] “What is speech recognition?” [Online]. Available: https://en.wikipedia.org/wiki/Speech_recognition
- [2] “What is Sentiment Analysis?” [Online]. Available: https://en.wikipedia.org/wiki/Sentiment_analysis
- [3] “What is multi - threading?” [Online]. Available: [https://en.wikipedia.org/wiki/Multithreading_\(computer_architecture\)](https://en.wikipedia.org/wiki/Multithreading_(computer_architecture))

A Fun Virtual Cam Design Project Presentation

Sreepathy Jayanand

CED17I038

A Fun Virtual Cam

- It is an application to enable a better user experience for video conferencing and video related applications.
- It changes the background of the video based on the detecting the sentiments from his / her speech.
- It can help change the monotonous nature of conventional video calls.

Working Principle

- There are mainly 2 components for this application:
- Video processing
 - Capture video
 - Get sentiments from audio processing task
 - Change the background of video based on the sentiments
- Audio processing
 - Obtain audio
 - Convert audio to text
 - Get sentiments from the text
 - Give the sentiments to the video processing task

PsuedoCode – Audio Processing Task

```
while True:
```

```
    audio = get_audio(index) // Get audio from mic with given index
```

```
    text = convert_audio_to_text(audio)
```

```
    sentiment = get_sentiment(text)
```

```
    if (sentiment == positive)
```

```
        flag = 1
```

```
    else
```

```
        flag = 0
```

PsuedoCode – Video Processing Task

```
cam = video_capture(index) // index – Index of the web cam to be captured
```

```
while True:
```

```
    img = cap.read()
```

```
    if (flag == 1)
```

```
        bg_image = "positive"
```

```
    else
```

```
        bg_image = "negative"
```

```
    img = change_background(img, bg_image)
```

```
    show(img)
```

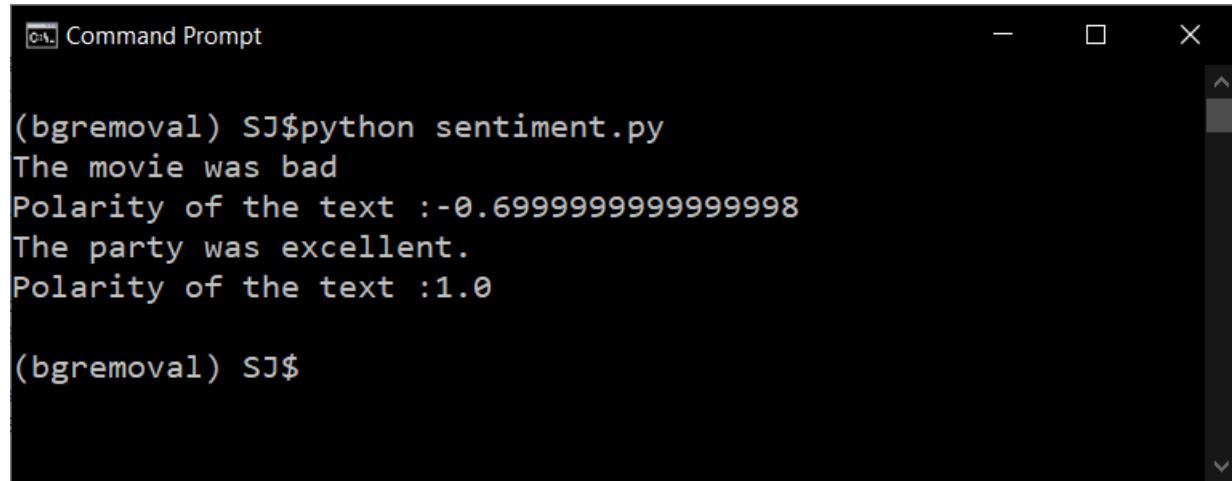
Outputs



Outputs



Outputs



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window contains the following text:

```
(bgremoval) SJ$python sentiment.py
The movie was bad
Polarity of the text :-0.6999999999999998
The party was excellent.
Polarity of the text :1.0

(bgremoval) SJ$
```

Outputs

Demo Video

THANK YOU