****

**Industrial Project Report**

*Submitted in partial fulfillment of the degree of*

**Btech in Electrical Engineering**

**By**

***PRIYA CHAKRABORTY[11901620009]***

***SOURADIP DAS[11901620007]***

***RANJAN GHOSH[11901621021]***

***VIDYASAGAR MALLICK[11901621032]***

**Third-year student of**

**SILIGURI INSTITUTE OF TECHNOLOGY**

**Under the supervision of**

Mr. Ripam Dutta

**Sikharthy Infotech Pvt. Ltd.**

**Department of Electrical Engineering**

Date: 15/09/2022

I hereby forward the documentation prepared under my supervision by **Rupam Kundu Sir** entitled **Siliguri Institute Of Technology** to be accepted as fulfillment of the requirement for the Degree of Bachelor of Technology in Electrical Engineering, **Siliguri Institute Of Technology** affiliated to **Maulana Abul Kalam Azad University of Technology** (**MAKAUT**).

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    **Mr.Ripam Kundu**  **Project Guide**  **Sikharthy Infotech Pvt. Ltd.**  **Director**  **Sikharthy Infotech Pvt. Ltd.** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    **HOD**  **Department Of Electrical Engineering, SIT** |

**TPO,**

**Siliguri Institute of Technology**

**AUTO CAPTURE SELFIE BY SMILE DETECTION WITH PYTHON AND OPENCV**

By

*PRIYA CHAKRABORTY[11901620009]*

*SOURADIP DAS[11901620007]*

*RANJAN GHOSH[11901621021]*

*VIDYASAGAR MALLICK[11901621032]*

UNDER THE GUIDANCE OF

**Mr. Ripam Kundu**

**Project Guide**

**Sikharthy Infotech Pvt. Ltd.**

*THIS IS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF*

**B.Tech**

IN

Electrical Engineering

**SILIGURI INSTITUTE OF TECHNOLOGY**

**AFFILIATED TO**

**Maulana Abul Kalam Azad University of Technology**

**Certificate of Approval**

The foregoing project is hereby approved as a creditable study for the B.Tech in Electrical Engineering presented in a manner of satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorsed or approved any statement made, opinion expressed or conclusion therein but approve this project only for the purpose for which it is submitted.

Final Examination for

Evaluation of the Project ----------------------------------------

-----------------------------------------

------------------------------------------

Signatures of Examiners

**ABSTRACT**

*Emotion detectors are used in many industries, one being the media industry where it is important for the companies to determine the public reaction to their products.*

*Most smartphones these days have a feature that automatically takes a selfie when we smile. It is amazing how accurately it detects smiles for not only one but multiple faces and captures a selfie immediately. If you have wondered how this is possible, it is actually quite simple. Using some of the libraries like OpenCV, it is possible to build a selfie-capturing application with just a few lines of code.*

*So, the main goal of the project is to analyze how it’s working and try to make it in a real-life application by using OpenCV-Python*

**ACKNOWLEDGEMENT**

It is a great pleasure for me to acknowledge the assistance and participation of a large number of individuals in this attempt. Our project report has been structured under the valued suggestion, support, and guidance of **Mr. Ripam Kundu**. Under his guidance, we have accomplished the challenging task in a very short time.

Finally, we express our sincere thankfulness to our family members for inspiring me all throughout and always encouraging us.

Team Members:

***PRIYA CHAKRABORTY[11901620009]***

***SOURADIP DAS[11901620007]***

***RANJAN GHOSH[11901621021]***

***BIDYASAGAR MALLICK[11901621032]***

**TABLE OF CONTENTS**

Chapter 1: Introduction

1: Introduction 8

Chapter 2: What We Used

2.1: Python 8

2.2: OpenCV

Chapter 3: Functionality

3.1: Working Principle 9-10

Chapter 4: Functional Requirement of the System

4.1: Software Requirement 11

Chapter 5: Code and Snapshots

5.1: Codes 11

5.2: Snapshots 12

Chapter 6: Conclusion 13

Chapter 7: References 13

**INTRODUCTION**

*Everyone loves a smiling picture, so we have developed a project which will capture images every time you smile, a* ***Python Project to automatically detect and capture selfies.***

**WHAT WE USED**

1. **Python: -**

**P**ython is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

* web development (server-side),
* software development,
* mathematics,
* system scripting.

**2.OpenCV: -**

OpenCV is a cross-platform library using which we can develop real-time **computer vision applications**. It mainly focuses on image processing, video capture, and analysis including features like face detection and object detection. In this tutorial, we explain how you can use OpenCV in your applications.

**FUNCTIONALITY**

**WORKING PRINCIPLE:**

* *We first import the openCV library.*
* *Now start webcam in the second line using the VideoCapture function of cv2.*
* *Then, include haarcascade files in the python file.*
* *Video is nothing but a series of images so we will run an infinite while loop for the same.*
* *Then we are reading images from the video through read().*
* *As feature recognition is more accurate in gray images we will convert the image to gray image using cvtColor() and BGR2GRAY which are basic openCV functions.*
* *Now we will read faces using an already included haarcascade file and detectMultiscale() function where we pass gray image, ScaleFactor, and minNeighbors.*
* *ScaleFactor: Parameter specifying zoom image, accuracy depends on it so we will keep it close to 1 but not very close as if we take 1.001(very close to 1), then it would detect even shadows so 1.1 is good enough for the face.*
* *minNeighbors: Parameter specifying how many neighbors each rectangle should have to retain it.*
* *If it detects a face we will draw an outer boundary of the face using rectangle() method of cv2 containing 5 arguments: image, initial point (x, y), an endpoint of principal diagonal (x + width, y + height), color of the rectangular periphery and last parameter is the thickness of drawn rectangular periphery.*
* *If the face is detected then we will similarly detect a smile and if a smile is detected too we will print Image<cnt> saved in the cmd/terminal and then we have to provide the location of the folder in which we want to save the images.*
* *To save the images we will use imwrite() which takes 2 parameters- location and image.*
* *To prevent memory overflow we will just save 2 images in one run and thus useif statement which breaks the loop if cnt>=2.*
* *To break infinite loop, we have used an if statement which becomes true when we press ‘q’ denoting ‘quit’.*
* *At last, we will release the video.*
* *Do not forget to destroy all the windows.*

**FUNCTIONAL REQUIREMENTS OF THE SYSTEM**

***SOFTWARE:***

* *Operating System*
* Windows OS 11

***WEB BROWSER:***

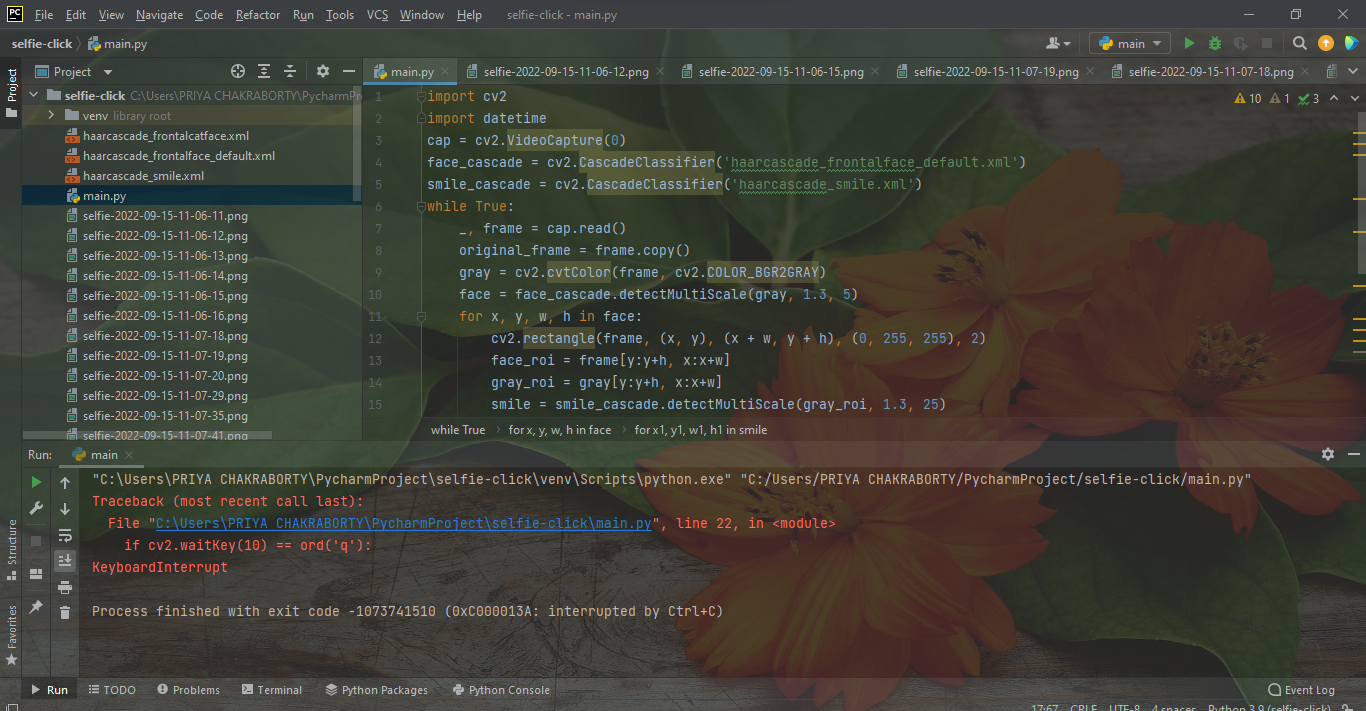
* Internet Explorer 7
* Google Chrome

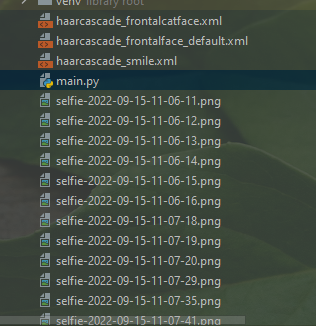
**CODE AND SNAPSHOTS**

Code🡪

import cv2  
import datetime  
cap = cv2.VideoCapture(0)  
face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')  
smile\_cascade = cv2.CascadeClassifier('haarcascade\_smile.xml')  
while True:  
 \_, frame = cap.read()  
 original\_frame = frame.copy()  
 gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)  
 face = face\_cascade.detectMultiScale(gray, 1.3, 5)  
 for x, y, w, h in face:  
 cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 255), 2)  
 face\_roi = frame[y:y+h, x:x+w]  
 gray\_roi = gray[y:y+h, x:x+w]  
 smile = smile\_cascade.detectMultiScale(gray\_roi, 1.3, 25)  
 for x1, y1, w1, h1 in smile:  
 cv2.rectangle(face\_roi, (x1, y1), (x1+w1, y1+h1), (0, 0, 255), 2)  
 time\_stamp = datetime.datetime.now().strftime('%Y-%m-%d-%H-%M-%S')  
 file\_name = f'selfie-{time\_stamp}.png'  
 cv2.imwrite(file\_name, original\_frame)  
 cv2.imshow('cam star', frame)  
 if cv2.waitKey(10) == ord('q'):  
 break

Snapshots-🡪





**Some of our selfies are clicked automatically by webcam by detecting our smile.**

**CONCLUSION**

At first, there were many reasons why the model didn’t work, it would detect non-smiling faces with his smile too. I had to work with the scale factor, as well as the minimum neighbors to get a perfect result.

In this project, we have developed a python project to detect the smile and capture selfies using OpenCV.

And we have used OpenCV because it’s a popular machine learning library in the field of computer vison.

**REFERENCE**

* **https://analyticsindiamag.com/my-fun-project-using-opencv/**
* **Books-OpenCV**