|  |
| --- |
| Final Project |
| Face Detection |
| Real time based with GUI |
|  |
|  |
|  |

Computer Programming

Dr. Asim Wagan

Adil Rao

Ishrat Fatima

Group members:

M. Babar Awan

Arsalan Ahmed

Haziq Ahmed Shaikh

Hassan Ali Gohar

Farooq Arman

|  |
| --- |
| Using Python language and its tools we try to make a program to detect faces. While giving the program a Graphic User Interface and capturing video by OpenCV library. Also marking the detected faces with the help of Dlib |

Real Time Face Detection Using GUI

# Report Abstract:

We would develop a program to detect a face. To do so we first start by importing Open CV library to capture video in real time. Then we code the required algorithms when those algorithms are true the face would have been successfully detected. Then with the help of D lib we position face capturing algorithm and place a marker around the face. Finally we build a GUI with a start and stop button as well as Data Saver button. Describing the start of the whole face detecting process with the push of “Start” button. As well as comparing the value of “End” button with the character “q” as “q” will be pressed to quit the process hence while pressing “End” button we are quitting the program. Finally we advance our program to display details like number of faces detected. To do so we use the help of a counter and displaying that counter in our GUI then we Save this data by pressing “Data Saver” button.

# Introduction:

The purpose of our project is to use face detecting algorithms to detect a face. But making a Graphic User Interface to make it appealing and other libraries to show a marker when face is detected. And using loops to show the number of face detected. While utilizing our GUI to quit the program or execute the program. We are also using OpenCV to connect the camera and capture video.

# Project Management:

**Coding and Compiling:**

* Arsalan Ahmed
* M. Babar Awan

**Dlib:**

* Haziq Ahmed Shaikh

**Graphic User Interface:**

* Hassan Ali Gohar
* Farooq Arman
* **Step I : Coding**

We first import the required libraries like OpenCV. Then we code a simple “Vidcapture()” function as well as passing the parameter “(0)” to capture video from the built-in camera of the laptop. Also importing NumPy to show the frame of the video.

Dlib is imported to run a face detecting Algorithm. By passing the function “dlib.get\_frontal\_face\_detetor()” we call our required face detecting algorithm to run when the program is executed.

Using a “while” loop as a sentinel to infinitely run our program until terminated. In the terminating condition we use “ret.frame” which is used to check if the camera is working or not. Passing the parameter “(GRAY)” to get grayscale live video. Also adding a counter for number of faces detected.

Using a “For” loop to initiate our counter “face\_counter” as well as for the numbers of face detected it prints a rectangle around the face. For the parameters of the rectangle around face we use “face.top() , face.left()” saving them in a variable and “ face.right() , face.bottom()” for the rest.

Coming back to our “while” loop , we use the function “cv2.putText(frame)” the frame parameter along with “(face\_counter)” to display a string displaying number of faces detected. The using the function “cv2.Imshow()”with parameter “(frame)” to display our real time video capturing. And function “.release()”to show the video.

We also use a self made variable to save the data in a notepad file. For this we use python’s built-in function “datetime.now()”to show the time when the data is saved at. For the initialization condition we use our face counter as “face\_counter>0” then opening our “data.txt” . On execution it overwrites our data file with the time of execution “datetime.now” and numbers of faces detected “face\_counter”

* **Step II : Graphic User Interface**

For GUI we are working on Tkinter. First we import tKinter. Then we make the size of our frame by using the function “Tk.geometry()” passing a resolution as the parameter. But to go full screen we use the function “.attribute()” and passing “(“fullscreen”,true)” as the parameter. By passing the function “frame()” we actually work on different aspects like border , border color, border width. Then with the help of function “frame.pack()” and in the parameter we define the location by using attribute “side=” .For example “frame.pack(side=bottom)” places that border and its element in the bottom of the screen. We define the function “tick()” such that it may display time. Then we created another function “fd()” which is used to combine some of the snippets of the cade to capture live video. The function “label()” is used to present a text or image. Finally we create our button by a tkinter built function “button()” and in the parameters we customize our button. Some of the customization options that we chose are : bg ,fg ,height, width, text, font and command.

Finally using “.pack()” to organize the button in our GUI frame providing it with a loction

END