**Department of Information Technology**

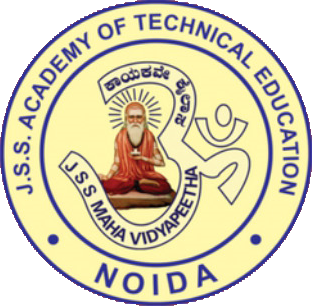
MINI PROJECT REPORT

ON

PYTHON

Submitted By:

RIYA MAHAJAN (2100910130089)



JSS Academy of Technical Education, NOIDA

Dr.APJ Abdul Kalam Technical University, Lucknow, U.P

Session 2022-2023

**Introduction:**

Python is a dynamic, interpreted (bytecode-compiled) language. There are no type declarations of variables, parameters, functions, or methods in source code. This makes the code short and flexible, and you lose the compile-time type checking of the source code. Python tracks the types of all values at runtime and flags code that does not make sense as it runs.

An excellent way to see how Python code works is to run the Python interpreter and type code right into it. If you ever have a question like, "What happens if I add an int to a list?" Just typing it into the Python interpreter is a fast and likely the best way to see what happens. (See below to see what really happens!)

The two lines python prints after you type python and before the >>> prompt tells you about the version of python you're using and where it was built. As long as the first thing printed is "Python 3.”, these examples should work for you.

As you can see above, it's easy to experiment with variables and operators. Also, the interpreter throws, or "raises" in Python parlance, a runtime error if the code tries to read a variable that has not been assigned a value. Like C++ and Java, Python is case sensitive so "a" and "A" are different variables. The end of a line marks the end of a statement, so unlike C++ and Java, Python does not require a semicolon at the end of each statement. Comments begin with a '#' and extend to the end of the line.

Python source files use the ".py" extension and are called "modules." With a Python module hello.py, the easiest way to run it is with the shell command "python hello.py Alice" which calls the Python interpreter to execute the code in hello.py, passing it the command line argument "Alice". See the [official docs page](http://docs.python.org/using/cmdline) on all the different options you have when running Python from the command-line.

Here's a very simple hello.py program (notice that blocks of code are delimited strictly using indentation rather than curly braces — more on this later!)

OpenCV is a Python library that allows you to perform image processing and computer vision tasks. It provides a wide range of features, including object detection, face recognition, and tracking. In this OpenCV Tutorial in Python, we’ll be learning more about the library.

OpenCV is an open-source software library for computer vision and machine learning. The OpenCV full form is Open Source Computer Vision Library. It was created to provide a shared infrastructure for applications for computer vision and to speed up the use of machine perception in consumer products. OpenCV, as a BSD-licensed software, makes it simple for companies to use and change the code. There are some predefined packages and libraries that make our life simple and OpenCV is one of them.

**Objective of the mini project:**

The objective of the course are as follows :

* Build basic programs using fundamental programming constructs like variables, conditional logic, looping, and functions.
* Work with user input to create fun and interactive programs.
* Create simple games with images, animations, and audio using our custom beginner-friendly programming library, Wizardlib.
* Can help to solve many real time problems in few lines of code like detection of liver cancer using image processing techniques.

**Learning Outcomes:**

* Learnt implementation of python.
* implementation of Open CV.
* This makes it a great choice to perform computationally intensive programs.
* Face detection technologies
* Haar feature-based cascade classifiers is an effectual machine learning based approach, in which a cascade function is trained using a sample that contains a lot of positive and negative images.

**Utilities of mini project :**

* It can Detect faces easily with less time of execution.
* It can Detect Face in any quality of Images even blur.
* It can Detect single, double or even multiple faces in any Image.
* It can tell the number of Peoples/faces in Image.
* It is fully automatic face recognition system
* It is one of the best in today world Innovation as per less number of code use in the system.
* It use Haar feature based Algorithm to detect which it makes very accurate and fast processing of face detection
* To identify and verify terrorists at airports, railway stations and malls the face recognition technology will be the best choice in India as compared with other biometric technologies since other technologies cannot be helpful in crowded places.
* This technology can also be used effectively in various important examinations such as SSC, HSC, Medical, Engineering, MCA, MBA, B- Pharmacy, Nursing courses etc. The examinee can be identified and verified using Face Recognition Technique.
* It can also be deployed in police station to identify and verify the criminals.
* To Identify the Number of student in classroom. It can collect in all the number of faces in less than 1 sec. Easy to take attendance.

**References :**

* [http://umpir.ump.edu.my/id/eprint/327/3/Real%20time%20face%20detection%20system%20-%20Chapter%201.pdf](https://www.sciencedirect.com/topics/computer-science/classifier-cascade)
* <https://www.sciencedirect.com/topics/computer-science/classifier-cascade>
* <https://ieeexplore.ieee.org/document/6315916>
* <https://link.springer.com/chapter/10.1007/978-3-642-72201-1_9>
* [https://thesai.org/Downloads/Volume9No6/Paper\_6-Study\_of\_Face\_Recog nition\_Techniques.pdf](https://thesai.org/Downloads/Volume9No6/Paper_6-Study_of_Face_Recog%20nition_Techniques.pdf)
* <http://www.pace.ac.in/documents/ece/FACE%20RECOGNITION%20SYSTEM%20WITH%20FACE%20DETECTION.pdf>

**PROJECT CODE:**

import cv2

# Load the cascade

face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

# To capture video from webcam.

cap = cv2.VideoCapture(0)

# To use a video file as input

# cap = cv2.VideoCapture('filename.mp4')

while True:

    # Read the frame

    \_, img = cap.read()

    # Convert to grayscale

    gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

    # Detect the faces

    faces = face\_cascade.detectMultiScale(gray, 1.1, 4)

    # Draw the rectangle around each face

    for (x, y, w, h) in faces:

        cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)

    # Display

    cv2.imshow('img', img)

    # Stop if escape key is pressed

    k = cv2.waitKey(30) & 0xff

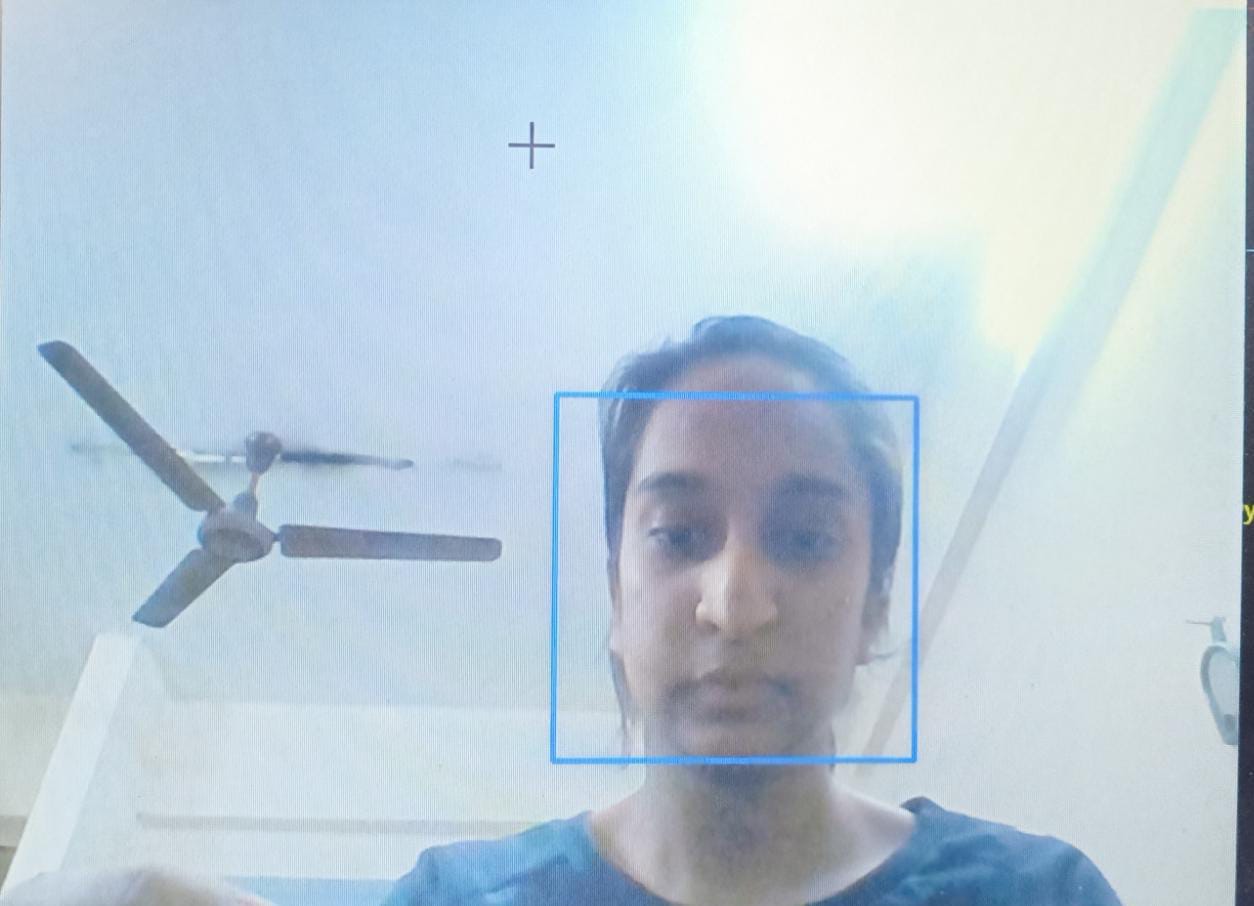
    if k==27:

        break

# Release the VideoCapture object

cap.release()

**PROJECT OUTPUT:**

****