PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology (Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in

Computer Science and Engineering

By

VULLI BHASKAR VENKATA SATISH
(Reg.No.37110848)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

AUGUST 2020

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" Grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.
Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344
www.sathyabama.ac.in

ISO 9001:2015

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VULLI BHASKAR VENKATA SATISH** (**37110848**) who carried out the project entitled "**PHOTOCAPTURE**" under our supervision from Jan 2019 to April 2019.

Internal Guide

Dr. S. Prince Marry.,

Head of the Department

Dr. VIGNESHWARI.S, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner **External Examiner**

DECLARATION

I ,**VULLI BHASKAR VENKATA SATISH** (Reg.No.**37110848**) hereby declare that the Project Report entitled "**PHOTOCAPTURE**" done by me under the guidance of **Dr. S. Prince Marry** at sathyabama institute of science and technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks **to Dr. SASIKALA.T, Ph.D., Dean, School of Computing** and **Dr. VIGNESHWARI.S, M.E., Ph.D., Head of the Department**, Department of **COMPUTER SCIENCE AND ENGINEERING** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide**.**
Dr. VIGNESHWARI.S, M.E., Ph.D., for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **COMPUETR SCIENCE AND ENGINEERING** who were helpful in many ways for the completion of the project.

Training Certificate

ABSTRACT

Face Recognition is a currently developing technology with multiple real- life applications. The developed system uses Convolutional Neural Networks in order to extract relevant facial features. These features allow to compare faces between them in an efficient way. The system can be trained to recognize a set of people, and to learn in an on-line way, by integrating the new people it processes and improving its predictions on the ones it already has. The accuracy in a set of 100 people has surpassed the 95%, and it has proven to robustly scale along with the number of people in the system. Two applications have been developed by user which makes use of this Face Recognition technology.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

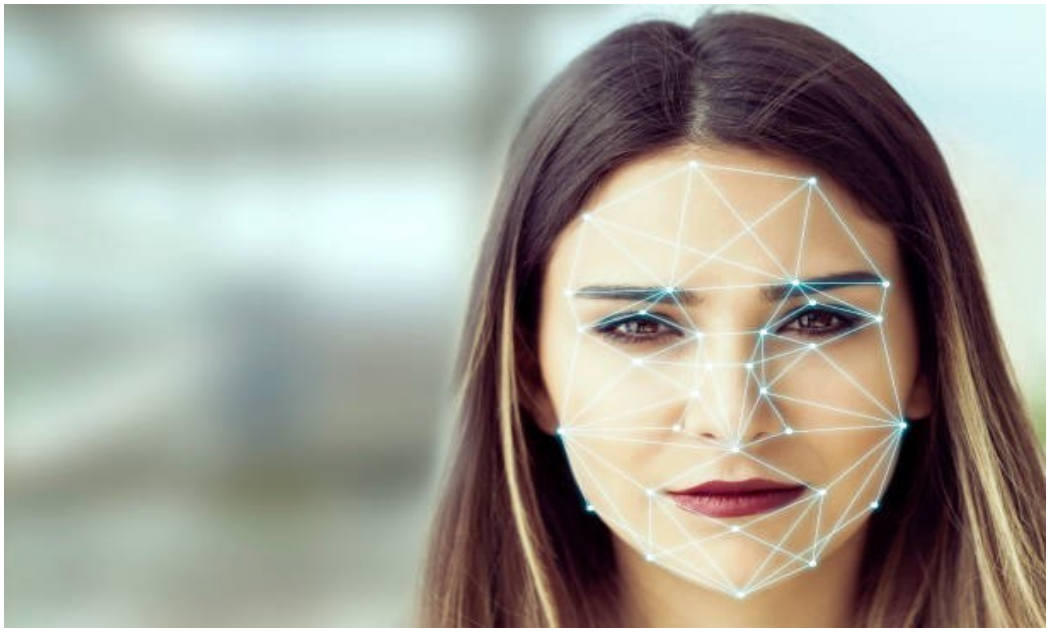
INTRODUCTION

The face can be considered as the unique identity of an individual. People across the world have unique faces and facial features. It plays a major role in interacting with other people in society. Considering these facts, facial recognition is implemented in the real world.

What is a Facial Recognition System?

In simple words, a Facial Recognition System can be defined as a technology that can identify or verify a person from a digital image or video source by comparing and analyzing patterns based on the person's facial contours.

Starting from the mid-1900s, scientists have been working on using computers to recognize human faces. Face recognition has received substantial attention from researchers due to its wide range applications in the real world.



WHY FACIAL RECOGNITION IS IMPORTANT?

Since face is a unique way of identifying people, facial recognition has gained high attention and growing rapidly across the world for providing safe and reliable security. It is gaining significant importance by corporate companies and government organizations because of its high level of security and reliability.

Facial recognition is now considered to have more advantages when compared to other biometric systems like palm print and fingerprint since facial recognition doesn't need any human interaction and can be taken without a person's knowledge which can be highly useful in identifying the human activities found in various applications of security like airport, criminal detection, face tracking, forensic, etc.

HOW TO BUILD A FACIAL RECOGNITION MODEL?

Over the years there were many methods used to implement facial recognition models but thanks to Artificial Intelligence it made our life easier. Using Deep Learning(part of AI), provided with the sufficient data a Facial Recognition System can be built simply. We use the OpenCV to build a simple Face Recognition Model.

WHAT IS OPEN CV?

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products.

INSTALLING AND IMPORTING OPEN CV

To install OpenCV on your machine, simply run the following commands based on your OS:

For Windows: pip install opencv-python

For Ubuntu: sudo apt-get install python-opencv

To import the OpenCV library simply run **import cv2**

1.3 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

In computer science, artificial intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligencedisplayed by humans. Colloquially, the term "artificial intelligence" is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving". As machines become increasingly capable, tasks considered to require "intelligence" are often removed from the definition of AI, a phenomenon known as the AI effect. A quip in Tesler's Theorem says "AI is whatever hasn't been done yet." For instance, optical character recognition is frequently excluded from things considered to be AI, having become a routine technology. Modern machine capabilities generally classified as AI include successfully understanding human speech, competing at the highest level in strategic game systems (such as chess and Go), autonomously operating cars, intelligent routing in content delivery networks, and military simulations. Artificial intelligence can be classified into three different types of systems: analytical, human-inspired, and humanized artificial intelligence. Analytical AI has only characteristics consistent with cognitive intelligence; generating a cognitive representation of the world and using learning based on past experience to inform future decisions. Human-inspired AI has elements from cognitive and emotional intelligence; understanding human emotions, in addition to cognitive elements, and considering them in their decision making. Humanized AI shows characteristics of all types of competencies (i.e., cognitive, emotional, and social intelligence), is able to be self-consciousand is self-aware in interactions. Artificial intelligence was founded as an academic discipline in 1956,

and in the years since has experienced several waves of optimism, followed by disappointment and the loss of funding (known as an "AI winter"), followed by new approaches, success and renewed funding.

1.4 HISTORY OF ARTIFICIAL INTELLIGENCE

Thought-capable artificial beings appeared as storytelling devices in antiquity, and have been common in fiction, as in Mary Shelley's Frankenstein or Karel Čapek's R.U.R. (Rossum's Universal Robots). These characters and their fates raised many of the same issues now discussed in the ethics of artificial intelligence. The study of mechanical or "formal" reasoning began with philosophers and mathematicians in antiquity. The study of mathematical logic led directly to Alan Turing's theory of computation, which suggested that a machine, by shuffling symbols as simple as "0" and "1", could simulate any conceivable act of mathematical deduction. This insight, that digital computers can simulate any process of formal reasoning, is known as the Church–Turing thesis. Along with concurrent discoveries in neurobiology, information theory and cybernetics, this led researchers to consider the possibility of building an electronic brain. Turing proposed that "if a human could not distinguish between responses from a machine and a human, the machine could be considered "intelligent". The first work that is now generally recognized as AI was McCullouch and Pitts' 1943 formal design for Turing-complete "artificial neurons". The field of AI research was born at a workshop at Dartmouth College in 1956. Attendees Allen Newell (CMU), Herbert Simon (CMU), John McCarthy (MIT), Marvin Minsky (MIT) and Arthur Samuel (IBM) became the founders and leaders of AI research. They and their students produced programs that the press described as "astonishing": computers were learning checkers strategies (c. 1954) (and by 1959 were reportedly playing better than the average human), solving word problems in algebra, proving logical

theorems (Logic Theorist, first run c. 1956) and speaking English. By the middle of the 1960s, research in the U.S. was heavily funded by the Department of Defense and laboratories had been established around the world.

ROLE OF ARTIFICIAL INTELLIGENCE IN FACIAL IDENTIFICATION

Today, various organizations are developing face recognition capabilities based on Artificial Intelligence. Facebook has developed a Deep Learning facial recognition system called "DeepFace." Deep Learning is an AI-based Machine Learning technique concerned with algorithms inspired by the human brain's neural networks. Deep Learning makes it possible to use brain simulations and make learning algorithms better and easier to use. Facebook makes use of a nine-layered neural network which has 120 million connection weight and is trained on 4 million connection images uploaded by the Facebook users. The system is said to be 97% accurate.

Similarly, Google's FaceNet is claimed to be a very accurate method for face identification achieving nearly 86% accuracy. It has an image data set with nearly 260 million images from all over the world, and it can give a name to a face and present images that match the face search.

Some models use predictive modeling techniques to incorporate facial data to understand how humans age. The method has been tested by a process named "de-aging," which involved taking a picture of an old person and running the Deep Learning algorithms backward to create a younger version of the same person. This younger image was then matched with the original image of the person in his younger days and the results were quite accurate. While there were older studies done on age progression work, this one was far more accurate.

Due to the massive amounts of data sets available for research and the Deep Learning algorithms ability to process this data and put it into work, today's facial identification technology is going places.

Use-Cases for Facial Identification

Facial identification can have extremely relevant use-cases in several industries.

Passport and Visas —It can control fraud detection for passports and visas. Already, the Australian Passport office is using the automatic face-recognition software, reporting 20% more efficiency in detecting fraud.

Banking —It is also useful in financial institutions like banks and their ATMs. China started using this technology in their ATMs. This has ensured a higher level of security for the card user.

Law Enforcement — Law enforcement agencies can deploy facial recognition systems for the identification of criminals. Many countries including the USA are developing their facial recognition database which will aid the criminal investigations making them swifter and more accurate.

Marketing — In 2013, retail giant Tesco rolled out targeted ads based on demographics like the gender and age of the customers at its petrol stations. It used facial recognition to identify these demographics using an AI-based software. Today, many more companies are aiming relevant advertising to its customer's basis facial identification as they enter the store by changing the display board ads to suit their personal preferences.

AI-powered facial identification can prevent fraud voting, track attendance and many such tasks in an error-free manner which would also be free from human bias. No matter the application and uses, Artificial Intelligence drives the facial recognition systems used across companies. Subsets of Artificial Intelligence such as Machine Learning and Deep Learning are enabling the collation and processing of troves of images which are compared for facial recognition. Without Artificial Intelligence algorithms, facial recognition systems would be far behind in their existence — both in terms of accuracy as well as speed. Artificial Intelligence is at the heart of facial identification.

REAL-WORLD APPLICATIONS OF FACIAL RECOGNITION

1. In the present world, Facial Recognition is being extensively used in surveillance systems.

2. It is also being used in crime detection and forensic. US Federal Bureau of Investigation is using face recognition to identify suspects from their driver's licenses. AI equipped cameras have also been trialed in the UK to identify those smuggling contraband into prisons.

3. Facial recognition is also being used in payments to make secure and reliable online payments.

CHAPTER 2

AIM AND SCOPE

2.1 AIM

Aim of the project is the act of trying to detect the number of faces and to capture the image of the faces that comes before the camera.

2.2 SCOPE

A facial recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. One of the way is to do this is by comparing selected facial features from the image. It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems.

CHAPTER 3

SYSTEM IMPLEMENTATION

3.1 INTRODUCTION TO PYTHON

Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and objectoriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is often described as a "batteries included" language due to its comprehensive standard library.

3.2 HISTORY OF PYTHON

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum continued as Python's lead developer until July 12, 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator For Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January, 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "Steering Council" to lead the project. Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector and support for Unicode.

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python 2.6.x and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.

3.3 FEATURES OF PYTHON

1. Python is a multiprogram paradigm language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

2. Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

3. Python's design offers some support for functional programming in the Lisp tradition. It has filter, map, and reduce functions; list comprehensions, dictionaries, sets and generator expressions.

4. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

5. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

6. Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one—and preferably only one—obvious way to do it" design philosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, writes that "To describe something as 'clever' is not considered a compliment in the Python culture."

3.3.1 MAIN FEATURES

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many

standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing. Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of March 2018, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 130,000 packages with a wide range of functionality, including:

- Graphical user interfaces
- Web frameworks
- Multimedia
- Databases
- Networking
- Test frameworks
- Automation
- Web scraping[101]
- Documentation
- System administration
- Scientific computing
- Text processing

- Image processing

3.4 ADVANTAGES

1) Easy to code

2) Contains rich libraries and are easy and direct to implement

3) Easily compatible on any os

4) Reduces the size of code

3.5 SOFTWARE REQUIREMENTS

The software requirements are:

1) Any OS

2) Python IDLE

3) Ram –minimum 4GB

4) Anaconda Navigator

3.6 HARWARE REQUIREMENTS

1) Keyboard

2) Monitor

3.7 LIBRARIES REQUIRED FOR SHARE PRICE PREDICTION

As we know, Python is an open source programming language. You may find many libraries to perform one function. The library used in the smart security system is:

1) Opencv

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the

commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The license implies that it is free to use for both academic and commercial use. It supports a couple of programming languages namely: python, java, c and C++. On the other hand, it supports Windows, Linux, Mac Os and even the Android operating systems.

Features of OpenCV Library

- Read and write images.
- Capture and save videos.
- Process images (filter, transform)
- Perform **feature** detection.
- Detect specific objects such as faces, eyes, cars, in the videos or images.
- Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

CHAPTER 4

RESULTS

4.1 RESULTS

Thus the program for smart security system using “ARTIFICIAL INTELLIGENCE” has been executed successfully. The execution is done in two parts i.e, testing and training.

4.2 SCREENSHOTS

Figure 4.2.1

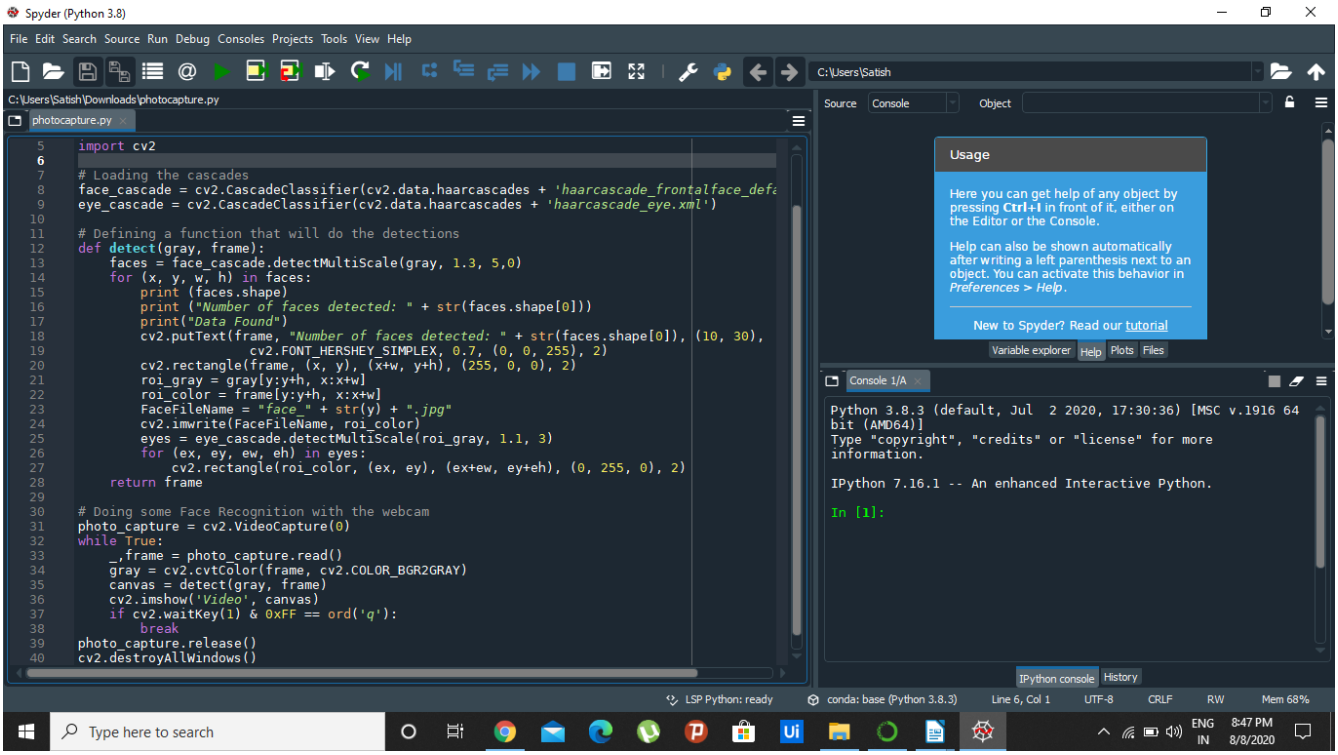
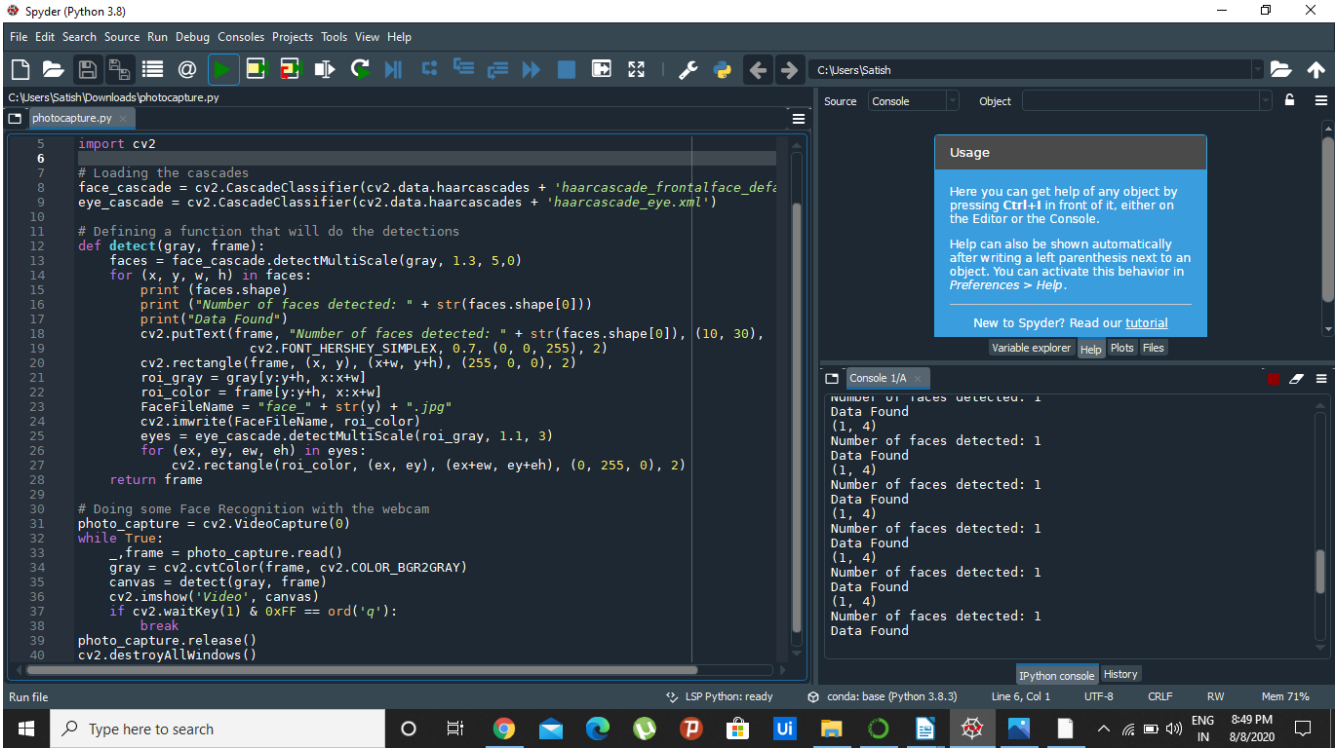


Figure 4.2.2


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Satish\Downloads\photocapture.py
photocapture.py
import cv2
# Loading the cascades
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_defa
eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')
# Defining a function that will do the detections
def detect(gray, frame):
faces = face_cascade.detectMultiScale(gray, 1.3, 5,0)
for (x, y, w, h) in faces:
print (faces.shape)
print ("Number of faces detected: " + str(faces.shape[0]))
print("Data Found")
cv2.putText(frame, "Number of faces detected: " + str(faces.shape[0]), (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
roi_gray = gray[y:y+h, x:x+w]
roi_color = frame[y:y+h, x:x+w]
FaceFileName = "face_" + str(y) + ".jpg"
cv2.imwrite(FaceFileName, roi_color)
eyes = eye_cascade.detectMultiScale(roi_gray, 1.1, 3)
for (ex, ey, ew, eh) in eyes:
cv2.rectangle(roi_color, (ex, ey), (ex+ew, ey+eh), (0, 255, 0), 2)
return frame
# Doing some Face Recognition with the webcam
photo_capture = cv2.VideoCapture(0)
while True:
_,frame = photo_capture.read()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
canvas = detect(gray, frame)
cv2.imshow('Video', canvas)
if cv2.waitKey(1) & 0xFF == ord('q'):
break
photo_capture.release()
cv2.destroyAllWindows()
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Console 1/A
Data Found
(1, 4)
Number of faces detected: 1
Run file
LSP Python: ready
conda: base (Python 3.8.3)
Line 6, Col 1
UTF-8
CRLF
RW
Mem 71%
Type here to search
8:49 PM
8/8/2020

CHAPTER 5
CONCLUSION

5.1 CONCLUSION:

Hence the above project is done successfully using artificial intelligence and it can be used in many areas for security reasons and it can determine the number of faces before the camera without manual help and captures the images of every face before the camera.

APPENDIX

A. SOURCE CODE

```
# Face Recognition

# Importing the libraries
import cv2

# Loading the cascades
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

# Defining a function that will do the detections
def detect(gray, frame):
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x, y, w, h) in faces:
        print (faces.shape)
        print ("Number of faces detected: " + str(faces.shape[0]))
        print("Data Found")
       cv2.putText(frame, "Number of faces detected: " + str(faces.shape[0]), (10, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h, x:x+w]
        FaceFileName = "face_" + str(y) + ".jpg"
        cv2.imwrite(FaceFileName, roi_color)
        eyes = eye_cascade.detectMultiScale(roi_gray, 1.1, 3)
        for (ex, ey, ew, eh) in eyes:
```

```
    cv2.rectangle(roi_color, (ex, ey), (ex+ew, ey+eh), (0, 255, 0), 2)
    return frame



# Doing some Face Recognition with the webcam
photo_capture = cv2.VideoCapture(0)
while True:
    _,frame = photo_capture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    canvas = detect(gray, frame)
    cv2.imshow('Video', canvas)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
photo_capture.release()
cv2.destroyAllWindows()
```

B.SCREEN SHOTS

Figure 1:

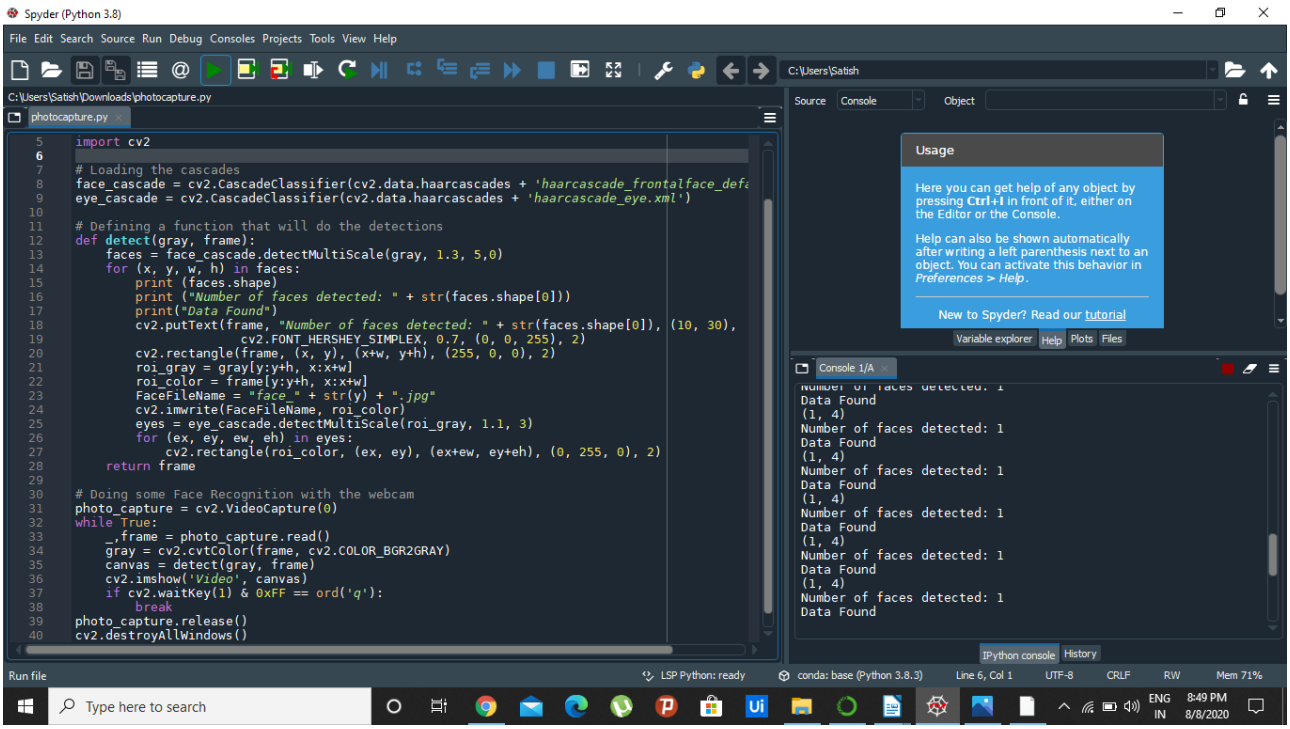
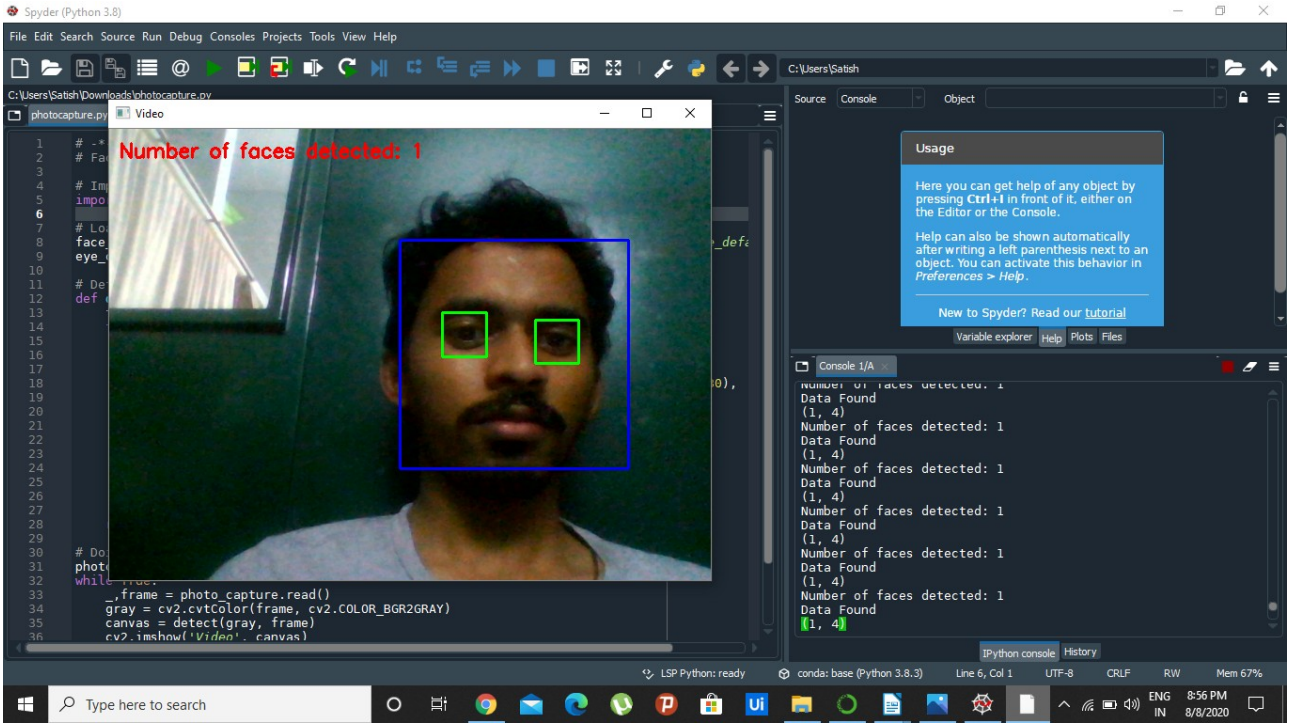


Figure 2:


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Video
Number of faces detected: 1
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Data Found
(1, 4)
Number of faces detected: 1
_,frame = photo_capture.read()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
canvas = detect(gray, frame)
Type here to search