**PROFESSIONAL INDUSTRIAL TRAINING**

**SUBMITTED TO**

**Department of Computer Science**

**And Engineering**

**Model Institute of Engineering and Technology (Autonomous)**

**Jammu, India**

**/ INTERNSHIP REPORT**

**ON**

“FACE MASK DETECTION”

## AT

**EXCELLENCE TECHNOLOGYPVT. LTD. CHANDIGARH**

## AN INDUSTRY INTERNSHIP REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

**FOR THE AWARD OF DEGREE OF**

# BACHELOR OF ENGINEERING

**In**

# Department of Computer Science and Engineering

**SUBMITTED BY**

PRASOON JASROTIA

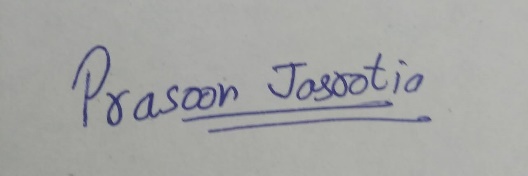
Roll Number: 2020A1R054



**CANDIDATES’ DECLARATION**

I, **PRASOON JASROTIA, Roll Number (2020A1R054)** hereby declare that the work which is being presented in the Industry Internship Report entitled, “**FACE MASK DETECTION”** in partial fulfillment of requirement for the award of degree of

B.E. (CSE) and submitted in the Department of Computer Science and Engineering, Model Institute of Engineering and Technology (Autonomous), Jammu is an authentic record of my own work carried by me at “EXCELLIENCE TECHONOLOGY PVT. LTD, CHANDIGARH” under the supervision and mentorship of **Ms. DEEPAK KASHYAP** (BRANCH MANAGER of EXCELLIENCE TECHONOLOGY PVT. LTD). The matter presented in this report has not been submitted in this or any other University / Institute for the award of B.E. Degree.



*Signature of the Student Dated*:

## PRASOON JASROTIA 20/10/2022

**2020A1R054**

**Department Of Computer Science and Engineering**

**Model Institute of Engineering and Technology (Autonomous)**

**Kott Bhalwal, Jammu, India**

***(NAAC “A” Grade Accredited)***

**Ref. No.: 2020A1R054 Date:20/10/2022**

# CERTIFICATE

Certified that this Industry Internship Report entitled **“FACE MASK DETECTION”** is the bonafide work of “**PRASOON JASROTIA, Roll No. 2020A1R054, of 5th Semester, Department of Computer Science and Engineering, Model Institute of Engineering and Technology (Autonomous), Jammu”,** who carried out the Industry Internship at **“EXCELLIENCE TECHONOLOGY PVT. LTD. CHANDIGARH”** work under my mentorship during 8th July,2022 to 31th August, 2022.

## Dr,Mekhla Sharma

## Assistant Professor

## Department of Computer Science and Engineering, MIET

*This is to certify that the above statement is correct to the best of my knowledge.*

**Dr. ASHOK KUMAR**

**DEAN ACADEMICS**

**HOD CSE**

**MIET**

**

**

**

# ACKNOWLEDGEMENTS

This Summer internship opportunity was a great chance for learning and professional development. I am grateful for having a chance to meet so many wonderful people and professionals who led me though this internship period.

It is my pleasant duty to pay my heartfelt gratitude **to Mr. Deepak Kashyap, branch manager EXCLLLIENCE TECHONOLOGY PVT. LTD** who have guided me through the course of this Internship.

I must record my deep sense of gratitude to **Prof. Dr. Ankur Gupta** (Director, MIET) and **Prof. Dr. Ashok Kumar** (Dean Academics & HOD CSE, MIET) for their guidance, constant inspiration and encouragement, and for their keen involvement throughout the course of present work.

Gratitude and thanks although mean a very small thing to convey my thanks to my parents who have always given me a parental source of love, motivation and strength right from the journey of my life.

Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to the teachers who in spite of being extraordinarily busy with their duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my project at their esteemed organization and extending during the training.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future.

I express my sincere gratitude **to EXCLLIENCE TECHONOLOGY**, Chandigarh and **Model Institute of Engineering and Technology (Autonomous),** Jammu for giving me the opportunity.

Prasoon Jasrotia

2020A1R054

|  |  |  |
| --- | --- | --- |
| **CHAPTERS** | **TOPIC** | **PAGE NO.** |

|  |  |
| --- | --- |
| **INTRODUCTION** | 12 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER 1** | **INTRODUCTION TO PYTHON**   1. Python history and applications 2. Using python interpreter 3. Python script file 4. Print message to standard output 5. Variables and data types 6. Reading input from console 7. Type conversion 8. Arithmetic operators and conditions | 13  13  15  15  16  16  18  18  19 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER 2** | **ARTIFICIAL INTELLIGENCE**   1. What is Artificial Intelligence 2. Why Artificial Intelligence? 3. Advantages of Artificial Intelligence 4. Artificial Intelligence and Python: Why? 5. Python Libraries for General Artificial Intelligence   Python for Machine Language (ML)   1. Conclusion | 21  22  23  23  23  24  24  25 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER 3** | **RELATED WORKS** | 26 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER 4** | **PYTHON PROGRAMMING**   1. Control flow 2. Operators 3. If - Else statement 4. If - Elif statement 5. Loops | 29  29  29  30  30  30 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER 5** | **PROPOSED METHODOLOGY**   1. The Input Image 2. The Pre-Processing Stage 3. The Face Detection Stage 4. The Feature Extraction Stage 5. The Classification Stage 6. Training stage 7. Prediction Stage | 33  33  33  34  34  34  34  35 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER 6** | **MODULES**   1. Pre-Processing Image 2. Capture Image 3. Upload Image 4. Classifier Image 5. Prediction Image | 36  36  36  36  37  37 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER 7** | **SYSTEM ARCHITECTURE**   1. User 2. Capture Images 3. Face Detection 4. Mask Detection 5. Person Identification 6. Dataset 7. Data Pre-Processing 8. Segmentation 9. Edge Detection 10. Localization 11. Feature Selection 12. CNN Architecture Creation 13. Training 14. Predication   RESULT  CONCLUSION | 38  39  39  39  39  39  41  41  41  41  42  42  42  43  43  44  45 |

|  |  |
| --- | --- |
| **PROJECT**  **FACE MAZK DETECTION** | **47** |

|  |  |
| --- | --- |
| FIGURES | PAGE NO. |

|  |  |
| --- | --- |
| 1.1  1.2  1.3 | 15  16  17 |

|  |  |
| --- | --- |
| 2.1 | 22 |

|  |  |
| --- | --- |
| 7.1  7.2  7.3 | 38  40  44 |

**INTRODUCTION**

Face Recognition is a technique that matches stored models of each human face in a group of people to identify a person based on certain features of that person’s face. Face recognition is a natural method of recognizing and authenticating people. Face recognition is an integral part of people's everyday contact and lives. The security and authentication of an individual is critical in every industry or institution. As a result, there is a great deal of interest in automated face recognition using computers or devices for identity verification around the clock and even remotely in today's world. Face recognition has emerged as one of the most difficult and intriguing problems in pattern recognition and image processing. With the aid of such a technology, one can easily detect a person's face by using a dataset of identical matching appearance. The most effective approach for detecting a person's face is to use Python and a Convolutional Neural Network in deep learning. This method is useful in a variety of fields, including the military, defense, schools, colleges, and universities, airlines, banks, online web apps, gaming, and so on. Face masks are now widely used as part of standard virus prevention measures, especially during the Covid-19 virus outbreak. Many individuals or organizations must be able to distinguish whether or not people are wearing face masks in a given location or time. This data's requirements should be very real-time and automated. The challenging issue which can be mentioned in face detection is inherent diversity in faces such as shape, texture, color, got a beard\moustache and/or glasses and even masks. From the experiments it is clear that the proposed CNN and Python algorithm is very efficient and accurate in determining the facial recognition and detection of individuals.

**CHAPTER 1**

**INTRODUCTION TO PYTHON**

* 1. **PYTHON HISTORY AND APPLICATIONS**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level Programming language.

It was created by Guido van Rossum during 1985- 1990.Like Perl, Python source code is also available under the GNU General Public License (GPL). Python was developed by Guido Van Rossum in the Netherlands.

Python was released in 1991. Python was developed at CWI (Centrum Wiskunde & Informatica) which is

the National Research Institute for Mathematics and Computer Science in the Netherlands. Guido Van

Rossum and his team were using "ABC" language but it was not good at “exception handling" then he

decided to develop a new language which is later named Python. Python is copyrighted like many other

languages but now its source code is easily available under GPL (General Purpose License). The latest

version is Python 3 Python is a high-level, interpreted, interactive and object-oriented scripting language.

Python is designed to be highly readable. It uses English keywords frequently where as other languages use

punctuation, and it has fewer syntactical constructions than other languages.

Python can do lots of magic for you. It can be used in many applications in different

domains. Here are some:

a. Python can be used for Web Development (Django and Flask are Web framework based on Python

b. Python can be used in Data Analysis to simplify the various difficult task and create charts and reports in seconds.

c. Python can be used in Machine Learning where you can train the machine to identify users' interest from users' activities. Face and voice recognition applications are the best examples of Machine Learning.

d. Python can be used to develop games.

e. Python can be used as a script to automate some tasks to save time.

f. Python can be used to develop desktop applications using libraries available in Python.

**APPLICATIONS OF PYTHON**

1. **Web Development:** Python is extremely used in the development of server-side web applications. Python

gives you the power to choose the framework you want to work with. Django and Pyramid are the popular

frameworks available in Python. Bottle and Flask are micro-frameworks to be used for web development. CMS

(Content Management System) Plone and Django also there for fast web development. These frameworks have

been used by Some of the world’s most amazing sites such as Spotify, Mozilla, Reddit, the Washington Post,

and Yelp. Python's standard library provides great supports for many Internet protocols: HTML, XML, JSON,

E-mail processing, Support for FTP, IMAP, other Internet protocols and Easy-to-use socket interface.

2. **Desktop GUI Development:** Python provides many solutions to develop a Graphical User Interface (GUI).

Python shipped with a toolkit named tkinter which is widely used for GUI developments. By using Python

with tkinter you can create GUI Applications very fast and easy. Some toolkits wxwidgets, Kivy (for

multitouch applications), Qt via pyqt is usable on several Platforms are available separately.

3. **Scientific and Numeric Applications:** Python is widely used in various scientific and numeric applications

in the whole world. It has many packages and libraries for the implementation of numerical and statistical data.

Some of them are given below:

a. ***Numpy:*** It is the library for Python for multidimensional array and matrices along with bundle of special mathematical functions to operate on these arrays.

b. ***Scipy:*** It is the tool which is free to use for Python. It has a variety of modules for linear algebra, signal and image processing, statistics, genetic algorithms.

c. ***Pandas:*** It is a widely used and popular library for data analysis.it is open source and free to use. Panda library provides very powerful and flexible data structures that make data manipulation and analysis easy and customized. Apart from the above libraries, Python has APM Python, sympy, pygsl, funcdesigner important libraries for doing Scientific and Numerical work.

4. **Education:** Python is the best choice for students to learn programming skills. It is very easy to learn and use.

Applications build in Python has extensive use in education to make education interesting and fun.

5. **Business Applications:** Python is being used to solve business problems. People are using Python to build Ecommerceand

various customized ERP systems. In these applications, you can create any kind of student

management system, feemanagement systems and online shopping website to provide services to your

customers.

6. **Audio/Video Applications:** Audio and Video applications are on top these days. You can see huge growth in

this sector.Python has libraries that can be used to analyze audio/video content.

a. ***Librosa:*** Music and audio analysis can be done by using a library known as librosa.

b. ***Pyaudioanalysis:*** This library can be used to perform the various audio analysis. It includes Feature Extraction, Classification, Segmentation.

c. ***Pylivestream:*** This Python library able you to streams to streaming sites simultaneously, it can be done by applying pure object-oriented Python and ffmpeg. Users can stream to Facebook Live, youtube Live and etc.

7. **Game Development:** Gaming industry is increasing day by day. Python can be used for game development.

There are many applications which use python to support games and in the development of games. All the

addons of the Battlefield 2 game and its functionality uses Python. World of Tanks uses Python for many

functionalities. Python provides you many tools and libraries for development of games. Python is extensible and you can use Panda3D engine to develop 3D games in Python.

8. **Machine Learning:** Machine learning is an application of Artificial Intelligence (AI). It focuses on the

development of programs that can access data and use it to learn for themselves. Python has enough machine

learning libraries available for free, like pandas or scikit, which can handle data and produce results very

quickly and efficiently. All these libraries are available under GNU license. Here is the list of some various

applications you can develop using machine learning and Python

a. Voice Recognition

b. Predicting Music Choices

c. Face Recognition

d. Product Recommendation

e. Online Froud Detection

9. **Database Integration:** Python supports a variety of database to use in the development of your application.

Python gives you the flexibility to use ODBC interfaces to mysql, Oracle, postgresql, MS SQL Server, and

others are free to download and use.

10. **Web Scraping:** Web Scraping is very popular these days and it is used to collect data from websites to make

the final decision based on the collection of the data. Python allows you to perform web scraping on websites

very fast and you can pull huge data from websites in a few seconds. Python has a collection of libraries such

as NumPy, Matplotlib, Pandas to serve this purpose. Here is the list of applications based on Web Scraping.

a. Price Comparison: Website like Trivago, Policy Bazaar using Web Scraping

b. Social Media: To find the trends on social media.

c. Email Gathering: Many Companies using Web Scraping to collect Email to use in marketing to send emails for the promotion of their brands or product.

d. Jobs Portal: Collect data from different websites on the job opening in various domains and use that data in a single website.

**1.2 USING PYTHON INTERPRETER**

1.2.1 **Invoking the Interpreter:** The Python interpreter is usually installed as /usr/local/bin/python3.11 on those

machines where it is available; putting /usr/local/bin in your Unix shell’s search path makes it possible to

start it by typing the command:

**Python3.11**

To the shell. 1 Since the choice of the directory where the interpreter lives is an installation option, other

places are possible; check with your local Python guru or system administrator. (E.g., /usr/local/python is a

popular alternative location.) On Windows machines where you have installed Python from the Microsoft

Store, the python3.11 command will be available



**1.3 PYTHON SCRIPT FILE**

A Python script, **the file containing the commands**, is structured to be executed like a program. These files

are designed to contain various functions and import various modules. Python interactive shell or the respective.

Command line is used to execute the script, to perform a specific task. Different ways to run python script are

a. Interactive mode

b. Command line

c. Text editor (VS Code)

d. IDE (PyCharm)

**1.4 PRINT MESSAGE TO STANDARD OUTPUT**



Every running program has a text output area called "standard out", or sometimes just "stdout". The Python print() function takes in python data such as ints and strings, and prints those values to standard out. To say that standard out is "text" here means a series of lines, where each line is a series of chars with a '\n' newline char marking the end of each line. Standard out is relatively simple. It is a single area of text shared by all the code in a program. Each printed line is appended at its end.

**a. Print Function**

The Python print() function takes in any number of parameters, and prints them out on one line of text. The items are each converted to text form, separated by spaces, and there is a single '\n' at the end (the "newline" char). When called with zero parameters, print() just prints the '\n' and nothing else. In the interpreter, standard-out displays to the screen in between the '>>>' prompts, so it's an easy way to see what print() does.

>>> **print(12, 24, -2)**

12 24 -2

>>> **print('hi', 'there', -2)**

Hi there -2

>>> **print('woot')** # 1 item, 1 \n

Woot

>>> **print()** # 0 items, 1 \n

**b. Print vs. Return**

Thinking about the black-box design of a function, return is the main way for a function to return results back to its caller. Standard-out is a secondary way for a function to communicate data out, but it is much simpler - standard out is textual (typically), and it is shared among all the functions. Therefore, use the return-value as the main, black-box data output and function testing mechanism. Standard-out is a secondary form of function output, most often used to produce text output for the user in the terminal.

**c. Print To File**

The super common case is printing to standard output. However, print() also works printing to an open file. To open a file for writing, add 'w' when calling the open() function (shown below) — 'w' for "writing". This deletes any existing contents of that file, so be careful when opening for writing like this. With the file open, the optional file= parameter to print() directs the text lines into the file instead of writing them to standard output. With open(filename, 'w') as f: Print('Hello world', file=f)

**1.5 VARIABLES AND DATATYPES**

Variables and data types in python as the name suggests are the values that vary. In a programming

language, a variable is a memory location where you store a value. The value that you have stored may

change in the future according to the specifications



**Variable in python** is created as soon as a value is assigned to it. It does not need any additional commands to declare a variable in python. There are a certain rules and regulations we have to follow while writing a variable, let’s take a look at the variable definition and declaration to understand how we declare a variable in python.

There is a certain rule that we have to keep in mind while declaring a variable:

a. The variable name cannot start with a number. It can only start with a character or an underscore.

b. Variables in python are case sensitive.

c. They can only contain alpha-numeric characters and underscores.

d. No special characters are allowed.

**Data Types in Python**

According to the properties they possess, there are mainly six data types in python. Although there is one more datatype range which is often used while working with loops in python.

a. **Numerical Data Types**

Numerical data type holds numerical value. In numerical data there are 4 sub types as well. Following are the subtypes of numerical data type:

1. Integers are used to represent whole number values.

2. Float data type is used to represent decimal point values.

3. Complex Numbers numbers are used to represent imaginary values. Imaginary values are denoted with ‘j’ at the end of the number

4. Boolean **is** used for categorical output, since the output of Boolean is either true or false

b. **Strings**

Strings in python are used to represent Unicode character values. Python does not have a character datatype, a single character is also considered as a string. We denote or declare the string values inside single quotes or double quotes.

To access the values in a string, we use the indexes and square brackets.

e.g., “Rohan”

c. **Lists**

List is one of the four collection data type that we have in python. When we are choosing a collection type, it is important to understand the functionality and limitations of the collection. Tuple, set and dictionary are the other collection data type is python. A list is ordered and changeable, unlike strings. We can add duplicate values as well.

To declare a list, we use the square brackets.

e.g. [2,3,” rohan”,”4”]

d. **Tuples**

Tuple is a collection which is unchangeable or immutable. It is ordered and the values can be accessed using the index values. A tuple can have duplicate values as well. To declare a tuple, we use the round brackets.

e.g. (1,2,3,1,5,6)

e. **Sets**

A set is a collection which is unordered, it does not have any indexes as well. To declare a set-in python, we use the curly brackets.

e.g. {2,3,4,5,6,7,8}

f. **Dictionary**

A dictionary is just like any other collection array in python. But they have key value pairs. A dictionary is unordered and changeable. We use the keys to access the items from a dictionary. To declare a dictionary, we use the curly brackets.

e.g. {‘rohan’:’2020a1r087’,’34’:’2’}

g. **Range**

Range is a data type which is mainly used when we are using a loop

h. **Type Casting**

Type casting basically is the process of changing one data type into another. We have constructors for each of the data types in python.

**1.6 READING INPUT FROM CONSOLE**

In this, we are going to learn how to take input from the console in Python. The **interactive** shell in *Python* is treated as a **console**. We can take the user entered data the **console** using **input()** function.

a = input() # taking input from the user

# printing the data

print("User data:-", a)

Tutorialspoint

User data:- Tutorialspoint

If you run the above code, then you will get the following result.

**Conclusion**

The data that is entered by the user will be in the string format. If you have any doubts in the tutorial, mention them in the comment section.

**1.7 TYPE CONVERSIONS**

The process of converting the value of one data type (integer, string, float, etc.) to another data type is called type conversion. Python has two types of type conversion.

**a. Implicit Type Conversion**

In Implicit type conversion, Python automatically converts one data type to another data type. This process

doesn't need any user involvement.

Let's see an example where Python promotes the conversion of the lower data type (integer) to the higher data type (float) to avoid data loss.

Example 1: Converting integer to float

num\_int = 123

num\_flo = 1.23

num\_new = num\_int + num\_flo

print("datatype of num\_int:",type(num\_int))

print("datatype of num\_flo:",type(num\_flo))

print("Value of num\_new:",num\_new)

print("datatype of num\_new:",type(num\_new))

the output will be:

datatype of num\_int: <class 'int'>

datatype of num\_flo: <class 'float'>

Value of num\_new: 124.23

datatype of num\_new: <class 'float'>

**b. Explicit Type Conversion**

In Explicit Type Conversion, users convert the data type of an object to required data type. We use the

predefined functions like int(), float(), str(), etc to perform explicit type conversion.

This type of conversion is also called typecasting because the user casts (changes) the data type of the objects.

Example 2. Addition of string and integer using explicit conversion

num\_int = 123

num\_str = "456"

print("Data type of num\_int:",type(num\_int))

print("Data type of num\_str before Type Casting:",type(num\_str))

num\_str = int(num\_str)

print("Data type of num\_str after Type Casting:",type(num\_str))

num\_sum = num\_int + num\_str

print("Sum of num\_int and num\_str:",num\_sum)

print("Data type of the sum:",type(num\_sum))

the output will be:

Data type of num\_int: <class 'int'>

Data type of num\_str before Type Casting: <class 'str'>

Data type of num\_str after Type Casting: <class 'int'>

Sum of num\_int and num\_str: 579

Data type of the sum: <class 'int'>

**1.8 ARITHEMETIC OPERATORS AND CONDITIONS**

a. Addition: This operator is used to add two values present on either side of the operator.

Input:

x = 2

y = 3

sum = x + y

print (sum)

Output:

5

b. Subtraction: This operator is used to subtract the value present on the right side of the operator from the value present on the left side of the operator.

Input:

x = 5

y = 2

sub = x - y

print (sub)

Output:

3

c. Multiplication: This operator is used to find the product of the two values present on either side of the operator.

Input:

x = 2

y = 3

mul = x \* y

print (mul)

Output:

6

d. Division: This operator is used to find the quotient. The value present on the left side of the operator acts as a Dividend and the one on the right side is Divisor.

Input:

x = 5

y = 2 9

div = x / y

print (div)

Output:

2.5

A division operation always results in a floating-point number.

e. Modulus: This operator is used to find the remainder. The value present on the left side of the operator acts as a Dividend and the one on the right side is Divisor.

Input:

x = 8

y = 3

mod = x % y

print (mod)

a = -5

b = 2

res1 = a % b

print (res1)

m = 5

n = -2

res2 = m % n

print (res2)

Output:

2

-1

1

The remainder will be positive if the Dividend is positive and vice-versa. Even if the Divisor is negative but the Dividend is positive, the remainder will be positive.

f. Exponentiation: This operator is used to raise the first value to the power of the second operator

Input:

x = 2

y = 3

exp = x \*\* y

print (exp)

Output:

8

g. Floor division: The Floor Division operator is used to floor the result to the nearest integer.

Input:

x = 5

y = 2

flo = x // y

print (flo)

Output:

2.0

**CHAPTER 2**

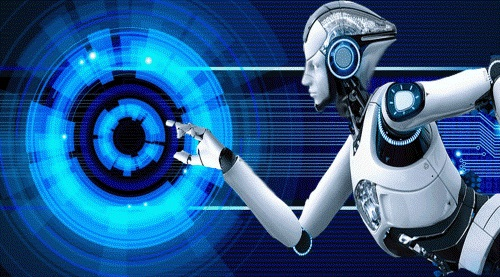
**ARTIFICIAL INTELLIGENCE**

**5.1 What is Artificial Intelligence?**

In today's world, technology is growing very fast, and we are getting in touch with different new technologies day by day.

Here, one of the booming technologies of computer science is Artificial Intelligence which is ready to create a new revolution in the world by making intelligent machines.The Artificial Intelligence is now all around us. It is currently working with a variety of subfields, ranging from general to specific, such as self-driving cars, playing chess, proving theorems, playing music, Painting, etc.

AI is one of the fascinating and universal fields of Computer science which has a great scope in future. AI holds a tendency to cause a machine to work as a human.



Artificial Intelligence is composed of two words **Artificial** and **Intelligence**, where Artificial defines *"man-made,"* and intelligence defines *"thinking power"*, hence AI means *"a man-made thinking power."*

So, we can define AI as:

 "It is a branch of computer science by which we can create intelligent machines which can behave like a human, think like humans, and be able to make decisions."

Artificial Intelligence exists when a machine can have human-based skills such as learning, reasoning, and solving problems

With Artificial Intelligence you do not need to preprogram a machine to do some work, despite that you can create a machine with programmed algorithms which can work with your own intelligence, and that is the awesomeness of AI.

It is believed that AI is not a new technology, and some people say that as per Greek myth, there were Mechanical men in the early days who can work and behave like humans.

**5.2 Why Artificial Intelligence?**

Before Learning about Artificial Intelligence, we should know that what is the importance of AI and why should we learn it. Following are some main reasons to learn about AI:

* With the help of AI, you can create such software or devices which can solve real-world problems very easily and with accuracy such as health issues, marketing, traffic issues, etc.
* With the help of AI, you can create your personal virtual Assistant, such as Cortana, Google Assistant, Siri, etc.
* With the help of AI, you can build such Robots which can work in an environment where survival of humans can be at risk.
* AI opens a path for other new technologies, new devices, and new Opportunities.

**5.3 Advantages of Artificial Intelligence**

Following are some main advantages of Artificial Intelligence:

* High Accuracy with less errors: AI machines or systems are prone to less errors and high accuracy as it takes decisions as per pre-experience or information.
* High-Speed: AI systems can be of very high-speed and fast-decision making, because of that AI systems can beat a chess champion in the Chess game.
* High reliability: AI machines are highly reliable and can perform the same action multiple times with high accuracy.
* Useful for risky areas: AI machines can be helpful in situations such as defusing a bomb, exploring the ocean floor, where to employ a human can be risky.
* Digital Assistant: AI can be very useful to provide digital assistant to the users such as AI technology is currently used by various E-commerce websites to show the products as per customer requirement.
* Useful as a public utility: AI can be very useful for public utilities such as a self-driving car which can make our journey safer and hassle-free, facial recognition for security purpose, Natural language processing to communicate with the human in human-language, etc.

**5.4 Artificial Intelligence and Python: Why?**

Python is the simplest language of all the programming languages, and in reality, is one-fifth when compared with other OOP languages. This is why it is currently among the most well-known languages in the marketplace.

* Python comes with Prebuilt Libraries such as Numpy to perform scientific calculations, Scipy for advanced computing, and Pybrain for machine learning (Python Machine Learning), making it among the top languages for AI.
* Python developers all over the globe offer extensive support and assistance through tutorials and forums, helping the programmer much easier than another popular language.
* Python is platform-independent and therefore is among the most adaptable and well-known options for various platforms and technologies, with minimal modifications to the basics of coding.
* Python has the greatest flexibility among other programs, with the option of choosing among OOPs method and scripting. Additionally, you can use the IDE to search for all codes and be a blessing to developers struggling with different algorithms.

**5.5 Python Libraries for General Artificial Intelligence**

* **AIMA** - Python algorithm implementation comes directly from Russell as well as Norvig's "Artificial intelligence: A Modern Methodology. "
* **pyDatalog** - Logic Programming engine in Python
* **SimpleAI** - Python implementation of various artificial intelligence algorithms explained in "Artificial Intelligence, a Modern Approach ". Its primary goal is to provide intuitive, well-documented, and tested libraries.
* **EasyAI** - A simple Python engine that allows two players to play games using AI (Negamax and transposition tables for Game-solving).

**5.6 Python for Machine Language (ML)**

Let's look at the reason Python is used in Machine Learning and the various libraries it provides for this reason.

* **PyBrain** - is an easy yet flexible algorithm to perform Machine Learning tasks. It also functions as an extensible Machine Learning Library for Python that provides a range of predefined environments for testing and evaluating algorithms.
* **PyML** - A bidirectional framework developed in Python concentrates on SVMs and other kernel-based approaches. It's accessible on Linux and Mac OS X.
* **Scikit-learn** - can be described as an effective instrument for data analysis making use of Python. This is a completely free and open-source library. It is the most widely used general-purpose machine learning library.
* **MDP** - Toolkit A different Python Data Processing Framework that is easily extended contains a variety of unsupervised and supervised learning algorithms and other computing units for data analysis that could be combined to create sequences of data processing and more intricate feed-forward networks. Implementation of the new algorithm is straightforward. The number of algorithms available is constantly growing. It includes signal processing techniques (Principal Component Analysis, Independent Component Analysis as well as slow Feature Analysis), manifold methods of learning ([Hessian Locally Linear embedding) as well as a number of classification methods, probabilistic techniques (Factor Analysis, RBM), and data pre-processing techniques, and many more.

**5.7 Conclusion**

Python is a key part of AI programming languages due to the fact that it has good frameworks, such as sci-kit-learn-Machine Learning in Python that meets almost all requirements in this area as well as D3.js data-driven documents JS. It is among the most efficient and user-friendly tools to visualize.

In addition to frameworks, Java's speedy prototyping makes Java an important language that shouldn't be missed. AI requires a lot of studies, which is why it is not necessary to have a 500-KB boilerplate for Java to test an innovative idea, but it will never be able to complete the project. In Python, nearly every concept is easily validated with twenty to thirty lines of code (same as JS using libraries). This makes it an extremely useful language to aid in the development of AI.

This is why it is clear that Python is the top AI Programming Language in the world. Alongside being the top AI programming language, Python has value for various other goals.

**CHAPTER 3**

**RELATED WORKS**

Local Binary Pattern (LBP) method is used to filter the candidate region. LBP reflects the details of the face characteristics, focusing on the description of texture features. Therefore, after using global feature to identify the face, the local feature recognition LBP method is used to filter the candidate region. LBP reflects the details of the face characteristics, focusing on the description of texture features. This paper combines skin color detection with LBP. If the number of pixels of the skin color points exceeds the set threshold, the face image is initially determined. Otherwise it is a non-face image. Then the LBP algorithm is used to detect the candidate window. If the match is successful, it is a face image. Otherwise it is non-human face. Otherwise it is non-human face. LBP is not capable of detecting faces with masks or glasses in faces. [1]

A robust approach to face & facial features detection must be able to handle the variation issues such as changes in imaging conditions, face appearances and image contents. Here we present a method in which utilizes color, local symmetry and geometry information of human face based on various models. The algorithm first detects most likely face regions or ROIs (Region-Of Interest) from the image using face color model and face outline model, produces a face color similarity map. Then it performs local symmetry detection within these ROIs to obtain a local symmetry similarity map. These two maps are fused to obtain potential facial feature points. Finally, similarity matching is performed to identify faces between the fusion map and face geometry model under affine transformation. The output results are the detected faces with confidence values [2].

Face detection and eyes extraction has an important role in many applications such as face recognition, facial expression analysis, security login etc. Detection of human face and facial structures like eyes, nose are the complex procedure for the computer. This paper proposes an algorithm for face detection and eyes extraction from frontal face images using Sobel edge detection and morphological operations. The proposed approach is divided into three phases; pre-processing, identification of face region, and extraction of eyes. Resizing of images and gray scale image conversion is achieved in pre-processing. Face region identification is accomplished by Sobel edge detection and morphological operations. In the last phase, eyes are extracted from the face region with the help of morphological operations. [3].

YOLO has a fast detection speed and is suitable for target detection in real-time environment. Compared with other similar target detection systems, it has better detection accuracy and faster detection time. This paper is based on YOLO network and applied to face detection. In this paper, YOLO target detection system is applied to face detection. Experimental results show that the face detection method based on YOLO has stronger robustness and faster detection speed. Still in a complex environment can guarantee the high detection accuracy. At the same time, the detection speed can meet real-time detection requirements [4].

Partially Occluded Face Detection (POFD) problem is addressed by using a combination of feature-based and part based face detection methods with the help of face part dictionary. In this approach, the devised algorithm aims to automatically detect face components individually and it starts from mostly un-occluded face component called Nose. Nose is very hard to cover up without drawing suspicion. Keeping nose component as a reference, algorithm search the surrounding area for other main facial features, if any. Once face parts qualify facial geometry, they are normalized (scale and rotational) and tag with annotation about each facial features so that partial face recognition algorithm can be adapted accordingly with the test image. [5].

We are focused on the face detection process and the role of interest regions of the human face. In order to locate exactly the facial area, we propose the use of horizontal and vertical IPC (Integral Projection Curves). The role of important patches of face: nose and eyes is investigated in this work. An efficient method based on PCA (Principal component analysis) followed by EFM (Enhanced Fisher Model) is used to build the characteristic features, these latter are sent to the classification step using two methods, Distance Measurements and SVM (Support Vector Machine). Finally, the effect of fusion of two modalities (2D and 3D) is studied and examined. [6].

Existing standards which were developed for recognizing the face with masks on it do not work well due to the unique structure of the human faces. Face recognition is one of the latest technologies being studied area in biometric as it has wide area of applications. But Face detection is one of the challenging problems in Image processing. The basic aim of face detection is determining if there is any face in an image & then locates position of a face in an image. Evidently face detection is the first step towards creating an automated system which may involve other face processing. The neural network is created & trained with training set of faces & non-faces. All results are implemented in MAT

**CHAPTER - 4**

**PYTHON PROGRAMMING**

**2.1 CONTROL FLOW**

A program’s control flow is the order in which the program’s code executes.

The control flow of a Python program is regulated by conditional statements, loops, and function calls.

Python has three types of control structures:

• **Sequential** - default mode

• **Selection** - used for decisions and branching

• **Repetition** - used for looping, i.e., repeating a piece of code multiple times.

**1. Sequential**

Sequential statements are a set of statements whose execution process happens in a sequence. The problem

with sequential statements is that if the logic has broken in any one of the lines, then the complete source code

execution will break.

**2. Selection/Decision control statements**

In Python, the selection statements are also known as *Decision control statements* or *branching statements*.

The selection statement allows a program to test several conditions and execute instructions based on which

condition is true.

**2.2 OPERATORS**

An operator is a symbol that will perform mathematical operations on variables or

on values. Operators operate on operands (values) and return a result.

Python has 7 types of operators that you can use:

a. Arithmetic Operators

b. Relational Operators

c. Assignment Operators

d. Logical Operators

e. Membership Operators

f. Identity Operators

g. Bitwise Operators

a. **Arithmetic operators**

Python has seven arithmetic operators for different mathematical operations. They

are:

• + (Addition), - (Subtraction)

• \* (Multiplication), / (Division)

• \*\* (Exponentiation), // (Floor division)

• % (Modulus)

b. **Relational Operators**

Python has 6 relational operators:

▪ (Greater than)

• < (Less than)

• == (Equal to)

• != (Not equal to)

• >= (Greater than or equal to)

• <= (Less than or equal to)

**c. Assignment operators**

Python has 8 assignment operators:

• = (Assign)

• += (Add and assign)

• -= (Subtract and assign)

• \*= (Multiply and assign)

• /= (Divide and assign)

• %= (Modulus and assign)

• \*\*= (Exponentiation and assign)

• //= (Floor-divide and assign)

**d. Logical Operators**

They can combine conditions. Python has 3 logical operators:

• and (Logical and)

• or (Logical or)

• not (Logical not)

**e. Membership operators**

Membership operators check whether a value is in another. Python has 2

membership operators:

• in

• not in

**f. Identity operators**

Identity operators check whether two values are identical. Python has 2 identities

operators as well:

• Is

• is not

**g. Bitwise operators**

Python has 6 bitwise operators:

• & (Bitwise and)

• | (Bitwise or)

• ^ (Bitwise xor)

• ~ (Bitwise 1’s complement)

• << (Bitwise left-shift)

• >> (Bitwise right-shift)

LAB 2013 environment. [7].

**CHAPTER 5**

**PROPOSED METHODOLOGY**

We use Convolutional Neural Network and Deep Learning for Real Time Detection and Recognition of Human Faces, which is simple face detection and recognition system is proposed in this paper which has the capability to recognize human faces in single as well as multiple face images in a database in real time with masks on or off the face. Pre-processing of the proposed frame work includes noise removal and hole filling in color images. After pre-processing, face detection is performed by using CNNs architecture. Architecture layers of CNN are created using Kera’s Library in Python. Detected faces are augmented to make computation fast. By using Principal Analysis Component (PCA) features are extracted from the augmented image. For feature selection, we use Sobel Edge Detector.

# A. The Input Image

Real-time input images are used in this proposed system. Face of person in input images must be fully or partially covered as they have masks on it. The system requires a reasonable number of pixels and an acceptable amount of brightness for processing. Based on experimental evidence, it is supposed to perform well indoors as well as outdoors i.e. passport offices, hospitals, hotels, police stations and schools etc.

# B. The Pre-processing Stage

Input image dataset must be loaded as Python data structures for pre-processing to overturn the noise disturbances, enhance some relevant features, and for further analysis of the trained model. Input image needs to be pre-processed before face detection and matching techniques are applied. Thus pre-processing comprises noise removal, eye and mask detection, and hole filling techniques. Noise removal and hole filling help eliminate false detection of face/ faces. After the pre-processing, the face image is cropped and re-localized. Histogram Normalization is done to improve the quality of the preprocessed image.

# C. The Face Detection Stage

We perform face detection using HAAR Cascade algorithm. This system consists of the value of all black pixels in greyscale images was accumulated. They then deducted from the total number of white boxes. Finally, the outcome is compared to the given threshold, and if the criterion is met, the function considers it a hit. In general, for each computation in Har-feature, each single pixel in the feature areas can need to be obtained, and this step can be avoided by using integral images in which the value of each pixel is equal to the number of grey values above and left in the image.

Feature =𝛴i.e. {1.N} win. Rec Sum (x, yowch),

where RecSum (x, y, w,h) is the summation of intensity in any given upright or rotated rectangle enclosed in a detection window and x, yowch is for coordinates, dimensions, and rotation of that rectangle, respectively. Har Wavelets represented as box classifier which is used to extract face features by using integral image

# D. The Feature-Extraction Stage

Feature Extraction improves model accuracy by extracting features from pre-processed face images and translating them to a lower dimension without sacrificing image characteristics. This stage allows for the classification of human faces.

# E. The Classification Stage

Principal Component Analysis(PCA) is used to classify faces after an image recognition model has been trained to identify face images. Identifying variations in human faces is not always apparent, but PCA comes into the picture and proves to be the ideal procedure for dealing with the problem of face recognition. PCA does not operate classifying face images based on geometrical attributes, but rather checks which all factors would influence the faces in an image. PCA was widely used in the field of pattern recognition for classification problems. CA demonstrates its strength in terms of data reduction and perception.

# F. Training Stage

The method is based on the notion that it learns from preprocessed face images and utilizes CNN model to construct a framework to classify images based on which group it belongs to. This qualified model is saved and used in the prediction section later. In CNN model, the stages of feature extraction are done by PCA and feature selection done by Sobel Edge Detector and thus it improves classification efficiency and accuracy of the training model.

# G. Prediction Stage

In this stage, the saved model automatically detects theoftheface maskimagecaptured by the webcam or camera. The saved model and the pre-processed images are loaded for predicting the person behind the mask. CNN offers high accuracy over face detection, classification and recognition produces precise and exactresults.CNN model follows a sequential model along with Kera’s Library in Python for prediction of human faces.

**CHAPTER 6**

**MODULES**

The proposed system contains the following modules:

A. Pre-processing Images

B. Capture image ( )

C. Upload image ()

D. Classifier(image)

E. Prediction(image)

# A. Pre-processing Images

The input image is captured from a webcam or camera in real-time world. The frames (images)from the dataset are loaded. Face images are cropped and resized after they have been loaded. Later, noise distortions in the images are suppressed. Normalization is then done to normalize the images from 0-255 to 0-1 range.

# B. Capture image ()

In this Module we are able to capture real time images. We do this by the help of Flutter and applying in to the Classifier Model. Input: Nothing

# C. Upload image ()

Here we can browse the image and upload for finding the Plant disease. We need to fetch the image. And this image passes to Classifier Module. Input: Nothing Output: Image

# D. Classifier(image)

Following data Prepossessing of the images, will apply to the Classifier. Here it will find out the feature of the images. Mainly in this module feature extraction occurs. Image similarity features will be stored in to the model which gets created.

Input: Image

Output: Model

# E. Prediction(image)

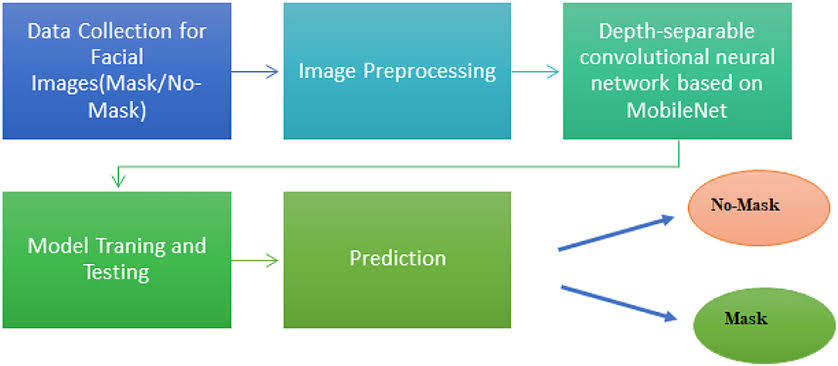
In this Module prediction of person take place. Here the browsed image will be placed in to the model and output will be shown as based on which label its get matched the most.

Input: Image

Output: Predicted Label.

**CHAPTER 7**

**SYSTEM ARCHITECTURE**



# A. User

User refers to person standing in front of a webcam or camera in a real world scenario.

# B. Capture Images

The webcam or camera captures images which are then used as dataset to train the model. If the dataset captures human faces in different masks and in different backgrounds along with large number of human face images, then the accuracy of the training model increases.

# C. Face Detection

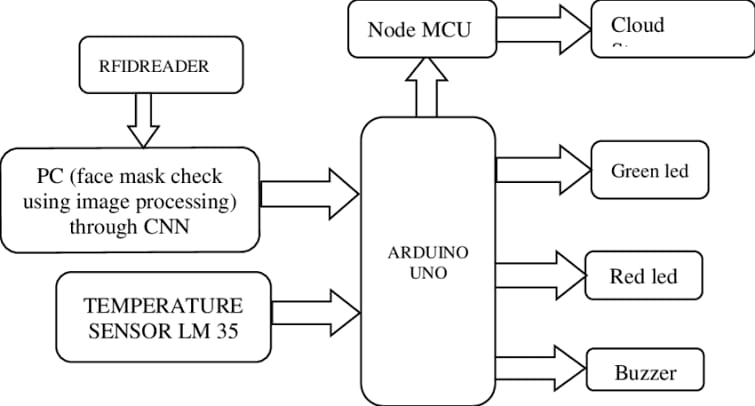
For face detection, we use HAAR Cascade algorithm. In this method all black pixels in greyscale images was accumulated. They then deducted from the total number of white boxes. Finally, the outcome is compared to the given threshold, and if the criterion is met, the function considers it a hit.

# D. Mask Detection

For Mask Detection, we use a sequential CNN model along with inbuilt Kera’s Library in Python. The sequential CNN model is trained from dataset of human faces with or without masks on the faces. It forms a logic from the pre-processed images like a human brain, then the model detects the face along with mask using feature extraction and feature selection. After identification of the mask along with face of the person, it forwards to the prediction or identification stage.

# E. Person Identification

In this stage, the trained model predicts the face of the person behind the mask according to the trained model. The prediction is based on the number of images trained by the model and its accuracy. Finally, the system displays that the person name along with the indication of he or she wearing a mask or not.



# F. Dataset

The proposed model has datasets captured from individual’s person. The dataset of faces is classified into with masks and without masks and is stored in different databases. Each folder consists 40 to 60 images of an individual person respectively. The individual’s person face images should have images captured from different masks and different backgrounds so the accuracy of training model increases The dataset is integrated with Kera’s Library in Python. Larger the dataset more accurate the training model. So dataset images are directly congruent to accuracy of the training model.

# G. Data Pre-Processing

This module is used for read image. After reading we resize the image if needed me rotate the image and also remove the noises in the image. Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise. Later normalization is done to clean the images and to change the intensity values to pixel format. The output of this stage is given to training model

Input: image

Output: pixel format

# H. Segmentation

Segment the image, separating the background from foreground objects and we are going to further improve our segmentation with more noise removal. We separate different objects in the image with markers.

Input: pixel format

Output: image

# I. Edge detection

Sobel edge detector is using. It is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. 2-D spatial gradient measurement on the image is performed by Sobel operator. Each pixel of the image is operated by Sobel operator and measured the gradient of the image for each pixel. Pair of 3×3 convolution masks is used by Sobel operator, one is for x direction and other is for y Direction. The Sobel edge enhancement filter has the advantage of providing differentiating (which gives the edge response) and smoothing (which reduces noise) concurrently.

Input: image

Output: image

# J. Localization

Find where the object is and draw a bounding box around it.

Input: image

Output: localized image

# K. Feature Selection

The biggest advantage of Deep Learning is that we do not need to manually extract features from the image. The network learns to extract features while training. You just feed the image to the network (pixel values). What you need is to define the Convolutional Neural Network architecture and a labelled dataset. Principal Component Analysis (PCA) is a useful tool for doing this. PCA checks all the factors influencing the faces rather just checking its geometrical factors. Thus using PCA gives accurate and precise detection and recognition result of faces.

Input: image pixel format

Output: labels

# L. CNN Architecture creation

A sequential CNN model is designed specifically for analyzing the human faces with mask on it or not. The Convolutional Neural Network Architecture layers will be created using the Kera’s library in Python. The convolutional layer is used for mask detection. Itextractsthefeaturesoffaceimages using Principal Component Analysis(PCA)and converts them into a lower dimension without losing the image characteristics. The output of the convolutional layer willbetheinputofthenextBatchNormalizationlayer. The Batch Normalization layer standardizes the inputs to a layer for each mini-batch. This has the effect of stabilizing the learning process and dramatically reducing the number of training epochs. After this, the face images undergo classification. If the images are tested, then model accuracy calculations and predications takes place. If non-test images come, then first the images are trained along with it its validation testing is also done. If it is validating, then the model is trained and saved for further calculations. Otherwise, if it is nonvalid ate, then it undergoes network training and calculations are done for losing weights and are adjusted accordingly. Finally, the CNN model gives accuracy and prediction of the human face behind the mask.

# M. Training

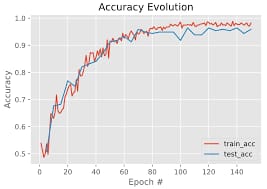
The pre-processed face images are directed to the CNN model for training. Based on the dataset given, a logic is formed in the CNN to categorize the faces according to their features. This trained model is saved. The trained model is capable of categorizing human faces based on with or without masks on it. Training model is done with the help of a sequential CNN model and HAAR Cascade Algorithm.

# N. Predication

In this phase, when a person comes in front on a webcam, the image is captured and predicted by the CNN model according to the logic learned by the sequential model. The image undergoes pre-processing. This pre-processed images and the saved CNN model are then loaded. Based on the algorithm interpreted by the system predicts and detects the human faces according to trained model

# RESULT

This proposed work uses a sequential Convolutional Neural Network for detecting and recognizing human faces of individuals with mask or without it. CNN model and Har Cascade Algorithm facilitates automatic detection and recognition of human face which overcome the noise variations and background variations caused by the surrounding and provide more accurate and precise result. It also helps to overcome the uneven nature of the current trend of face recognition and detection. From the experiments it is clear that the proposed CNN achieves a high accuracy when compared to other architectures. The proposed algorithm works effectively for different types of images. These results suggest that the proposed CNN model reduces complexity and make method computationally effective. The proposed system works well effectively for grayscale as well as for the color image with masks on it or without masks on it.



# CONCLUSION

Our proposed system can detect and recognize human face(s) in real-time world. Compared to the traditional face detection and recognition system, the face detection and recognition based on CNN model along with the use of Python libraries has shorter detection and recognition time and stronger robustness, which can reduce the miss rate and error rate. It can still guarantee a high test rate in a sophisticated atmosphere, and the speed of detection can meet the real time requirement, and achieve good effect. The proposed CNN model shows greater accuracy and prediction for detecting and recognizing human faces. The results show us that the current technology for face detection and recognition is compromised and can be replaced with this proposed work. Therefore, the proposed method works very well in the applications of biometrics and surveillance.

**PROJECT**

**FACE MASK DETECTION**

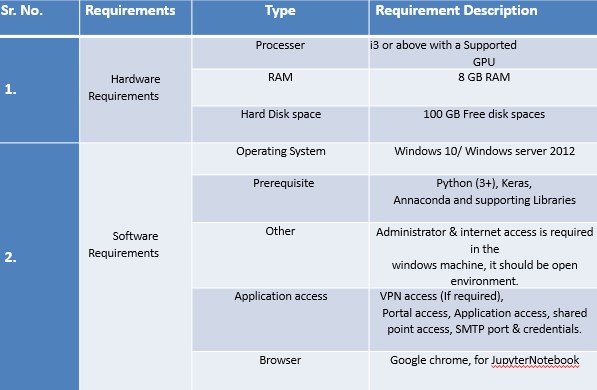
**INTRODUCTION:**

* To make machines more intelligent, the developers are diving into machine learning and deep learning techniques. A human learns to perform a task by practicing and repeating it again and again so that it memorizes how to perform the tasks. Then the neurons in his brain automatically trigger and they can quickly perform the task they have learned. Deep learning is also very similar to this. It uses different types of neural network architectures for different types of problems. For example – object recognition, image and sound classification, object detection, image segmentation, etc.
* In this python project, we implemented a CNN to detect a person or a human being wear a mask and check the accuracy of wearing a mask.

**MAIN FEATURES:**

* + Detect faces
  + Classify into the person with a mask and without a mask
  + Check the accuracy of wearing mask
  + Put the results on the live video and display it.

**HARDWARE AND SOFTWARE REQUIREMENTS:**



**TECHNOLOGY AND DATASET USED:**

* Technology: Deep learning is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision-making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabelled. Also known as deep neural learning or deep neural network.

* Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing. However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information.

**DATASET:**

* For this python project, we’ll use the Adience dataset; the dataset is available in the public domain. This dataset serves as a benchmark for face photos and is inclusive of various real-world imaging conditions like noise, lighting, pose, and appearance. The images have been collected from Flickr albums and distributed under the Creative Commons (CC) license. It has a total of 26,580 photos of 2,284 subjects in eight mask ranges (as mentioned above) and is about 1GB in size. The models we will use have been trained on this dataset.

**LIBRARY/MODULES USED:**

**Argparse:-**

* The argparse module makes it easy to write user-friendly command-line interfaces. The program defines what arguments it requires, and argparse will figure out how to parse those out of sys.argv. The argparse module also automatically generates help and usage messages and issues errors when users give the program invalid arguments.
* **Numpy:**
* NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.
* Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.
* **Operations using NumPy:**
* Using NumPy, a developer can perform the following operations −
  + Mathematical and logical operations on arrays.
  + Fourier transforms and routines for shape manipulation.
  + Operations related to linear algebra.
  + NumPy has in-built functions for linear algebra and random number generation.
* **Matplot:**
* Matplot library is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc. It is used along with NumPy to provide an environment that is an effective open source alternative for MatLab. It can also be used with graphics toolkits like PyQt and wxPython.
* **Imutils:**
* Before we continue to the code we need install imutils.  
  Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and both Python 2.7 and Python 3.
* For installing:
* Open your Command Prompt and install it via:

pip install imutils

* **Pandas**: Pandas is an opensource Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like Active State’s ActivePython.
* **tensorflow/keras:**
* **KERAS:**
* Like TensorFlow, Keras is an open-source, ML library that’s written in Python. The biggest difference, however, is that Keras wraps around the functionalities of other ML and DL libraries, including TensorFlow, Theano, and CNTK. Because of TF’s popularity, Keras is closely tied to that library.
* Many users and data scientists, us included, like using Keras because it makes TensorFlow much easier to navigate—which means you’re far less prone to make models that offer the wrong conclusions.
* Keras builds and trains neural networks, but it is user friendly and modular, so you can experiment more easily with deep neural networks. Keras is a great option for anything from fast prototyping to state-of-the-art research to production. The key advantages of using Keras, particularly over TensorFlow, include:
* **Ease of use.** The simple, consistent UX in Keras is optimized for use cases, so you get clear, actionable feedback for most errors.
* **Modular composition.** Keras models connect configurable building blocks, with few restrictions.
* **Highly flexible and extendable.** You can write custom blocks for new research and create new layers, loss functions, metrics, and whole models.
* So here, we use Keras because it offers something unique in machine learning i.e single API that works across several ML frameworks to make that work easier.

**WORKFLOW OF PROJECT:**

* **1. Import the libraries and load the dataset:**
* First, we are going to import all the modules that we are going to need for training our model. The Keras library already contains some datasets and MNIST is one of them. So we can easily import the dataset and start working with it. The mnist.load\_data() method returns us the training data, its labels and also the testing data and its labels.
* **2. Preprocess the data**
* The image data cannot be fed directly into the model so we need to perform some operations and process the data to make it ready for our neural network.
* This dataset consists of 4095 images belonging to two classes:
* With \_mask: 2165 images
* Without \_mask: 1930 images
* **3. Create the model**
* In the part we’ll learn about face mask detection, including the steps required to automatically predict the mask wearing by a person from an image or avideo stream (and why face mask detection is best treated as a classification problem rather than a regression problem).
* From there, we’ll discuss our deep learning-based mask detection model and then learn how to use the model for both:
* Face mask detection in static images
* Face mask detection in real-time video streams
* **4. Train the model**
* Once your face detector has produced the bounding box coordinates of the face in the image/video stream, you can move on to Stage #2 — identifying the person who were masks or who is not**.**
* **5. Evaluate the model**
* Mask detection is the process of automatically discerning the wearing mask of a person solely from a photo of their face.
* Typically, you’ll see face mask detection implemented as a two-stage process:
* Stage #1: Detect faces in the input image/video stream
* Stage #2: Display on the screen the person is wearing the mask or not.
* **6. Running real time embedded system**
* For Stage #1, any face detector capable of producing bounding boxes for faces in an image can be used, including but not limited to Haar cascades, HOG + Linear SVM, Single Shot Detectors (SSDs), etc.

**CONCLUSION:**

* This project presents a system for a smart city to reduce the spread of coronavirus by informing the authority about the person who is not wearing a facial mask that is a precautionary measure of COVID-19. The motive of the work comes from the people disobeying the rules that are mandatory to stop the spread of coronavirus. The system contains a face mask detection architecture where a deep learning algorithm is used to detect the mask on the face. To train the model, labeled image data are used where the images were facial images with masks and without a mask. The proposed system detects a face mask with an accuracy of 98.7%. The decision of the classification network is transferred to the corresponding authority. The system proposed in this study will act as a valuable tool to strictly impose the use of a facial mask in public places for all people.

# REFERENCES

[1] Zheng Jun, Hua Rizhao, Tang Zheng an, Wang Feng “Face detection based on LBP",2017 IEEE 13th International Conference on Electronic Measurement & Instruments.

[2] Q. B. Sun, W. M. Huang, and J. K. Wu "Face Detection Based on Color and Local Symmetry Information”, National University of Singapore Heng Mui King Terrace, Kent Ridge Singapore.

[3] Based on Color and Local Symmetry Information”, National University of Singapore Heng Mui King Terrace, Kent Ridge Singapore

[4] Wang Yang, Zheng Jinchuan "Real-time face detection based on YOLO",1st IEEE International Conference on Knowledge Innovation and Invention 2018

[5] Dr. P. Shanmugavadivu, Ashish Kumar, "Rapid Face Detection and Annotation with Loosely Face Geometry",2016 2nd International Conference on Contemporary Computing and Informatics (ic3i)

[6] T. F. Coots, G. J. Edwards, and C. J. Taylor, "Active appearance models," IEEE Transactions on pattern analysis and machine intelligence, vol. 23, pp. 681-685, 2001.

[7] S. Sapidity and S. Ramavath, "Real-Time Multiple Face Recognition using Deep Learning on Embedded GPU System," 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, 2018, pp. 1318-1324

[8] S. Ren, K. He, R. Kirchick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.

[9] Matthew D Zeigler, Rob Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014: Computer Vision – ECCV 2014 pp 818-833.

[10] H. Jiang and E. Learned-Miller, "Face Detection with the Faster R-CNN," 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), Washington, DC, 2017, pp. 650-657.