**A**

**MINI PROJECT REPORT**

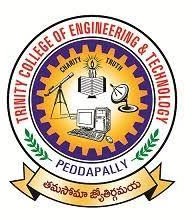
**ON**

**“PREDICTING IDENTICAL TWINS USING ML”**

**In partial fulfillment for the award of the degree Of**

**BACHELOR OF TECHNOLOGY**

**IN**

**ARTIFICIAL INTELLIGENCE & MACHINE LEARNING**

**SUBMITTED BY**

**A.SAI TEJA (22UD5A7301)**

**P.SAI CHARAN (22UD5A7305)**

**S.Y.TAHEER (22UD5A7306)**

Under the Esteemed Guidance of

**Mrs. G. LAKSHMI**

**Associate Professor**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**TRINITY COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(Approved by AICTE, New Delhi, Affiliated to JNTUH, Hyderabad)**

**PEDDAPALLI-505172**

**2022-2025**

**COLLEGE CODE: UD**

**TRINITY COLLEGE OF ENGINEERING AND TECHNOLOGY**

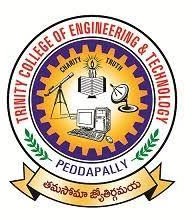
**(Approved by AICTE, New Delhi, Affiliated to JNTUH, Hyderabad)**

**PEDDAPALLI-505172**

**2022-2025**

**Date:**

**CERTIFICATE**



Certified that this project report entitled, **“PREDICTING IDENTICAL TWINS USING ML”** is the Bonafide work of **A.SAITEJA(22UD5A7301), P.SAICHARAN(22UD5A7305), S.Y.TAHEER (22UD5A7306) of IV Year, AI&ML** in the year 2024 in partial fulfillment of the requirements to award the Degree of Bachelor of Technology in **ARTIFICIAL INTELLIGENCE & MACHINE LEARNING** branch of Trinity College of Engineering and Technology.

**Internal Guide Head of the Department**

**External Examiner Principal**

**ACKNOWLEDGEMENT**

We wish to acknowledge with gratitude to the College Authority for the opportunity and support given me to do the project without which I could not have made this project successful.

We wish to place on my record my deep sense of gratitude to my project guide, **Mrs**. **G. LAKSHMI,** Associate Professor of AIML for her constant motivation and valuable help throughout the project work.

We highly obliged to **Mrs.J.SWATHI, Associate Professor & Head of the Department of CSE** for her motivation and encouragement during the period of this project.

We feel privileged to thank to Academic Director **Dr.V.ASHOK KUMAR** for his encouragement during the progress of the work.

We highly indebted to our Principal, **Dr. MANI GANESH** support using the period of this project work.

We express my utmost thanks to **Sri. D.PRASHANTH REDDY Chairman,** Trinity Group of Institutions**.**

We express my utmost thanks to **Sri.D.MANOHAR REDDY** Founder of Trinity Group of Institutions, for his continuous care towards my achievements.

We would like to thank the teaching and non-teaching staff of AIML Department for sharing their knowledge with us.

Finally, special thanks to our parents, sisters and brothers for their support and encouragement throughout our life and this course. Thanks to all our friends and well- wishers for their constant support.

**DECLARATION**

We hereby declare that the work which is being presented in this report entitled “PREDICTING IDENTICAL TWINS USING ML” submitted towards the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Artificial Intelligence & Machine Learning, Trinity College of Engineering and Technology, Peddapally is an authentic record of my own work carried out under the supervision of **Mrs. G. LAKSHMI**, **Associate Professor**, Department of AIML Trinity College of Engineering and Technology, Peddapalli. To the best of my knowledge and belief, this project bears no resemblance with any report submitted to Trinity College of Engineering and Technology or any other University for the award of any degree.

**A. SAI TEJA (22UD5A7301**)

**P. SAI CHARAN (22UD5A7305)**

**S.Y. TAHEER (22UD5A7306)**

**CONTENTS**

**Abstract I**

**List of figures II**

**List of screenshots III**

**CHAPTER-1 INTRODUCTION 1-4**

1.1: Existing system 1-2

1.2: Proposed System 3

1.3: Input and output design 4

**CHAPTER-2 LITERATURE OF SURVEY 5**

**CHAPTER-3 ANALYSIS 6-8**

3.1: Software Requirements 6

3.2: Hardware Requirements 6

3.3: Feasibility Study 7

3.3.1: Economic Feasibility 7

3.3.2: Technical Feasibility 7

3.3.3: Social Feasibility 8

**CHAPTER-4 DESIGN 9-13**

4.1: UML Diagrams 9-10

4.2: Use case Diagram 10

4.3: Class Diagram 11

4.4: Sequence Diagram 12

4.5: Activity Diagram 13

**CHAPTER-5 IMPLEMENTATION 14-50**

5.1: Source Code 14-20

5.2: Screen Shots 21-28

5.3: Python 28-50

**CHAPTER-6 TESTING 51-54**

6.1: Unit Testing 51

6.2: Integration Testing 51

6.3: functional Testing 52

6.4: System Testing 52

6.5: White Box Testing 52

6.6: Black Box Testing 53-54

**CHAPTER-7: FUTURE ENHANCEMENT 55-56**

**CHAPTER-8: TEST CASES 57**

**CHAPTER-9: CONCLUSION 58**

**CHAPTER-10: REFERENCE 59**

**CHAPTER -11: BASE PAPER 60-62**

# ABSTRACT

The "Prediction of Identical Twins using ML" project seeks to develop a machine learning-based solution for accurately identifying identical twins from a population based on their physical characteristics and biometric data. Identifying identical twins is a complex task with significant implications in various fields, including healthcare, forensics, and even personalized marketing. Leveraging machine learning, this project aims to advance the state-of-the-art in twin identification, offering a novel and efficient approach to a longstanding challenge.

**LIST OF FIGURES:**

|  |  |  |
| --- | --- | --- |
| **SI.NO** | **NAME OF THE FIGURE** | **PAGE**  **NUMBER** |
| 1. | Existing System | 1 |
| 2. | Use case diagram | 10 |
| 3. | Class diagram | 11 |
| 4. | Sequence diagram | 12 |
| 5. | Activity diagram | 13 |

**LIST OF SCREENSHOTS:**

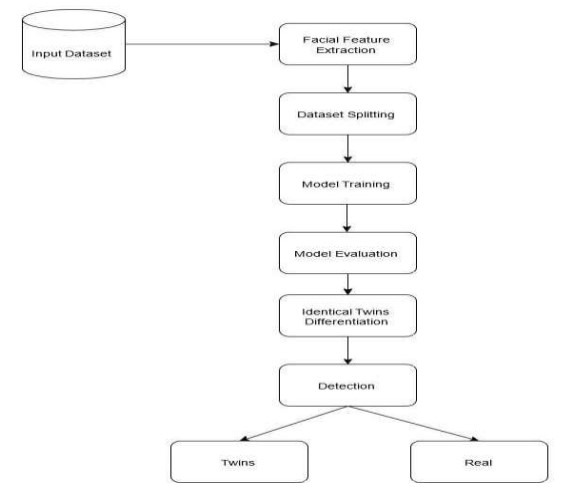
|  |  |  |
| --- | --- | --- |
| **SI NO.** | **NAME OF THE**  **SCREENS** | **PAGE NUMBER** |
| 1 | Dataset review | 21 |
| 2 | Dataset folder | 22 |
| 3 | Run the main program | 23 |
| 4 | Loading dataset | 23 |
| 5 | Dataset Pre-Processing | 24 |
| 6 | Run Naïve | 24 |
| 7 | Naïve bayes Accuracy | 25 |
| 8 | Run Random Forest | 26 |
| 9 | Random Forest Accuracy | 26 |
| 10 | Comparison | 27 |
| 11 | Testing Dataset Preview | 27 |
| 12 | Result 1 for 1st activity | 27 |
| 13 | Result 2 for 2nd activity | 28 |
| 14 | Accuracy | 28 |
| 15. | Install python | 44 |
| 16. | Download python | 45 |
| 17. | Select python version | 45 |
| 18. | Select the OS | 46 |
| 19. | Open downloaded python version | 47 |
| 20. | Add python to path | 47 |
| 21. | Click on install | 48 |
| 22. | Open CMD | 48 |
| 23. | Check python installed | 49 |

# CHAPTER – 1

# INTRODUCTION

Identical twins, also known as monozygotic twins, share 100% of their genetic material and often bear a striking physical resemblance to each other. Despite their genetic similarity, there is a lack of automated and accurate methods to identify identical twins based on their unique physical characteristics. Existing identification methods rely on subjective observations or biometric data, such as fingerprinting or facial recognition, which can yield unreliable results. This project aims to tackle this issue by harnessing the power of machine learning to create a more reliable and objective means of identifying identical twins.

## EXISTING SYSTEM



**FIG-1: EXISTING SYSTEM**

Identifying identical twins accurately is a challenging problem with several implications:

1. Forensics: In criminal investigations, the inability to distinguish between identical twins can lead to wrongful accusations or the release of a guilty party. Current forensic methods often rely on non-genetic traits that may not be unique to each twin.
2. Healthcare: In medical contexts, misidentification of twins can lead to incorrect treatments or prescriptions. This is particularly relevant in cases of organ transplantation, where genetic compatibility is crucial.
3. Education and Research: Identical twins can provide valuable insights into the role of genetics in various traits and diseases. Accurate twin identification is vital for studies in behavioral genetics, epidemiology, and related fields.

**DISADVANTAGES OF EXISTING SYSTEM:**

* Low Efficiency
* Low accuracy

## PROPOSED SYSTEM:

The "Prediction of Identical Twins using ML" project aims to address the challenges of twin identification through the following proposed solution:

**1**. **Data analysis**: A comprehensive dataset of identical twins, including their physical characteristics, biometric data, and genetic information, will be collected. This dataset will serve as the foundation for training the machine learning model.

**2. Machine Learning Model Development:** State-of-the-art machine learning techniques, such as deep learning and feature extraction, will be employed to develop a model capable of identifying identical twins based on a combination of physical traits and biometric data. The model will be trained on the collected dataset to learn unique patterns that distinguish identical twins from other individuals.

**3.** **Deployment and Integration:**: The trained model will be integrated into a user-friendly application that can be used in various domains, including forensics, healthcare, and research. This application will provide reliable and objective twin identification, helping to solve the problem of misidentification.

**ADVANTAGES OF PROPOSED SYSTEM:**

* High Efficiency
* High Accuracy

## INPUT & OUTPUT DESIGN:

**INPUT DESIGN:**

1.Data Sources: Genetic data (e.g., DNA sequences), demographic data (e.g., age, sex), and medical history data (e.g., diseases, medications).

2. Input Features:

1. Genetic Features: SNPs (Single Nucleotide Polymorphisms), indels (insertions/deletions), and other genetic variations.

2. Demographic Features: Age, sex, ethnicity, and other relevant demographic characteristics.

3. Medical History Features: Presence of specific diseases, medications, and other relevant medical history.

3. Data Format: CSV, JSON, or other suitable formats for machine learning algorithms.

4. Data Preprocessing:

1. Handling missing values

2. Normalizing/scale data

3. Encoding categorical variables

**OUTPUT DESIGN:**

1. Prediction Output: A binary classification output indicating whether the twins are identical (1) or not (0).

2. Probability Output: A probability score indicating the likelihood of the twins being identical.

3. Classification Metrics: Accuracy, precision, recall, F1-score, and other relevant metrics to evaluate the performance of the model.

4. Output Format: CSV, JSON, or other suitable formats for easy interpretation and further analysis.

**CHAPTER -2**

## LITERATURE REVIEW :

### **Literature Review: Biometric Data and Twin Recognition**

This literature review explores the intersection of biometric data and twin recognition in the context of machine learning. It delves into the various biometric modalities, such as facial recognition, fingerprinting, and iris scans, and their potential utility in identifying identical twins. The review also investigates existing challenges and limitations in utilizing biometric data for twin recognition and provides insights into how machine learning can enhance the accuracy and reliability of twin identification.

### **Literature Review: Twin Studies and Genetics in Identical Twin Identif ication**

This literature review focuses on the genetic underpinnings of identical twin identification. It examines twin studies, genetic markers, and the heritability of physical traits in monozygotic twins. By analysing the existing research, this review offers a comprehensive overview of the unique genetic factors that can be leveraged for accurate twin recognition, particularly through machine learning approaches.

### **Literature Review: Ethics and Privacy in Identical Twin Identification Using ML**

This literature review explores the ethical and privacy considerations associated with employing machine learning for identical twin identification. It investigates the potential risks of biometric data collection and the implications for individuals' privacy and consent. The review examines current ethical guidelines, legal frameworks, and best practices in twin identification to ensure that machine learning models are developed and applied responsibly and with utmost regard for ethical standards.

# CHAPTER -3

**ANALYSIS**

* 1. **HARDWARE REQUIREMENTS**
* **System    :** i3 or above.
* **Ram    :** 4 GB.
* **Hard Disk :** 40 GB
  1. **SOFTWARE REQUIREMENTS**



* **Operating system : Windows 8 Ultimate.**
* **Coding Language : Python.**

## FEASIBILITY STUDY :

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

**Three key considerations involved in the feasibility analysis are :**

* **ECONOMICAL FEASIBILITY**
* **TECHNICAL FEASIBILITY**
* **SOCIAL FEASIBILITY**

**3.3.1 ECONOMICAL FEASIBILITY**



This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **3.3.2 TECHNICAL FEASOBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### **SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

# CHAPTER-4

## 4.1 SYSTEM DESIGN

### **Modules :**

* User
  + System

### **User**

In this module,user has to give the retina image as input.

### **System**

In this module,system has to preprocess the image,Run the CNN algorithm,predict retinopathy and show the accuracy.

## DESIGN REPRESENTATION :

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

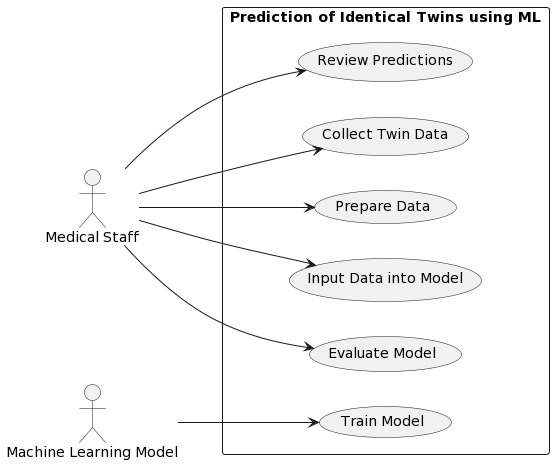
The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS :**

**The Primary goals in the design of the UML are as follows :**

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of object oriented tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.

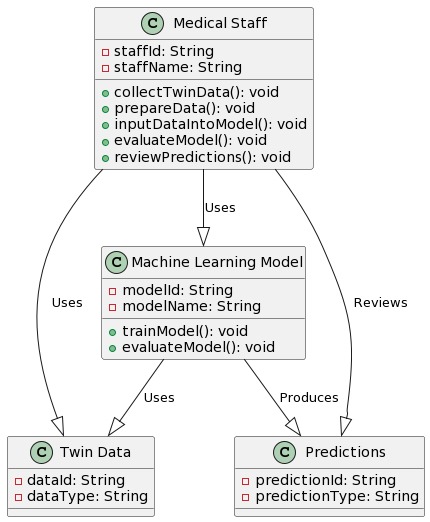
### **USE CASE DIAGRAM :**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

**FIGURE 4.2: USE CASE DIAGRAM**

### **CLASS DIAGRAM :**

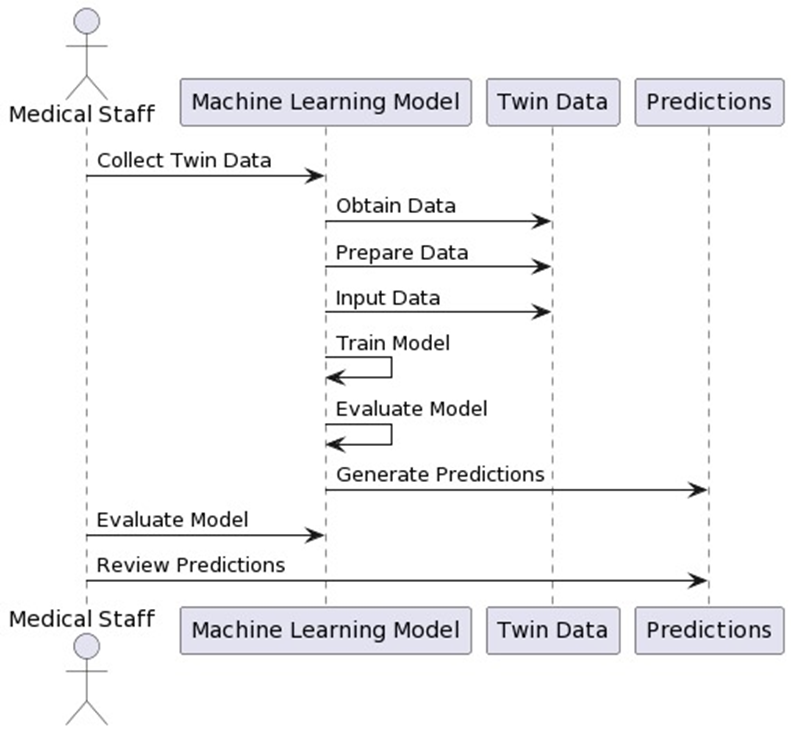
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.



**FIGURE 4.3: CLASS DIAGRAM**

**4.4 SEQUENCE DIAGRAM**

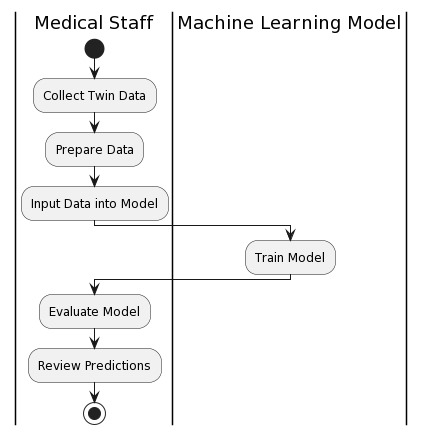
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**FIGURE 4.4: SEQUENCE DIAGRAM**

**4.6 ACTIVITY DIAGRAM**

Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

****

**FIGURE 4.5: ACTIVITY DIAGRAM**

# CHAPTER-5

**5.1 SOURCE CODE**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

import matplotlib.pyplot as plt

import numpy as np

from tkinter import simpledialog

from tkinter import filedialog

import os

import cv2

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score

from sklearn.preprocessing import StandardScaler

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

import seaborn as sns

import pandas as pd

main = tkinter.Tk()

main.title("Prediction of Identical Twins using ML") #designing main screen

main.geometry("1300x1200")

global filename

global X, Y

global X\_train, X\_test, y\_train, y\_test

global accuracy, precision, recall, fscore, labels, rf

global scaler

labels = ['Real', 'Twins']

def getID(name):

index = 0

for i in range(len(labels)):

if labels[i] == name:

index = i

break

return index

def uploadDataset():

global filename

global X, Y

filename = filedialog.askdirectory(initialdir=".")

text.delete('1.0', END)

text.insert(END,filename+" loaded\n")

if os.path.exists("model/X.txt.npy"):

X = np.load('model/X.txt.npy')

Y = np.load('model/Y.txt.npy')

else:

for root, dirs, directory in os.walk(filename):

for j in range(len(directory)):

name = os.path.basename(root)

if 'Thumbs.db' not in directory[j]:

img = cv2.imread(root+"/"+directory[j])

bilateral\_filter = cv2.bilateralFilter(img,15,80,80)

bilateral\_filter = cv2.cvtColor(bilateral\_filter, cv2.COLOR\_BGR2GRAY)

clahe = cv2.createCLAHE(clipLimit = 2, tileGridSize = (8, 8))

bilateral\_filter = clahe.apply(bilateral\_filter)

detected\_image = cv2.Canny(bilateral\_filter,50,150)

img = cv2.resize(detected\_image, (32, 32))

X.append(img.ravel())

label = getID(name)

Y.append(label)

X = np.asarray(X)

Y = np.asarray(Y)

np.save('model/X.txt',X)

np.save('model/Y.txt',Y)

text.insert(END,"Labels in Dataset : "+str(labels)+"\n")

text.insert(END,"Total Real & Twins Images found in dataset : "+str(X.shape[0]))

def DatasetPreprocessing():

text.delete('1.0', END)

global X, Y

global X\_train, X\_test, y\_train, y\_test, scaler

X = X.astype('float32')

scaler = StandardScaler()

X = scaler.fit\_transform(X)

indices = np.arange(X.shape[0])

np.random.shuffle(indices)

X = X[indices]

Y = Y[indices]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.5) #split dataset into train and test

text.insert(END,"Dataset Normalization & Preprocessing Task Completed\n\n")

text.insert(END,"Dataset Train & Test Splits\n")

text.insert(END,"Total images found in dataset : "+str(X.shape[0])+"\n")

text.insert(END,"80% dataset used for training : "+str(X\_train.shape[0])+"\n")

text.insert(END,"20% dataset user for testing : "+str(X\_test.shape[0])+"\n")

def calculateMetrics(algorithm, testY, predict):

global labels

p = precision\_score(testY, predict,average='macro') \* 100

r = recall\_score(testY, predict,average='macro') \* 100

f = f1\_score(testY, predict,average='macro') \* 100

a = accuracy\_score(testY,predict)\*100

accuracy.append(a)

precision.append(p)

recall.append(r)

fscore.append(f)

text.insert(END,algorithm+" Accuracy : "+str(a)+"\n")

text.insert(END,algorithm+" Precision : "+str(p)+"\n")

text.insert(END,algorithm+" Recall : "+str(r)+"\n")

text.insert(END,algorithm+" FSCORE : "+str(f)+"\n\n")

conf\_matrix = confusion\_matrix(testY, predict)

ax = sns.heatmap(conf\_matrix, xticklabels = labels, yticklabels = labels, annot = True, cmap="viridis" ,fmt ="g");

ax.set\_ylim([0,len(labels)])

plt.title(algorithm+" Confusion matrix")

plt.ylabel('True class')

plt.xlabel('Predicted class')

plt.show()

def runNaiveBayes():

text.delete('1.0', END)

global accuracy, precision, recall, fscore, cnn\_model

global X\_train, y\_train, X\_test, y\_test

accuracy = []

precision = []

recall = []

fscore = []

nb = GaussianNB()

nb.fit(X\_train, y\_train)

predict = nb.predict(X\_test)

calculateMetrics("Naive Bayes", y\_test, predict)

def runRandomForest():

global rf

global X\_train, y\_train, X\_test, y\_test

rf = RandomForestClassifier()

rf.fit(X\_train, y\_train)

predict = rf.predict(X\_test)

calculateMetrics("Random Forest", y\_test, predict)

def graph():

df = pd.DataFrame([['Naive Bayes','Accuracy',accuracy[0]],['Naive Bayes','Precision',precision[0]],['Naive Bayes','Recall',recall[0]],['Naive Bayes','FSCORE',fscore[0]],

['Random Forest','Accuracy',accuracy[1]],['Random Forest','Precision',precision[1]],['Random Forest','Recall',recall[1]],['Random Forest','FSCORE',fscore[1]],

],columns=['Algorithms','Accuracy','Value'])

df.pivot("Algorithms", "Accuracy", "Value").plot(kind='bar')

plt.title("All Algorithm Comparison Graph")

plt.show()

def predict():

global rf, scaler

filename = filedialog.askopenfilename(initialdir="testImages")

img = cv2.imread(filename)

bilateral\_filter = cv2.bilateralFilter(img,15,80,80)

bilateral\_filter = cv2.cvtColor(bilateral\_filter, cv2.COLOR\_BGR2GRAY)

clahe = cv2.createCLAHE(clipLimit = 2, tileGridSize = (8, 8))

bilateral\_filter = clahe.apply(bilateral\_filter)

detected\_image = cv2.Canny(bilateral\_filter,50,150)

image = cv2.resize(detected\_image, (32, 32))

X = []

X.append(image.ravel())

X = np.asarray(X)

X = X.astype('float32')

X = scaler.transform(X)

predict = rf.predict(X)[0]

img = cv2.imread(filename)

img = cv2.resize(img, (700,400))

cv2.putText(img, 'Predicted As : '+labels[predict], (10, 25), cv2.FONT\_HERSHEY\_SIMPLEX,0.7, (0, 0, 255), 2)

cv2.imshow('Predicted As : '+labels[predict], img)

cv2.imshow("Detected Object", detected\_image)

cv2.waitKey(0)

font = ('times', 16, 'bold')

title = Label(main, text='Prediction of Identical Twins using ML')

title.config(bg='LightGoldenrod1', fg='medium orchid')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 12, 'bold')

text=Text(main,height=22,width=140)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=200)

text.config(font=font1)

font1 = ('times', 12, 'bold')

uploadButton = Button(main, text="Upload Twins Dataset", command=uploadDataset)

uploadButton.place(x=50,y=100)

uploadButton.config(font=font1)

preButton = Button(main, text="Dataset Preprocessing", command=DatasetPreprocessing)

preButton.place(x=300,y=100)

preButton.config(font=font1)

nbButton = Button(main, text="Run Naive Bayes Algorithm", command=runNaiveBayes)

nbButton.place(x=510,y=100)

nbButton.config(font=font1)

rfButton = Button(main, text="Run Random Forest Algorithm", command=runRandomForest)

rfButton.place(x=740,y=100)

rfButton.config(font=font1)

graphButton = Button(main, text="Comparison Graph", command=graph)

graphButton.place(x=50,y=150)

graphButton.config(font=font1)

predictButton = Button(main, text="Twins or Real Face Prediction", command=predict)

predictButton.place(x=300,y=150)

predictButton.config(font=font1)

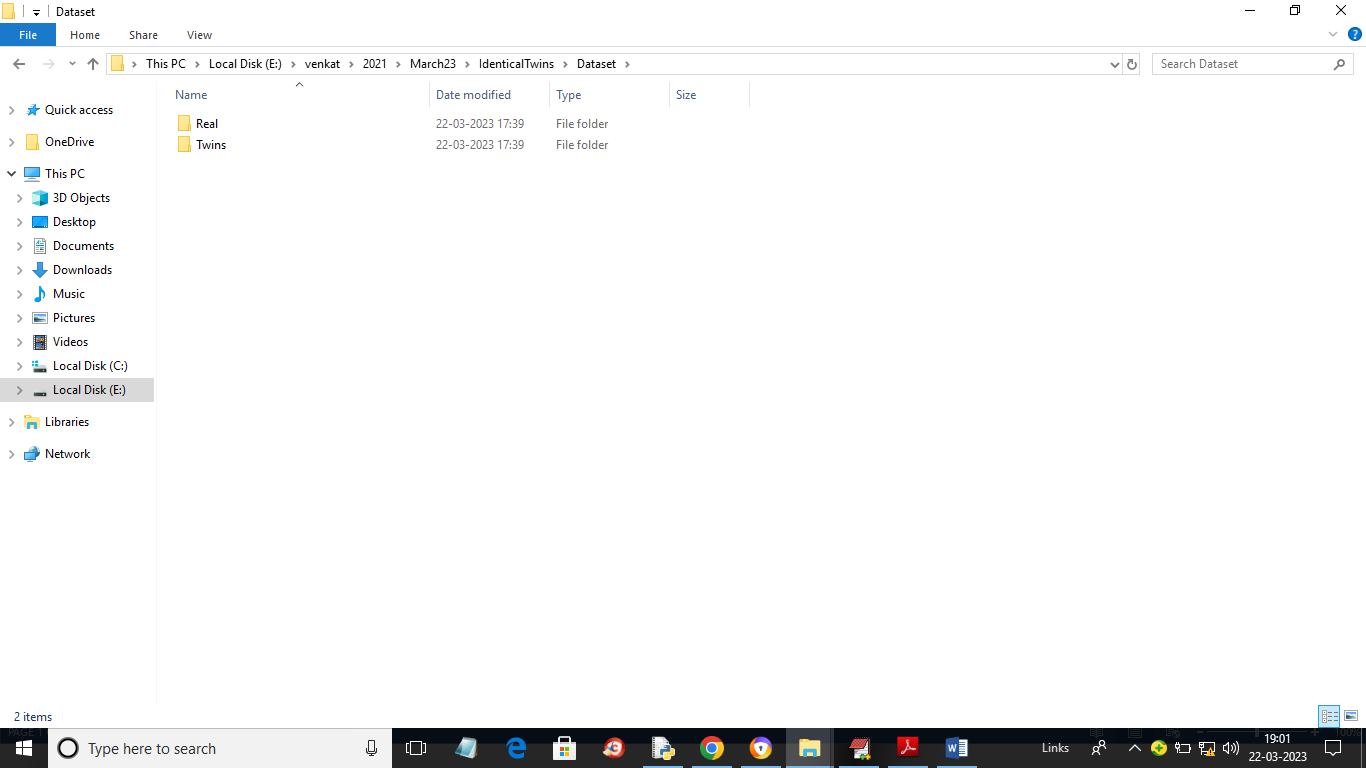
#main.config(bg='OliveDrab2')

main.mainloop()

## SCREEN SHOTS:

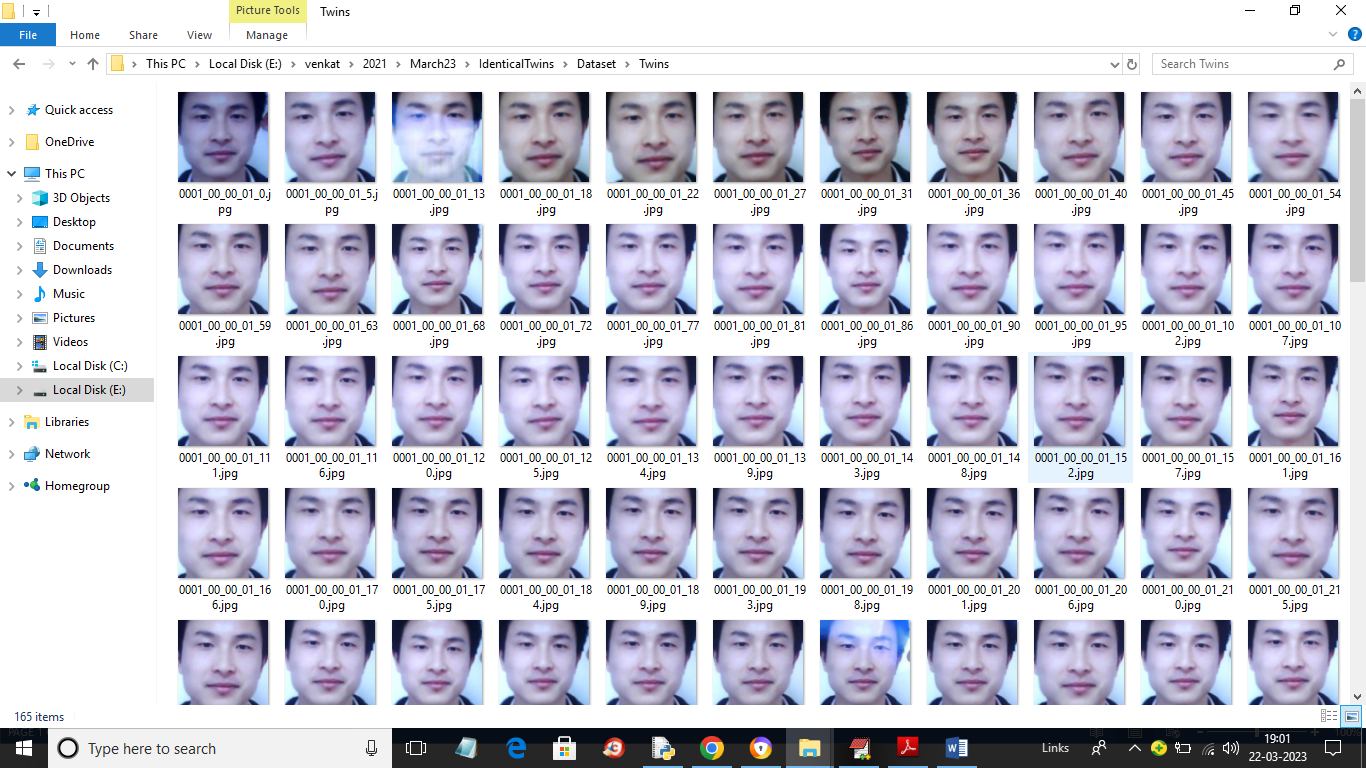
In real words twins faces are exists and this twins can utilize advantages to dupe peoples in examination or any other organizations. To detect such twins we are applying machine learning algorithms such as Naïve Bayes and Random Forest which may get trained on possible Real and Twins faces. Once after training we can input face to this trained model to identify weather face is Real or Twin. Before training we are applying various image processing techniques such as applying Bilateral Filters to enhance image quality and then convert image to Black & White format and then apply Object detection technique to detect face from image. This processed image will be input to Machine learning algorithm to train a model.

For training we are using below images dataset



**FIGURE 5.2.1: DATASETS**

In above screen we have two folders called Twins and Real and just go inside any folder to view images like below screen



**FIGURE 5.2.2: DATASET IMAGES**

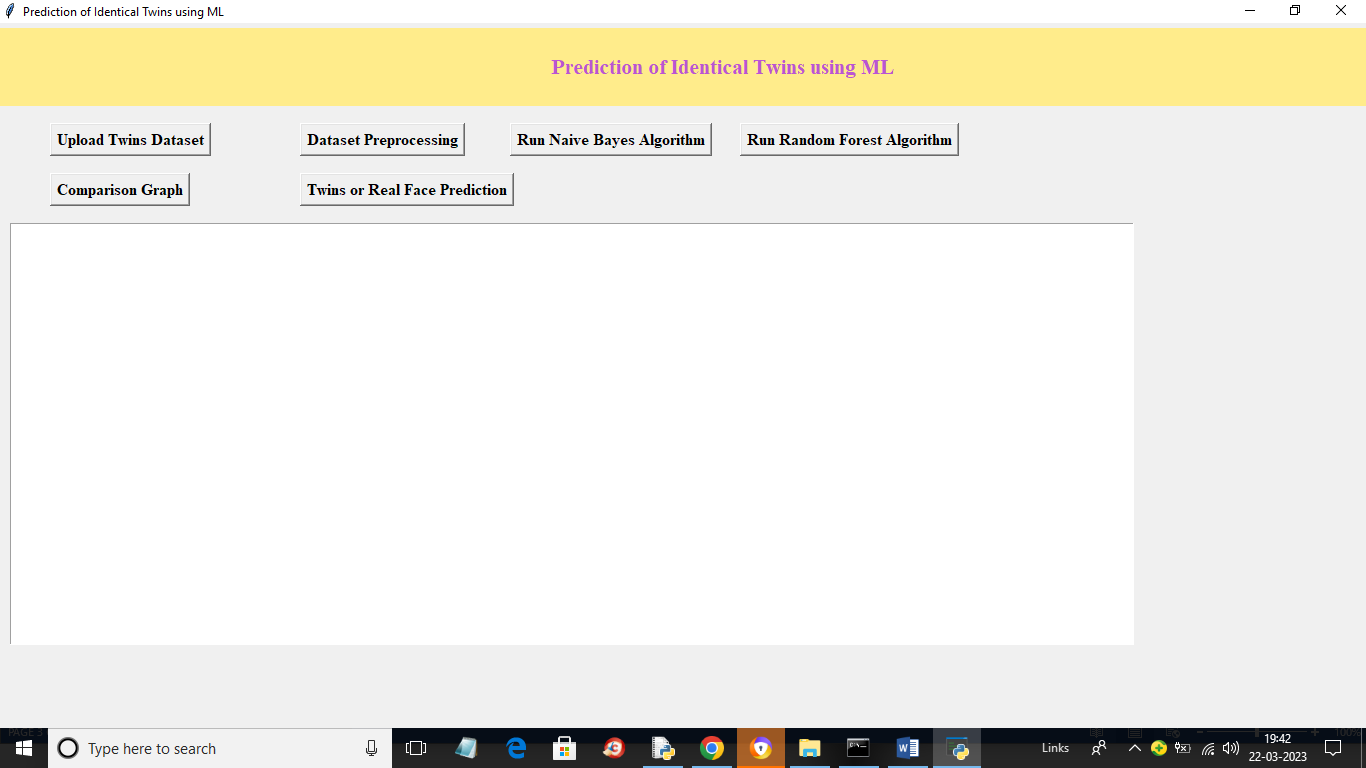
So by using above images we will evaluate performance of both Random Forest and Naïve Bayes Algorithm.

To implement this project we have designed following modules

1. Upload Twins Dataset: using this module we will upload dataset to application and then apply filtration and object detection techniques
2. Dataset Preprocessing: using this module we will normalized and then shuffle and split dataset into train and test where application using 80% dataset for training and 20% for testing
3. Run Naive Bayes Algorithm: 80% processed train images will be input to Naive Bayes Algorithm to train a model and this model will be applied on 20% test images to calculate prediction accuracy
4. Run Random Forest Algorithm: 80% processed train images will be input to Random Forest Algorithm to train a model and this model will be applied on 20% test images to calculate prediction accuracy
5. Comparison Graph: using this module we will plot comparison graph between both algorithms
6. Twins or Real Face Prediction: using this module we will upload test images and then algorithm will predict weather image is real or belongs to twins.

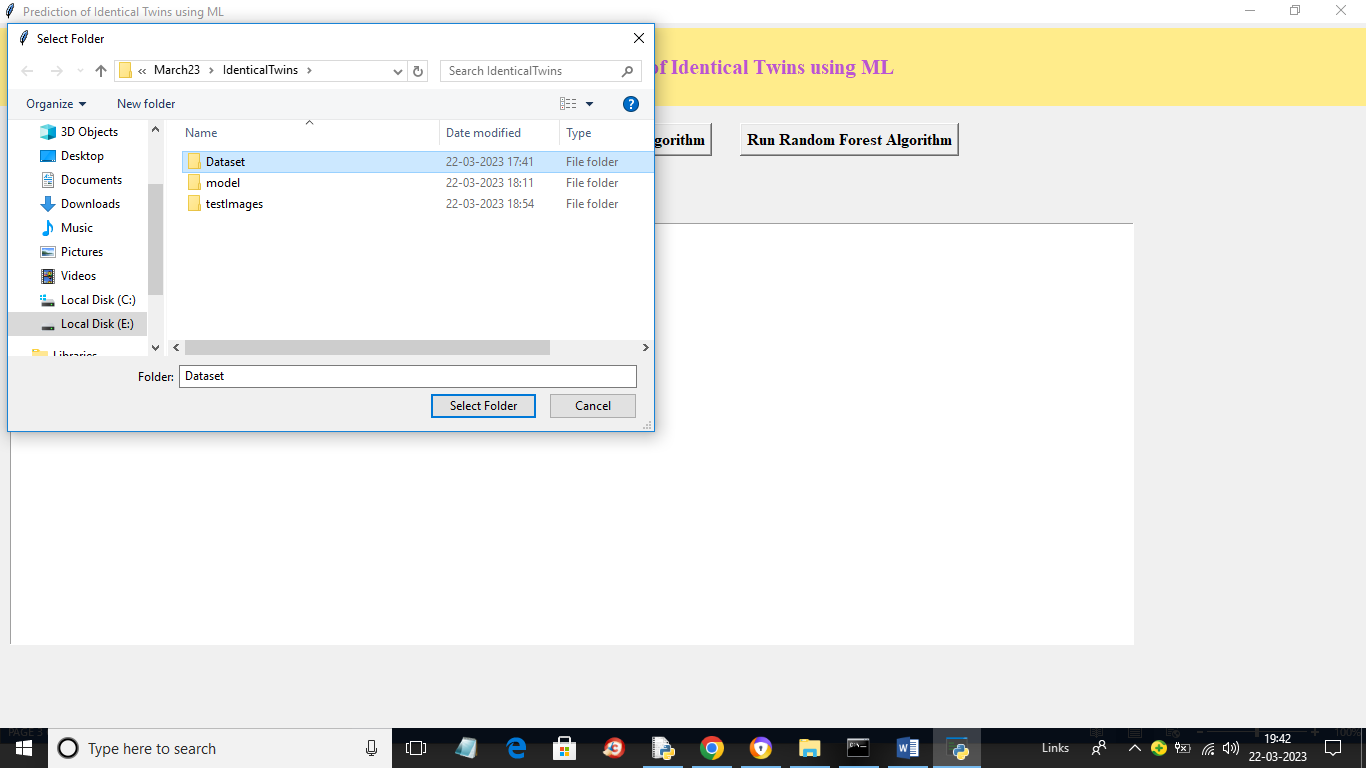
**SCREEN SHOTS**

To run project double click on ‘run.bat’ file to get below screen



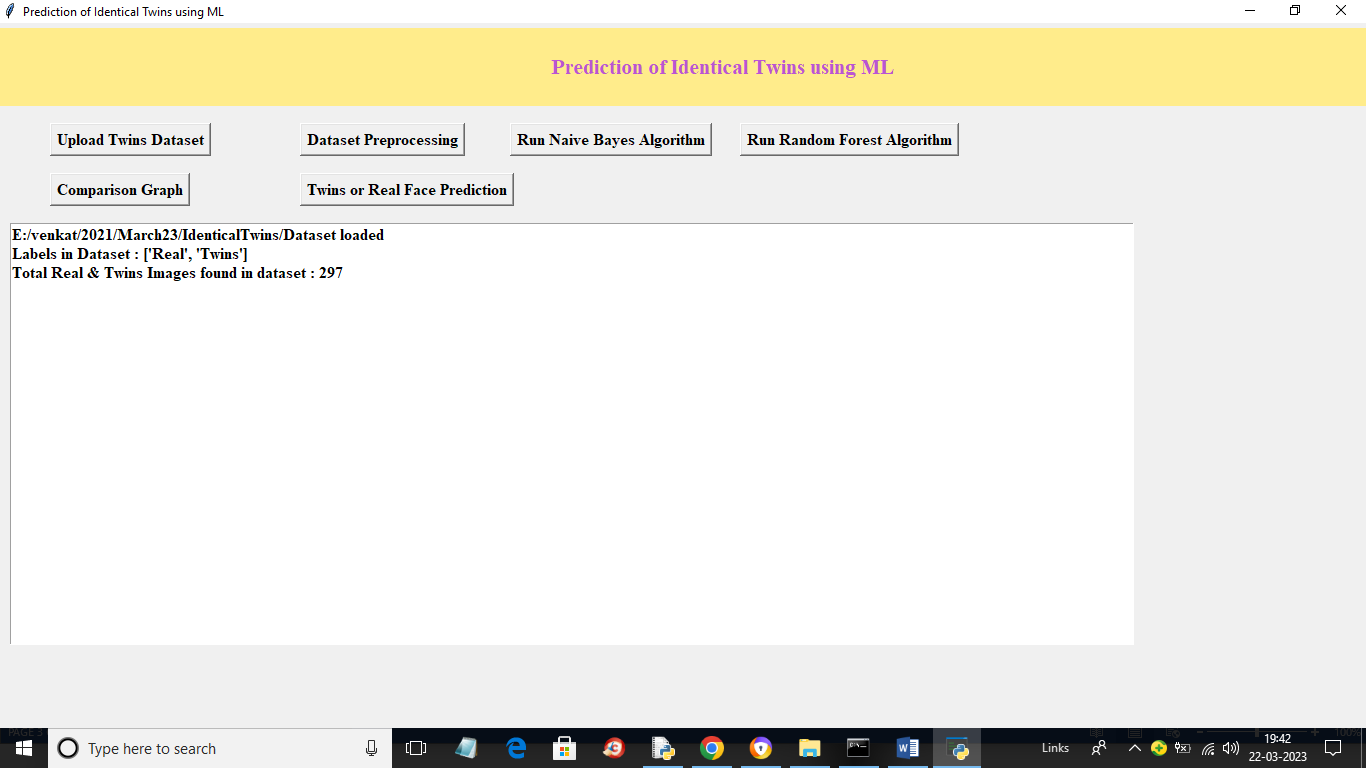
**FIGURE 5.2.3: USER INTERFACE**

In above screen click on ‘Upload Twins Dataset’ button to upload dataset and get below output



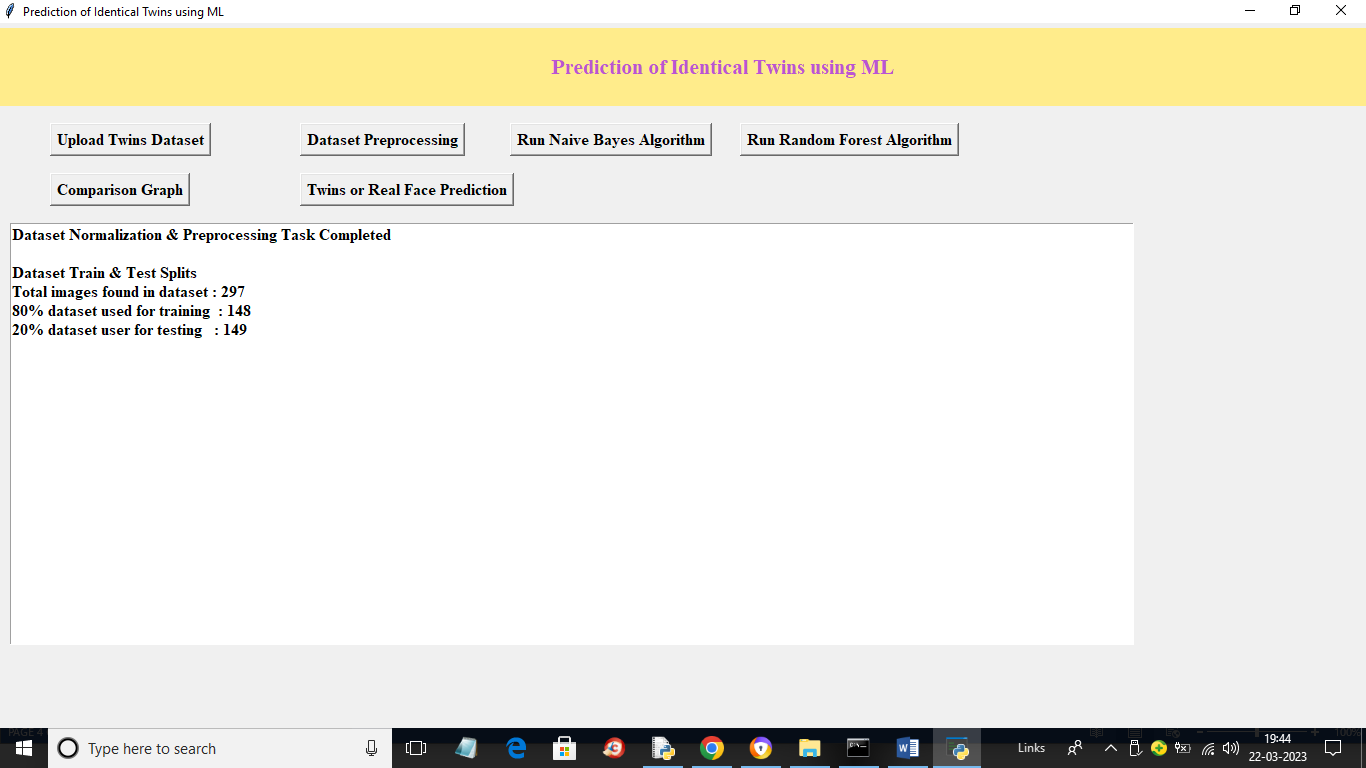
**FIGURE 5.2.4: UPLOADING DATA**

In above screen selecting and uploading ‘Dataset’ folder and then click on ‘Select Folder’ button to load dataset and get below output



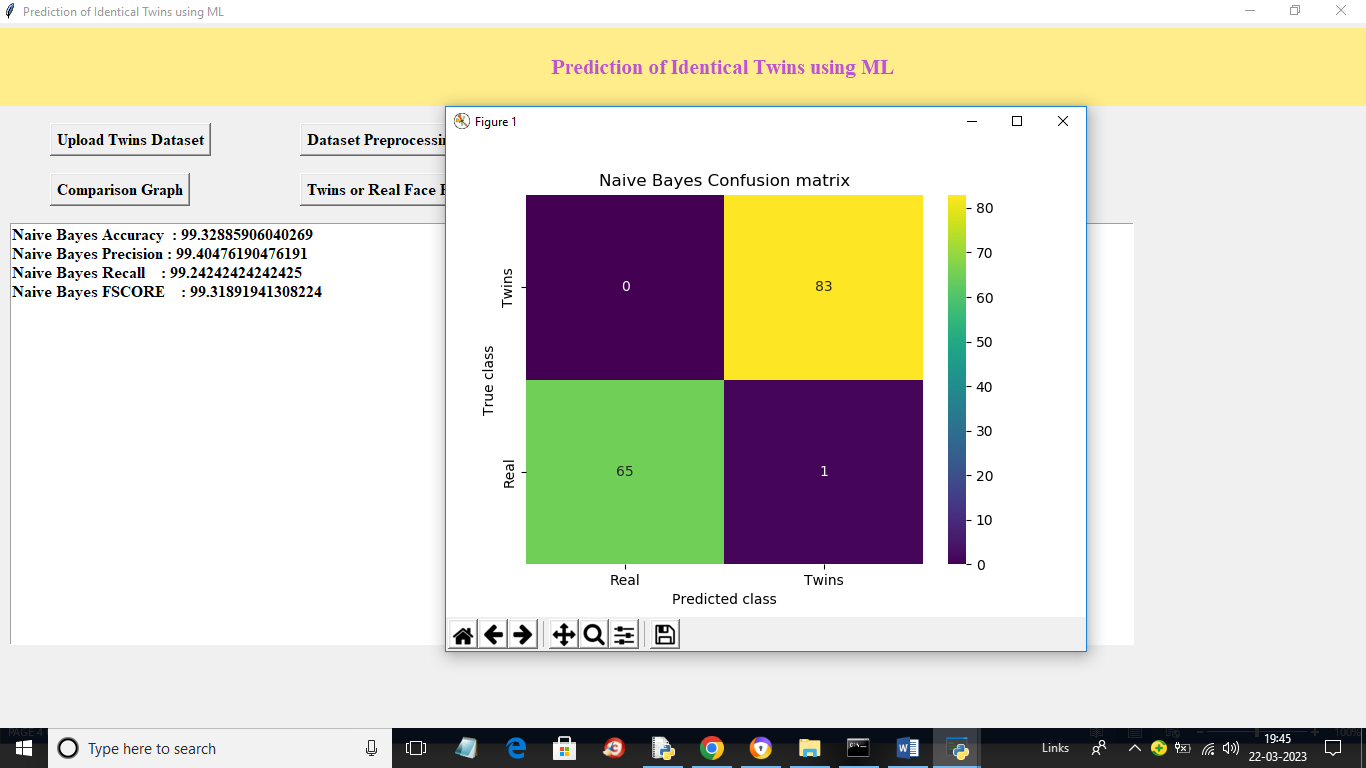
**FIGURE 5.2.5: TRAINING**

In above screen we can see dataset loaded and we can see available labels and images in the dataset and now click on ‘Dataset Preprocessing’ button to normalize, shuffle and split dataset into train and test and will get below output



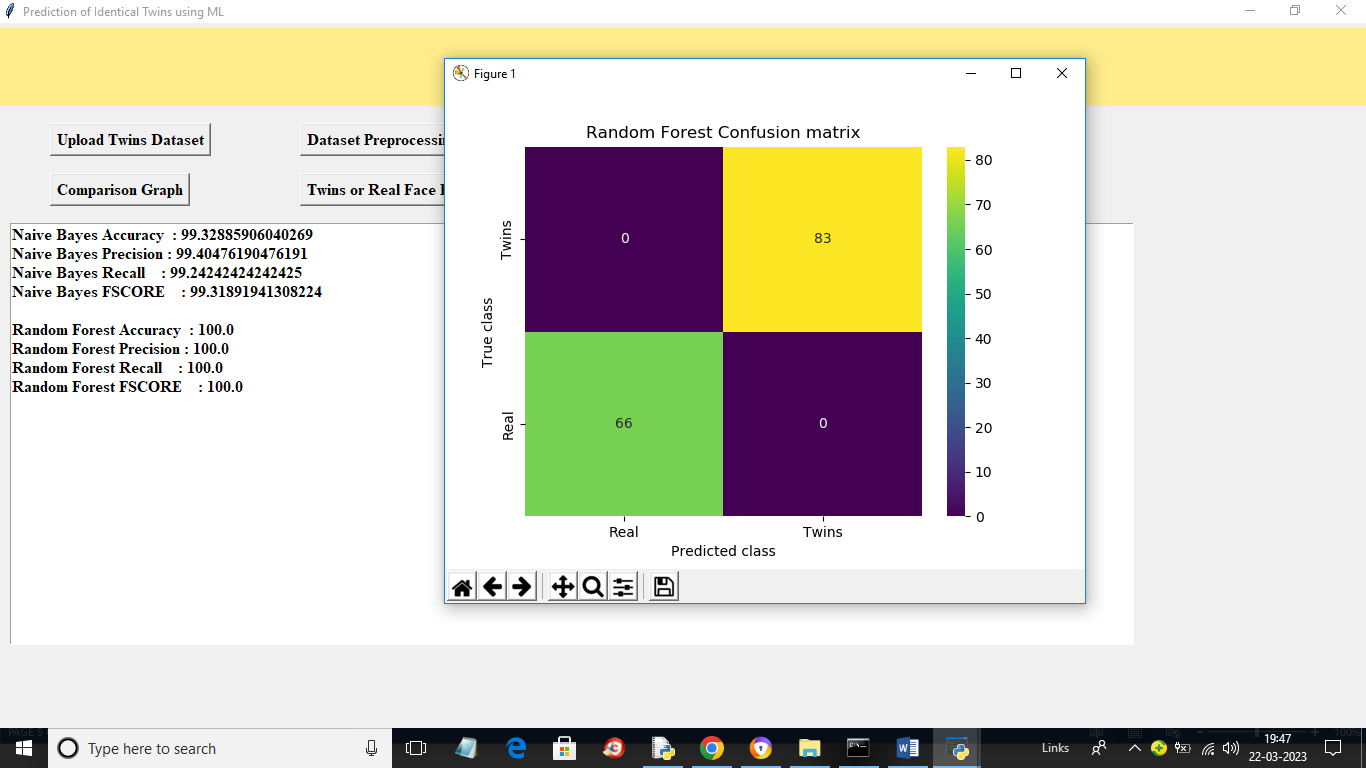
**FIGURE 5.2.6: TRAINING RESULTS**

In above screen we can see dataset processed and we can see total images used for train and test and now click on ‘Run Naïve Bayes Algorithm’ button to train Naïve Bayes and get below output



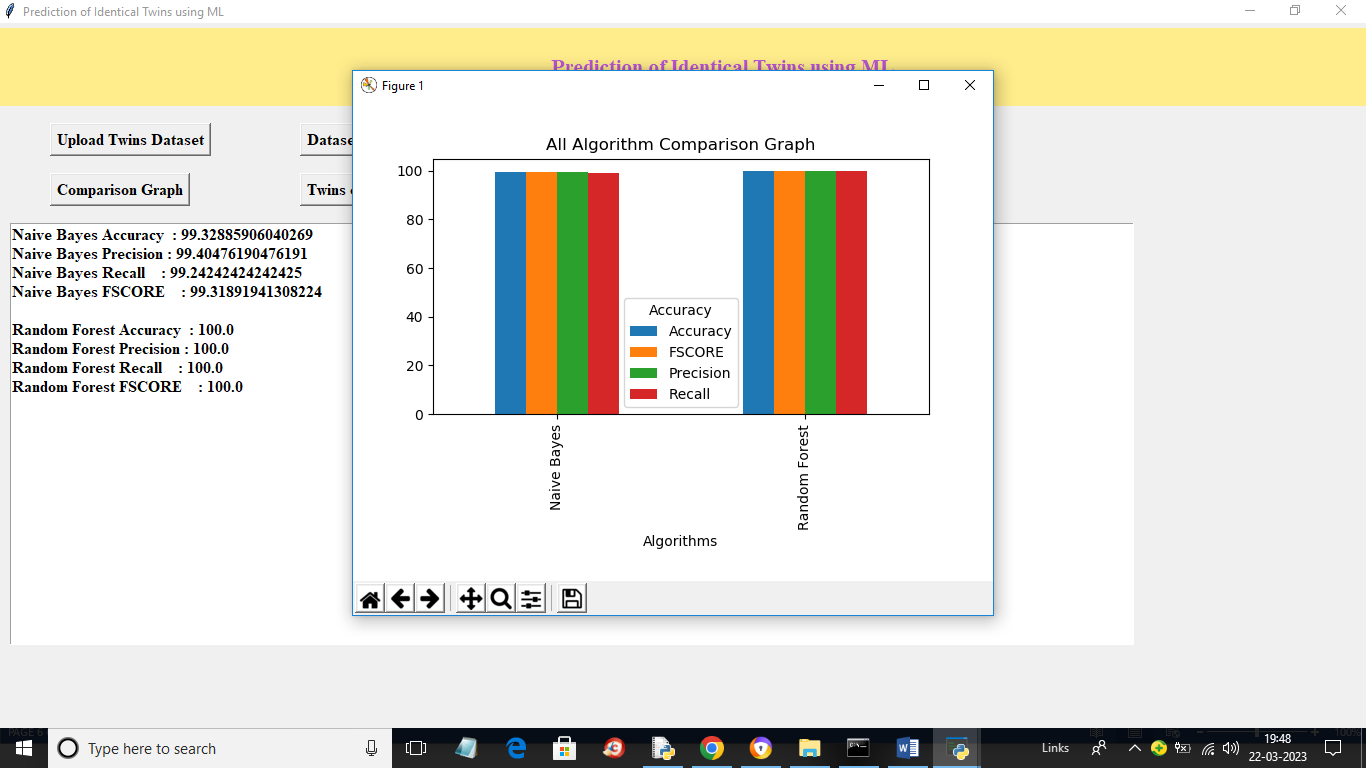
**FIGURE 5.2.7: NAÏVE BAYES**

In above screen with Naïve Bayes we got accuracy as 99% and we can see other metrics also and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and green and yellow boxes contains Correct Prediction count and blue boxes represents incorrect prediction count which is 1 only and now close above window and then click on ‘Run Random Forest’ button to train Random Forest and get below output



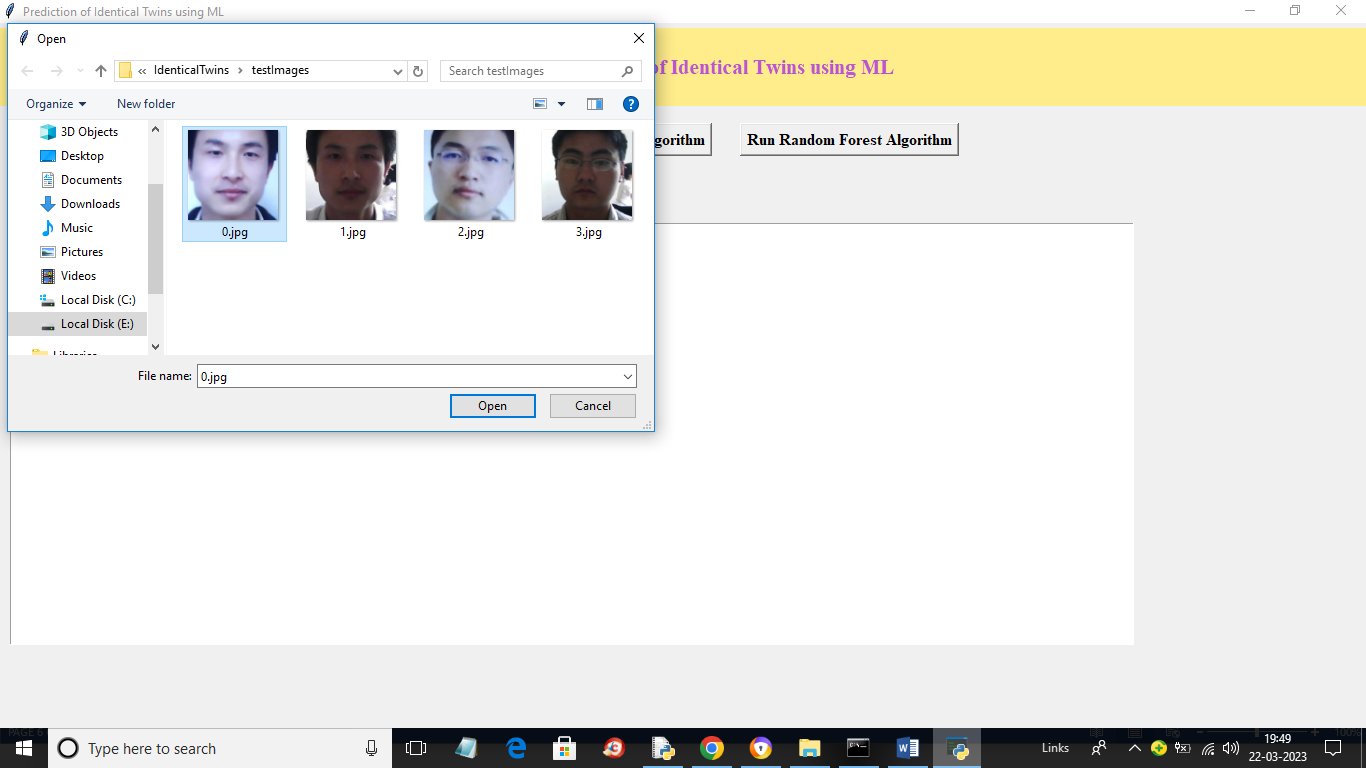
**FIGURE 5.2.8: RANDOM FOREST**

In above screen with Random Forest we got 100% accuracy and we can see confusion graph also and now click on ‘Comparison Graph’ button to get below graph



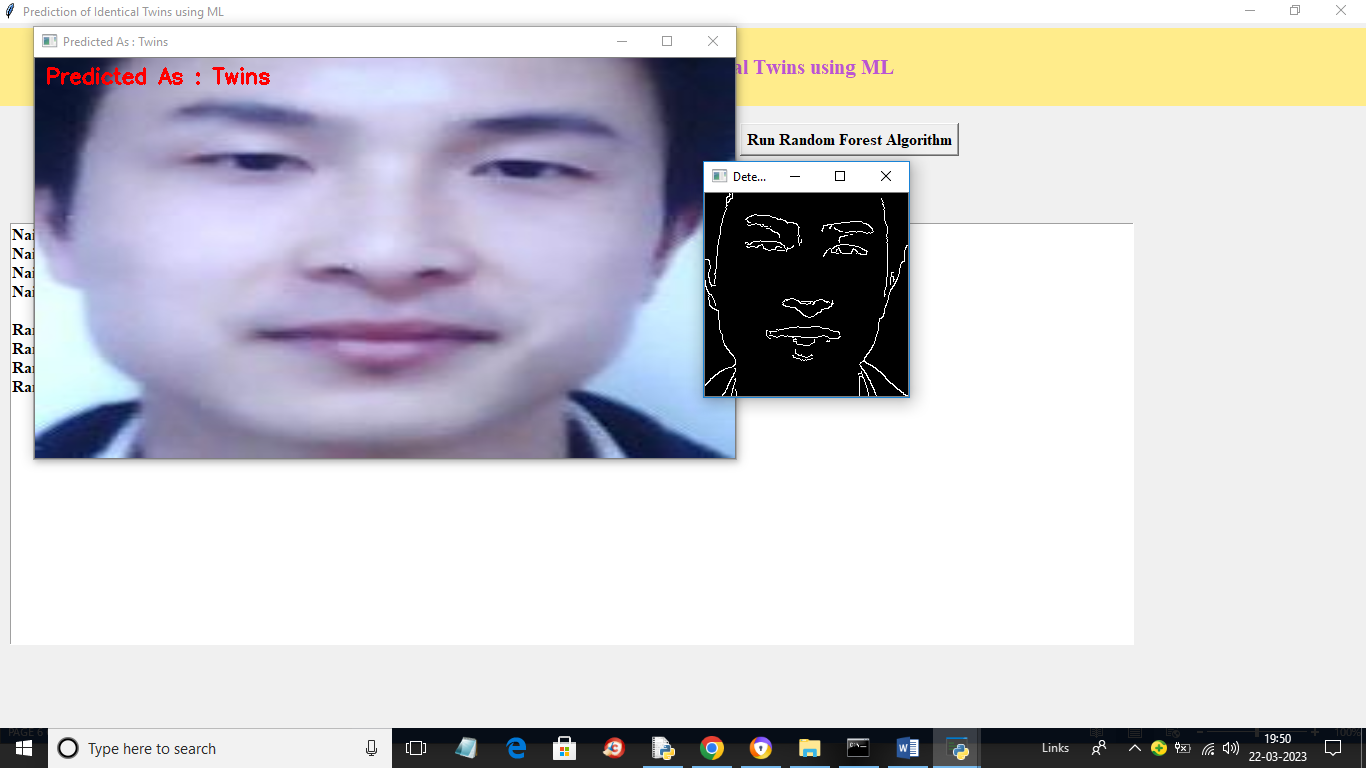
**FIGURE 5.2.9: COMPARISON**

In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in both algorithms Random Forest got high performance and now click on ‘Twins or Real Face Prediction’ button to upload test image and get below output



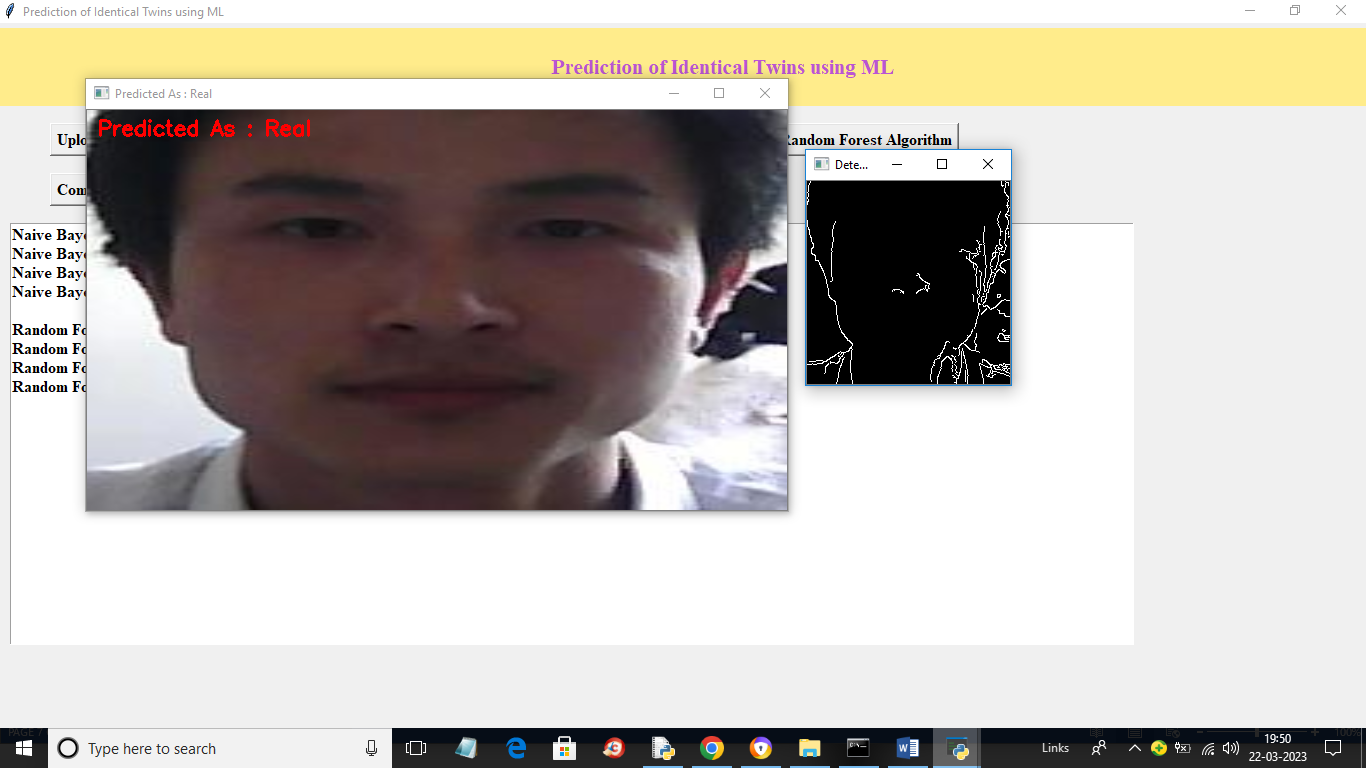
**FIGURE 5.2.10: TESTING**

In above screen selecting and uploading ‘0.jpg’ image and then click on ‘Open’ button to load image and get below output



**FIGURE 5.2.11: RESULT**

In above screen in red colour text we can see image predicted as Twins and we can see detected object in face in black and white colour and similarly you can upload and test other images



**FIGURE 5.2.12: RESULT-2**

In above screen image predicted as real.

## SOFTWARE ENVIRONMENT :

### **What is Python :**

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

1. [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
2. GUI Applications (like Kivy, Tkinter, PyQt etc. )
3. Web frameworks like Django (used by YouTube, Instagram, Dropbox)
4. Image processing (like Opencv, Pillow)
5. Web scraping (like Scrapy, BeautifulSoup, Selenium)
6. Test frameworks
7. Multimedia

#### **Advantages of Python :-**

Let’s see how Python dominates over other languages.

##### **Extensive Libraries**

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

##### **Extensible**

As we have seen earlier, Python can be**extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

##### **Embeddable**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities**to our code in the other language.

##### **Improved Productivity**

The language’s simplicity and extensive libraries render programmers**more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

##### **IOT Opportunities**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and**code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

##### **Readable**

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

##### **Object-Oriented**

This language supports both the **procedural and object-oriented**programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

##### **Free and Open-Source**

Like we said earlier, Python is **freely available.** But not only can you[**download Python**](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

##### **Portable**

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to**code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

##### **Interpreted**

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

#### **Advantages of Python Over Other Languages :**

##### **Less Coding**

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

##### **Affordable**

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

**The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

##### **Python is for Everyone**

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

#### **Disadvantages of Python**

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

##### **Speed Limitations**

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

##### **Weak in Mobile Computing and Browsers**

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

##### **Design Restrictions**

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can**raise run-time errors**.

##### **Underdeveloped Database Access Layers**

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

##### **Simple**

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

#### **History of Python : -**

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

### **What is Machine Learning : -**

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

#### **Categories Of Machine Leaning :-**

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

#### **Need for Machine Learning**

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

#### **Challenges in Machines Learning :-**

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

#### **Applications of Machines Learning :-**

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML

1. Emotion analysis
2. Sentiment analysis
3. Error detection and prevention
4. Weather forecasting and prediction
5. Stock market analysis and forecasting
6. Speech synthesis
7. Speech recognition
8. Customer segmentation
9. Object recognition
10. Fraud detection
11. Fraud prevention
12. Recommendation of products to customer in online shopping

#### **How to Start Learning Machine Learning?**

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.** And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed](http://blog.indeed.com/2019/03/14/best-jobs-2019/), Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

#### **How to start learning ML?**

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

##### **(a) Learn Linear Algebra and Multivariate Calculus**

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

##### **(b) Learn Statistics**

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!  
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

##### **(c) Learn Python**

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/)! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Keras](https://keras.io/), [TensorFlow](https://www.tensorflow.org/), [Scikit-learn](https://scikit-learn.org/stable/), etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as [**Fork Python**](https://practice.geeksforgeeks.org/courses/fork-python) available Free on GeeksforGeeks.

#### **Step 2 – Learn Various ML Concepts**

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

##### **(a) Terminologies of Machine Learning**

1. **Model –**A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
2. **Feature –**A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
3. **Target (Label) –**A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
4. **Training –**The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
5. **Prediction –**Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

##### **(b) Types of Machine Learning**

1. **Supervised Learning –**This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
2. **Unsupervised Learning –**This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
3. **Semi-supervised Learning –**This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
4. **Reinforcement Learning –**This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

#### **Advantages of Machine learning :-**

##### **1. Easily identifies trends and patterns**

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

##### **2. No human intervention needed (automation)**

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

##### **3. Continuous Improvement**

As [**ML algorithms**](https://data-flair.training/blogs/machine-learning-algorithms/) gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

##### **4. Handling multi-dimensional and multi-variety data**

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

##### **5. Wide Applications**

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

#### **Disadvantages of Machine Learning :-**

##### **1. Data Acquisition**

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

##### **2. Time and Resources**

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

##### **3. Interpretation of Results**

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

##### **4. High error-susceptibility**

[Machine Learning](https://en.wikipedia.org/wiki/Machine_learning) is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**Python Development Steps : -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.  
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

1. Print is now a function
2. Views and iterators instead of lists
3. The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
4. There is only one integer type left, i.e. int. long is int as well.
5. The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
6. Text Vs. Data Instead Of Unicode Vs. 8-bit

**Purpose :-**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

### **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

1. Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
2. Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

#### **Modules Used in Project :-**

**Tensorflow**

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google).‍

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

1. A powerful N-dimensional array object
2. Sophisticated (broadcasting) functions
3. Tools for integrating C/C++ and Fortran code
4. Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery](https://matplotlib.org/gallery/index.html).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

1. Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
2. Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Install Python Step-by-Step in Windows and Mac :**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

#### **How to Install Python on Windows and Mac :**

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

##### **Download the Correct version into the system**

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [https://www.python.org](https://www.python.org/)



**FIGURE5.3.1: INSTALL PYTHON**

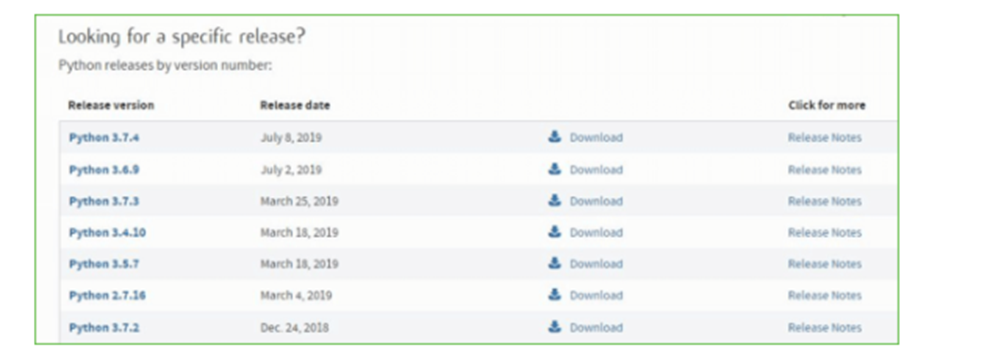
Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.



**FIGURE5.3.** **2: DOWNLOAD PYTHON**

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



**FIGURE 5.3.3: SELECT PYTHON VERSION**

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



**FIGURE5.3.4: SELECT THE OS**

* To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
* To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on

the Release Note Option.

**Installation of Python**

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**FIGURE5.3.5: OPEN DOWNLOADED PYTHON VERSION**

**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**FIGURE5.3.6: ADD PYTHON TO PATH**

**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



**FIGURE5.3.7: CLICK ON INSTALL**

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Verify the Python Installation**

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**FIGURE5.3.8: OPEN CMD**

**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python-v** and press Enter.



**FIGURE5.3.9: CHECK PYTHON IS INSTALLED**

**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

**Check how the Python IDLE works**

**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**FIGURE5.3.10: CHECK IDLE**

**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file.

**Click on File > Click on Save**



**FIGURE5.3.11: SAVE THE FILE**

**Step 5:** Name the file and save as type should be Python files. Click on SAVE.

Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print**

# CHAPTER-6

**TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## TYPES OF TESTS

**6.1 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**6.2 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**6.3 FUNCTIONAL TESTING**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**6.4 SYSTEM TESTING**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**6.5 WHITE BOX TESTING**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**6.6 BLACK BOX TESTING**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### **UNIT TESTING**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

1. All field entries must work properly.
2. Pages must be activated from the identified link.
3. The entry screen, messages and responses must not be delayed.

**Features to be tested**

1. Verify that the entries are of the correct format
2. No duplicate entries should be allowed
3. All links should take the user to the correct page.

### **Integration testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

### **Acceptance testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered

# CHAPTER-7

## FUTURE ENHANCEMENTS

In the domain of predicting identical twins using machine learning models like Random Forest and Naive Bayes, significant opportunities exist to improve both the methodologies and applications of these models. Below are some key areas for future enhancements:

### **1. Incorporating domain-specific features**

Most current models rely on generic feature extraction techniques without leveraging the unique biological and genetic traits associated with identical twins. Future research could focus on integrating domain-specific knowledge, such as genetic markers, facial recognition patterns, or behavioral traits, into the feature extraction process. By incorporating these specialized features, the models could achieve greater accuracy and robustness in predictions.

### **2. Expanding dataset diversity**

Existing datasets for identical twin studies often suffer from limited diversity in terms of ethnicity, age groups, and environmental factors. This limitation can lead to biased predictions when applied to broader populations. Future efforts could aim to build diverse, large-scale datasets that capture a wide range of genetic and phenotypic variations, ensuring the models generalize well across different populations.

### **3. Multi-Class Classification for Twin Similarity**

Current models typically focus on binary classification (identical vs. non-identical twins). However, moving towards a multi-class classification framework could open up new avenues, such as predicting degrees of similarity between twins or clustering twins based on shared phenotypic traits. This would provide more nuanced insights into twin relationships.

### **4. Leveraging deep learning models**

While Random Forest and Naive Bayes are effective, integrating advanced deep learning models such as Convolutional Neural Networks (CNNs) or Transformer-based architectures could improve the ability to process complex patterns in images, videos, or genetic sequences. Hybrid models that combine traditional machine learning with deep learning techniques could further enhance predictive performance.

### **5. Explainable AI for Biological Interpretability**

A significant challenge in ML models is their "black-box" nature, which limits interpretability. Future enhancements could involve incorporating explainable AI techniques to identify and explain the key features that drive predictions. For instance, highlighting specific genetic or visual traits that contribute to a prediction can provide actionable insights for researchers and clinicians.

### **6. Real-time twin identification**

Building systems capable of real-time prediction, such as identifying identical twins through live video streams, would have practical applications in various domains. This could include security systems, personalized healthcare, or educational settings. Optimizing computational efficiency and scalability would be critical for such advancements.

### **7. Integration with Genomic Data**

Advances in genomic sequencing and analysis provide a rich source of data for studying identical twins. Integrating machine learning models with genomic data pipelines could lead to breakthroughs in understanding genetic predispositions, heritable diseases, and phenotypic expressions unique to identical twins.

### **8. Cross-domain applications**

The methodologies developed for twin prediction could have applications in other domains, such as familial relationship detection, medical diagnostics, or personalized medicine. For instance, integrating these models with facial recognition or biometrics systems could enhance their utility in broader contexts.

### **9. Collaborative Filtering for Twin Studies**

Future systems could utilize collaborative filtering techniques to recommend personalized healthcare interventions, dietary plans, or lifestyle modifications for twins based on the shared preferences and outcomes of similar pairs. This approach would leverage predictive models to augment healthcare decision-making.

### **10. Ethical and Bias Considerations**

As ML models are increasingly applied to sensitive domains like genetic and biometric analysis, it is critical to address ethical concerns, including data privacy, consent, and bias. Future research should prioritize the development of transparent, fair, and privacy-preserving models that ensure the responsible application of technology.

These enhancements offer promising directions to further refine the predictive capabilities and expand the applications of machine learning in the context of identical twins. By combining advanced methodologies, interdisciplinary approaches, and ethical frameworks, researchers can unlock new possibilities in this fascinating field.

# CHAPTER-8

## TEST CASES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Test Case** | **Expected Results** | **Result** | **Remarks** |
|  | Load Dataset | Dataset should load successfully | Pass | If dataset not found, it will fail |
|  | Split Dataset | Dataset is split into training and testing datasets | Pass | Splitting fails if data format is incorrect |
|  | Train Random Forest Model | Model trains successfully and generates predictions | Pass | Requires proper feature-engineered dataset |
|  | Train Naive Bayes Model | Model trains successfully and generates predictions | Pass | If dataset contains missing values, it will fail |
|  | Test Random Forest Model | Predictions are generated for test dataset | Pass | Model accuracy may vary with dataset quality |
|  | Test Naive Bayes Model | Predictions are generated for test dataset | Pass | May fail for imbalanced datasets |
|  | Compare Model Accuracies | Accuracy of both models is calculated and compared | Pass | May provide misleading results on noisy data |
|  | Predict Identical Twins | System predicts whether twins are identical or not | Pass | Requires sufficient training data for accuracy |
|  | Validate Predictions | Validation set is used to confirm prediction reliability | Pass | If validation data is sparse, accuracy drops |

# CHAPTER-9

## CONCLUSION

This work has successfully provided a new pathway to support digital forensic investigation, employing artificial intelligence (machine learning) to improve existing face recognition systems. The current research work mainly concentrates on identifying the identical looking twins on basis of the features extracted. The proposed system uses Python as the programming language for development of the project. The proposed system uses various number of frameworks like Open Computer Vision Library (OpenCV), TesnorFlow, NumPy, Keras module and Naïve Bayes and Random Forest Algorithm. Flask framework is used to build a Graphical User Interface for the system. It is used to connect to cloud and store in the database using the queries. Amazon Public Cloud is used to store the features in the cloud. Service used is Database as a service. The practical performance of this project is analyzed and it shows that the system works perfectly for the limited dataset. This proposed work recognizes the identical twins and displays the result on the bases of features extracted, which may be any one among the three possible outputs ofthe system.

# CHAPTER-10

## REFERENCES

[1] Recognition techniques for differentiate similar faces and twin faces R. Prema ; P. Shanmuga priya IEEE 2018.

[2] Face recognition using wavelet transforms-based feature extraction & spatial differentiation-based pre- processing Shridhar S Shanbhag; Sourabh B; K. Manikantan; S.Ramachandran IEEE 2018.

[3] Distinguishing Identical Twins by Face Recognition P. Jonathon Phillips, Patrick J. Flynn, Kevin W. Bowyer, Richard W. Vorder Bruegge, Patrick J. Grother, George W. Quinn, Matthew Pruitt IEEE 2018.

[4] Differentiating Identical Twins by Using Conditional Face Recognition Algorithms Swati Y. Dhote, A. D. Gotmare , M. S.Nimbarte IJSR 2015.

[5] Face Recognition: Issues, Methods and Alternative Applications Waldemar Wójcik, Konrad Gromaszek and Muhtar Junisbekov INTEC 2016.

[6] Face Recognition: From Traditional to Deep Learning Methods Daniel Saez Trigueros, Li Meng, Margaret Hartnett IJCT 2018.

[7] Performance Analysis of Human Face Recognition Techniques Sharmila; Raman Sharma; Dhanajay Kumar; Vaishali Puranik; KritikaGautham2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) Year: 2019 | Conference Paper | Publisher: IEEE.

[8] Blur and Motion Blur Influence on Face Recognition Performance Katarina Knežević; Emilija MandićRankoPetrović; BrankaStojanovi2018 14th Symposium on Neural Networks and Applications (NEUREL) Year: 2018.

[9] Alister Cordiner, Philip Ogunbona, and WanqingLi,“Face Detection Using Generalised Integral ImageFeatures,”IEEE 2009.

# CHAPTER-11

## BASE PAPER

### **INTRODUCTION**

Advancements in machine learning and predictive modeling have opened up new frontiers in medical and genetic research. Predicting identical twins based on genetic and phenotypic markers has significant implications for healthcare, personalized medicine, and forensic science. Traditional methods for identifying identical twins often rely on labor-intensive genetic testing or observational studies, which can be both time-consuming and costly. Leveraging machine learning techniques such as Naive Bayes and Random Forest algorithms offers a promising approach for automating and enhancing the accuracy of twin prediction while reducing reliance on manual methodologies.

### **PROBLEM STATEMENT**

Develop a machine learning-based system that predicts the likelihood of two individuals being identical twins using genetic and phenotypic data.

This problem can be broken down into three distinct tasks:

1. Preprocessing and analyzing genetic and phenotypic datasets.
2. Training machine learning models (Naive Bayes and Random Forest) for predictive analysis.
3. Evaluating and comparing the models’ performance based on accuracy, sensitivity, and specificity.
4. In this paper, we focus on implementing a robust framework for predicting identical twins by using real-world datasets. The system is designed to achieve high accuracy and reliability while addressing scalability for large-scale data.

### **EXISTING SYSTEM**

Existing systems for identifying identical twins often rely on conventional methods such as DNA fingerprinting, which is accurate but resource-intensive. While some research has been conducted on automated approaches, many existing systems suffer from limitations in scalability and adaptability, especially when dealing with diverse datasets.

#### **DISADAVNTAGES:**

* High cost and time consumption for traditional methods.
* Limited adaptability to non-standardized datasets.
* Inefficient handling of noise and missing data in current models.

**PROPOSED SYSTEM:**

The proposed system aims to address these limitations by employing machine learning algorithms such as Naive Bayes and Random Forest for predicting identical twins. The system will analyze key genetic and phenotypic markers and provide a probabilistic prediction of twin status. The use of ensemble methods like Random Forest ensures robustness against noise, while Naive Bayes offers computational efficiency.

#### **ADVANTAGES:**

* High accuracy and sensitivity in predictions.
* Enhanced efficiency and scalability for large datasets.
* Effective handling of noisy and incomplete data.

### **MODULES:**

* **Dataset Upload:** Using this module, users can upload datasets containing genetic and phenotypic markers for analysis.
* **Data Preprocessing:** This module cleans and normalizes the dataset by handling missing values, removing outliers, and scaling features for machine learning.
* **Feature Selection:** This module identifies the most relevant features influencing the prediction of identical twins.
* **Naive Bayes Model Training:** This module trains a Naive Bayes classifier using the preprocessed dataset and evaluates its performance.
* **Random Forest Model Training:** This module trains a Random Forest classifier and compares its performance to the Naive Bayes model.
* **Prediction Interface:** This module allows users to input test data and obtain predictions on twin status.

### **SYSTEM REQUIREMENTS**

#### **HARDWARE REQUIREMENTS**

* System: i3 or above.
* RAM: 4 GB or higher.
* Hard Disk: 40 GB or higher.

#### **SYSTEM REQUIREMENTS**

* Operating System: Windows 8 or above.
* Programming Language: Python.
* Libraries: NumPy, Pandas, Scikit-learn, Matplotlib.

### **SCOPE**

The future scope of this system includes integrating advanced genetic markers and phenotypic data from diverse populations to enhance accuracy further. Future research could also explore deep learning models to capture more complex relationships in the data. Expanding the framework to include epigenetic factors and environmental variables would make the system more comprehensive and reliable. Potential applications extend to prenatal diagnostics, personalized healthcare planning, and forensic investigations.

### **CONCLUSION**

In this paper, we present a machine learning-based approach for predicting identical twins using Naive Bayes and Random Forest algorithms. The framework successfully:

1. Preprocesses and analyzes genetic and phenotypic datasets.
2. Employs robust algorithms for predictive modelling.
3. Achieves high accuracy and scalability.
4. Provides a user-friendly interface for real-time predictions.

This system has the potential to revolutionize twin prediction by making it more accessible, efficient, and accurate.

### **REFERENCES**

* “Introduction to Machine Learning,” https://www.coursera.org/learn/machine-learning.
* “Scikit-learn: Machine Learning in Python,” https://scikit-learn.org/stable/.
* “Random Forests,” https://towardsdatascience.com/random-forest.
* J. L. Shlens, “A tutorial on Principal Component Analysis,” arXiv preprint arXiv:1404.1100, 2014.
* S. Haykin, “Neural Networks and Learning Machines,” Pearson, 2009.