Institute of Computer Technology

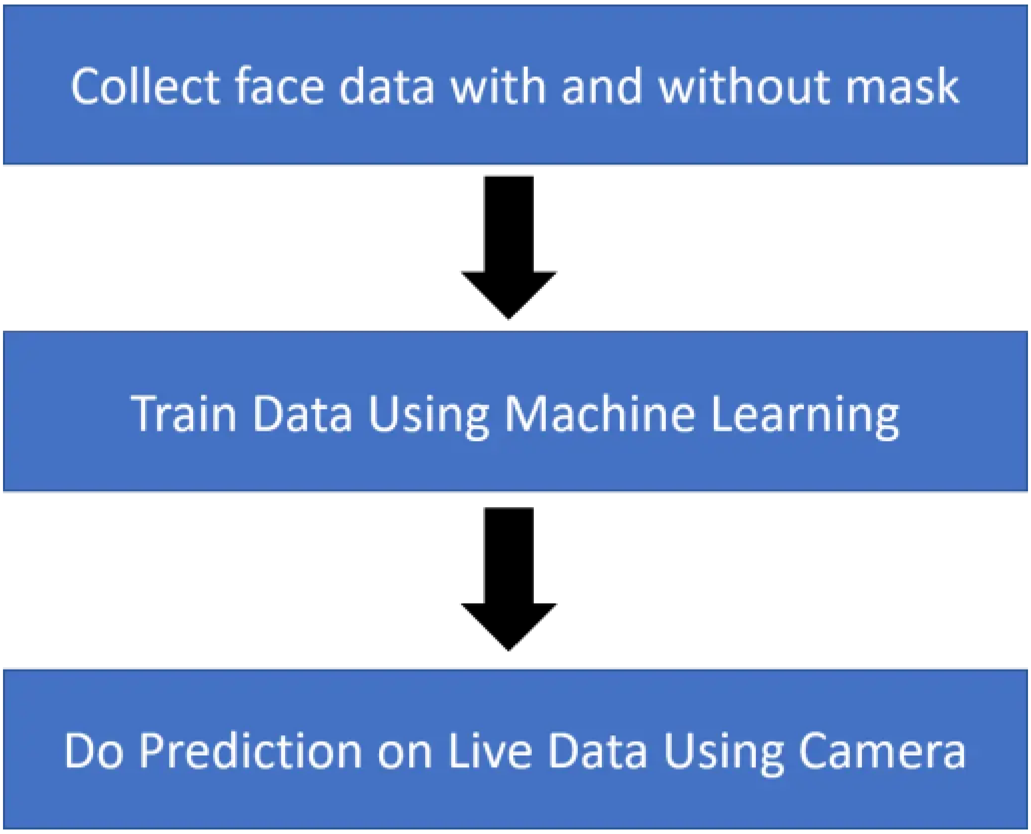
B.Tech Computer Science Engineering

Sub : (2CSE60E20) IMPR

Face Mask Detection

Steps to Perform Image Processing :

- Load images using Python or any other programming you are working on.
- Convert images into array
- And finally apply some algorithm on that array


Collect face data with and without mask
Train Data Using Machine Learning
Do Prediction on Live Data Using Camera

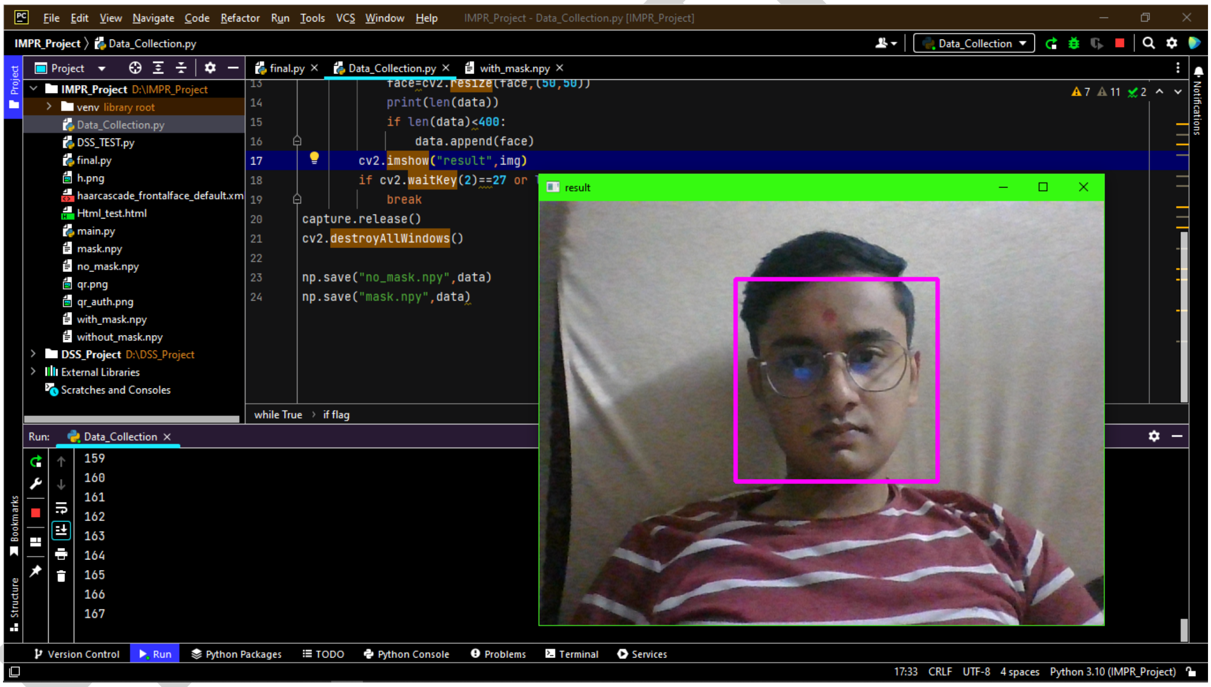
Data Collection :

Code :

```
import cv2
import numpy as np
haar_data=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
 data=[]
 capture = cv2.VideoCapture(0)
 while True :
    flag,img = capture.read()
    if flag :
    faces=haar_data.detectMultiScale(img)
    for x,y,w,h in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,255),4)
        face=img[y:y+h,x:x+w,:]
        face=cv2.resize(face,(50,50))
        print(len(data))
        if len(data)<400:
            data.append(face)
    cv2.imshow("result",img)
    if cv2.waitKey(2)==27 or len(data)>=200:
        break
capture.release()
cv2.destroyAllWindows()

np.save("without_mask.npy",data)
np.save("with_mask.npy",data)
```

Output :

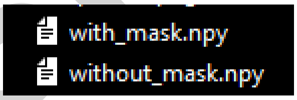


```
                face=cv2.resize(face,(50,50))
                print(len(data))
                if len(data)<400:
                    data.append(face)
            cv2.imshow("result",img)
            if cv2.waitKey(2)==27 or l
                break
    capture.release()
    cv2.destroyAllWindows()

    np.save("no_mask.npy",data)
    np.save("mask.npy",data)
```

```
159
160
161
162
163
164
165
166
167
```

Collections Data File :



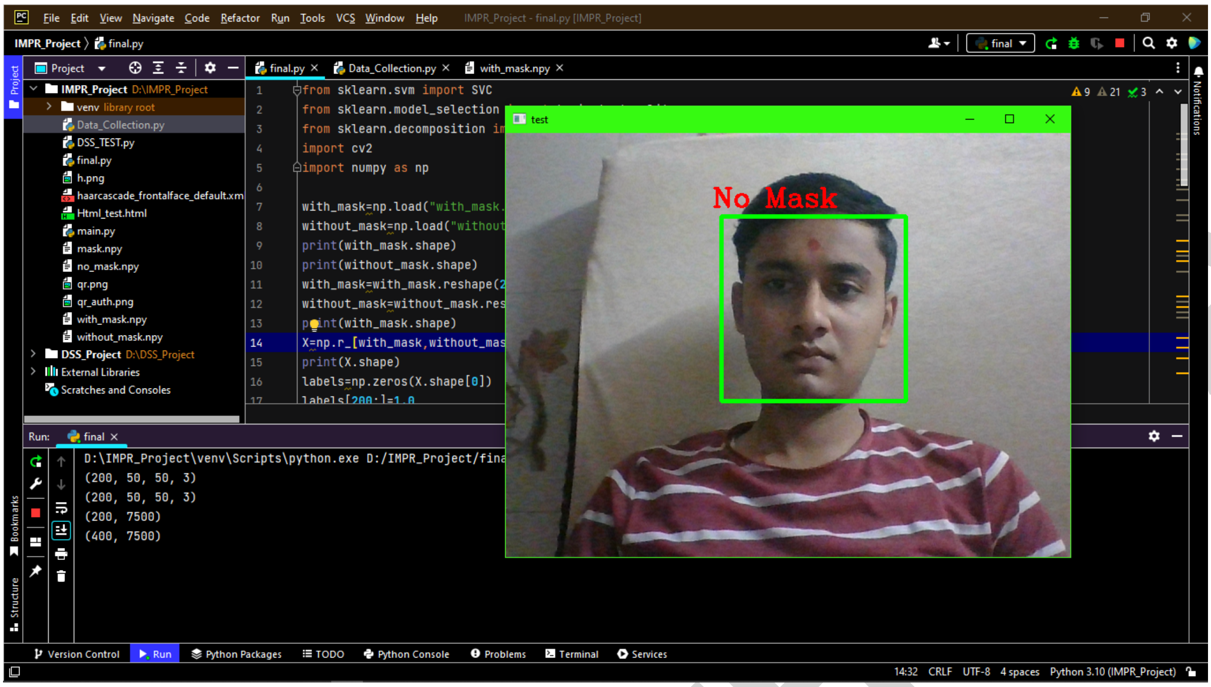
with_mask.npy
without_mask.npy

Applying Algorithm on Datasets for Mask Detection :

Code :

```python
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
import cv2
import numpy as np
with_mask=np.load("with_mask.npy")
without_mask=np.load("without_mask.npy")
print(with_mask.shape)
print(without_mask.shape)
with_mask=with_mask.reshape(200,50*50*3)
without_mask=without_mask.reshape(200,50*50*3)
print(with_mask.shape)
X=np.r_[with_mask,without_mask]
print(X.shape)
labels=np.zeros(X.shape[0])
labels[200:]=1.0
names={0:"Mask",1:"No Mask"}
x_train,x_test,y_train,y_test=train_test_split(X,labels,test_size=0.20)
# print(x_train.shape)
pca=PCA(n_components=3)
x_train=pca.fit_transform(x_train)
svm = SVC()
svm.fit(x_train, y_train)
haar_data=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
capture=cv2.VideoCapture(0)
data=[]
font=cv2.FONT_HERSHEY_COMPLEX
while True:
    flag,img=capture.read()
    if flag:
    faces = haar_data.detectMultiScale(img)
    # haar_data.detectMultiScale(img)
    for x, y, w, h in faces:
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 4)
        face_fetch=img[y:y+h,x:x+w, :]
        face_fetch=cv2.resize(face_fetch,(50,50))
        face_fetch=face_fetch.reshape(1,-1)
        face_fetch=pca.transform(face_fetch)
        pred = svm.predict(face_fetch)[0]
        n = names[int(pred)]
        cv2.putText(img,n,(x-10,y-11),font,1,(0,0,255),2)
        # print(n)
    cv2.imshow("test", img)
    # 27 is ascii value of escape
    if cv2.waitKey(2) == 27:
        break
capture.release()
cv2.destroyAllWindows()
```

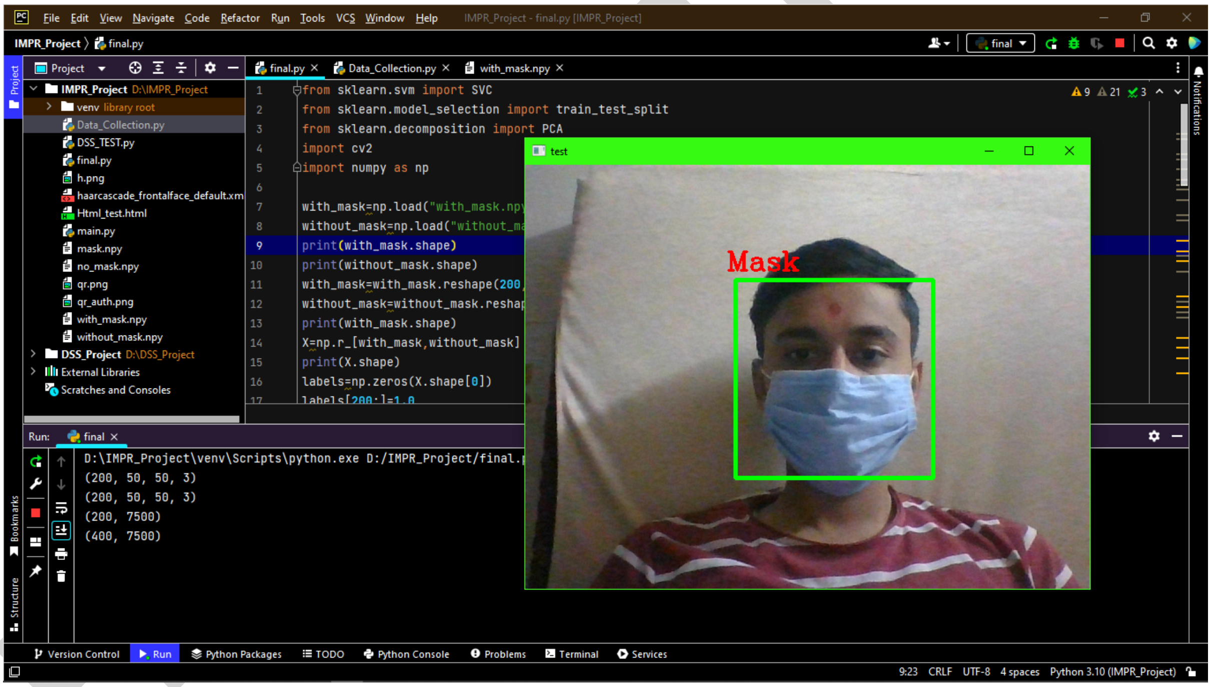
Output :



```python
from sklearn.svm import SVC
from sklearn.model_selection
from sklearn.decomposition im
import cv2
import numpy as np

with_mask=np.load("with_mask.
without_mask=np.load("without
print(with_mask.shape)
print(without_mask.shape)
with_mask=with_mask.reshape(2
without_mask=without_mask.res
print(with_mask.shape)
X=np.r_[with_mask,without_mas
print(X.shape)
labels=np.zeros(X.shape[0])
labels[200:]=1.0
```

```
D:\IMPR_Project\venv\Scripts\python.exe D:/IMPR_Project/fina
(200, 50, 50, 3)
(200, 50, 50, 3)
(200, 7500)
(400, 7500)
```



```python
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
import cv2
import numpy as np

with_mask=np.load("with_mask.npy
without_mask=np.load("without_ma
print(with_mask.shape)
print(without_mask.shape)
with_mask=with_mask.reshape(200,
without_mask=without_mask.reshap
print(with_mask.shape)
X=np.r_[with_mask,without_mask]
print(X.shape)
labels=np.zeros(X.shape[0])
labels[200:]=1.0
```

```
D:\IMPR_Project\venv\Scripts\python.exe D:/IMPR_Project/final.p
(200, 50, 50, 3)
(200, 50, 50, 3)
(200, 7500)
(400, 7500)
```