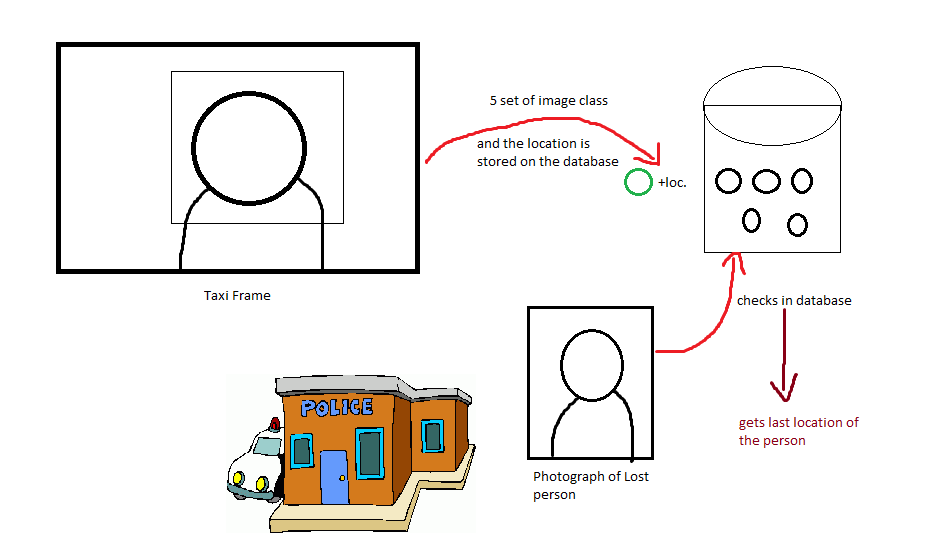
CONTENTS

1. Overview
2. Face Detected(Code)
3. Further Code Overview
   1. Designing Face Recogisation System
4. Feature Extraction Function Code
5. Final Storage Code
6. Detection Code for Police.
7. Future Developments

**Overview**

In this project of implementation of Image processing (Computer Vision) on CCTV footage we are mainly focusing on the camera inside the taxies.  
So, there was an initial project various features to make a car or viehcle secure like alarm system, GPS tracking etc. So, what we propose with this project is that how about govt. offers every Taxi this level of security and return put up a CCTV cam inside each taxi. For security and surviellence purposes.  
 So, the CCTV part of the whole project is done in this project.

So, Consider a case that a person in sitting in that smart Taxi. The sytem takes 5 photos and makes a class of photographs.  
Now, this person never reaches home. His Mom takes his photo to the nearest police station to make a FIR.  
Now what the police officials can do is scan that photo and scan the whole database if the photo matches the police official will know all the details including the last location, No. plate of the Taxi, Taxi person name and address etc.

**So, this project is divided into several parts:**-Detection of face from CCTV  
-Storage of the face from CCTV on database  
-Assigning the face the same class.  
- Making the Police end code for the recognisation of the face.  
-Comparing both the faces.  
-Displaying the class of the image.

**Code for face detection**

faceDetector = vision.CascadeObjectDetector();

%Get the input device using image acquisition toolbox,resolution = 640x480 to improve performance

obj =imaq.VideoDevice('winvideo', 1, 'YUYV\_320x240','ROI', [1 1 320 240]);

set(obj,'ReturnedColorSpace', 'rgb');

figure('menubar','none','tag','webcam');

while (true)

frame=step(obj);

bbox=step(faceDetector,frame);

boxInserter = vision.ShapeInserter('BorderColor','Custom',...

'CustomBorderColor',[255 255 0]);

videoOut = step(boxInserter, frame,bbox);

imshow(videoOut,'border','tight');

f=findobj('tag','webcam');

if (isempty(f));

[hueChannel,~,~] = rgb2hsv(frame);

% Display the Hue Channel data and draw the bounding box around the face.

figure, imshow(hueChannel), title('Hue channel data');

rectangle('Position',bbox,'EdgeColor','r','LineWidth',1)

hold off

noseDetector = vision.CascadeObjectDetector('Nose');

faceImage = imcrop(frame,bbox);

imshow(faceImage)

noseBBox = step(noseDetector,faceImage);

noseBBox(1:1) = noseBBox(1:1) + bbox(1:1);

videoInfo = info(obj);

ROI=get(obj,'ROI');

VideoSize = [ROI(3) ROI(4)];

videoPlayer = vision.VideoPlayer('Position',[300 300 VideoSize+30]);

tracker = vision.HistogramBasedTracker;

initializeObject(tracker, hueChannel, bbox);

while (1)

% Extract the next video frame

frame = step(obj);

% RGB -> HSV

[hueChannel,~,~] = rgb2hsv(frame);

% Track using the Hue channel data

bbox = step(tracker, hueChannel);

% Insert a bounding box around the object being tracked

videoOut = step(boxInserter, frame, bbox);

%Insert text coordinates

% Display the annotated video frame using the video player object

step(videoPlayer, videoOut);

pause (.2)

end

% Release resources

release(obj);

release(videoPlayer);

close(gcf)

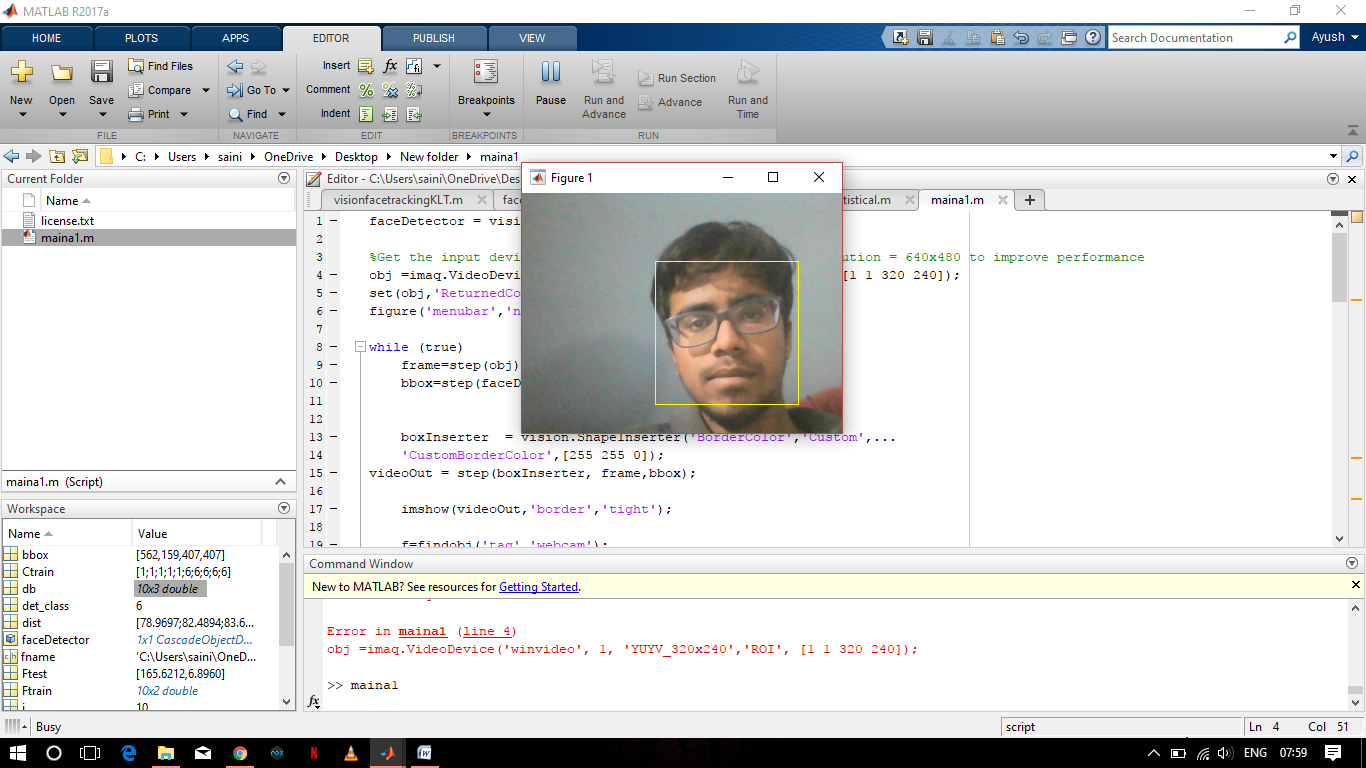
break

end

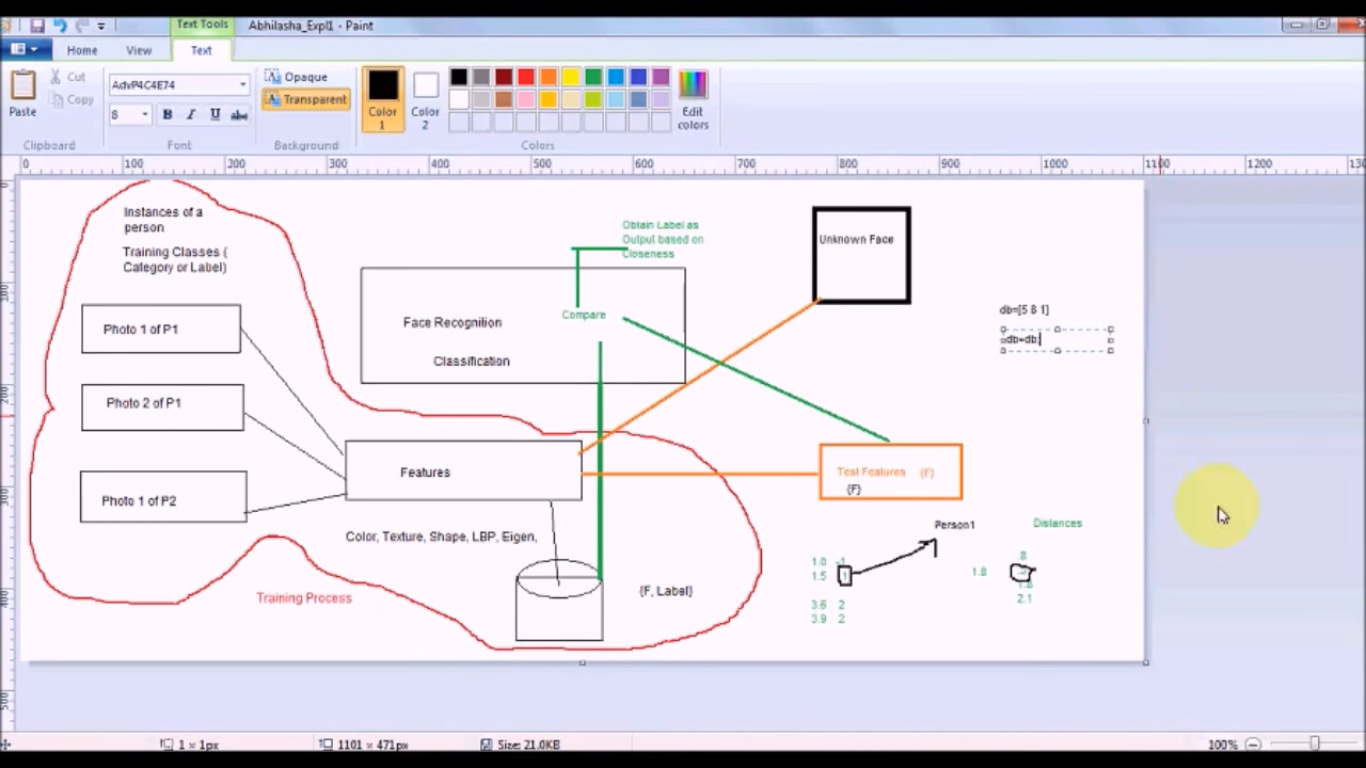
pause(0.05)

end

release(obj)

**Output:**

**Coding Overview:**

****

**Designing face Recogninsation System**

Here In the diagram we have a database of photo graphs of different people.

Different Photos of the same person can be classfied into a class and now all these photos can be called as part of this class.  
To recognise an unknown face if we subtract two photo graphs we might get many diiference even if they are of the same person.  
So, to remove this redundancy we use Features.  
Here in this case we assume that the mean and the standard deviation as the unique charactistic feature of a photograph.

In our case, we have to first extract the photo from the CCTV footage and store it.  
Then extract it features and store it on database.  
Now this feature function will be avilable to both the unknown face and the face extracted from the footage.

Features extracted will be stored on a database.  
Now for the facial recognisation system firstly the feature extraction function works on both the unknown face and the known face.  
Now these extracted features are compared while finding the distances between the features and finding the least amongst them and declaring the class with which it finds the least distance.

**Feature extraction code**

function [F]=FeatureStatistical(im)

im=double(im);

m=mean(mean(im));

s=std(std(im));

F=[m,s];

**Final Code for storage**

clear all;

faceDetector = vision.CascadeObjectDetector();

%Get the input device using image acquisition toolbox,resolution = 640x480 to improve performance

obj =imaq.VideoDevice('winvideo', 1, 'YUY2\_320x240','ROI', [1 1 320 240]);

set(obj,'ReturnedColorSpace', 'rgb');

figure('menubar','none','tag','webcam');

i=0;

k=1;

try

load db;

c=size(db,1);

c=c+1;

catch

disp("not yet");

end

while true

frame=step(obj);

bbox=step(faceDetector,frame);

i=i+1;

disp(i);

if i == 10 || i == 20 || i == 30 || i == 40 || i == 50

out = imcrop(frame,bbox);

imshow(out);

out1 = rgb2gray(out);

filename=sprintf('new%d.jpg',k)

imwrite(out1, filename);

im=imread(filename)

F=FeatureStatistical(im);

k=k+1;

try

load db;

F=[F c];

db=[db;F];

save db.mat db

catch

c=1;

db=[F c];

save db.mat db

end

end

boxInserter = vision.ShapeInserter('BorderColor','Custom',...

'CustomBorderColor',[255 255 0]);

videoOut = step(boxInserter, frame,bbox);

imshow(videoOut,'border','tight');

f=findobj('tag','webcam');

if (isempty(f));

[hueChannel,~,~] = rgb2hsv(frame);

% Display the Hue Channel data and draw the bounding box around the face.

figure, imshow(hueChannel), title('Hue channel data');

rectangle('Position',bbox,'EdgeColor','r','LineWidth',1)

hold off

noseDetector = vision.CascadeObjectDetector('Nose');

faceImage = imcrop(frame,bbox);

imshow(faceImage)

noseBBox = step(noseDetector,faceImage);

noseBBox(1:1) = noseBBox(1:1) + bbox(1:1);

videoInfo = info(obj);

ROI=get(obj,'ROI');

VideoSize = [ROI(3) ROI(4)];

videoPlayer = vision.VideoPlayer('Position',[300 300 VideoSize+30]);

tracker = vision.HistogramBasedTracker;

initializeObject(tracker, hueChannel, bbox);

while (1)

% Extract the next video frame

frame = step(obj);

% RGB -> HSV

[hueChannel,~,~] = rgb2hsv(frame);

% Track using the Hue channel data

bbox = step(tracker, hueChannel);

% Insert a bounding box around the object being tracked

videoOut = step(boxInserter, frame, bbox);

%Insert text coordinates

% Display the annotated video frame using the video player object

step(videoPlayer, videoOut);

pause (.2)

end

% Release resources

release(obj);

release(videoPlayer);

close(gcf)

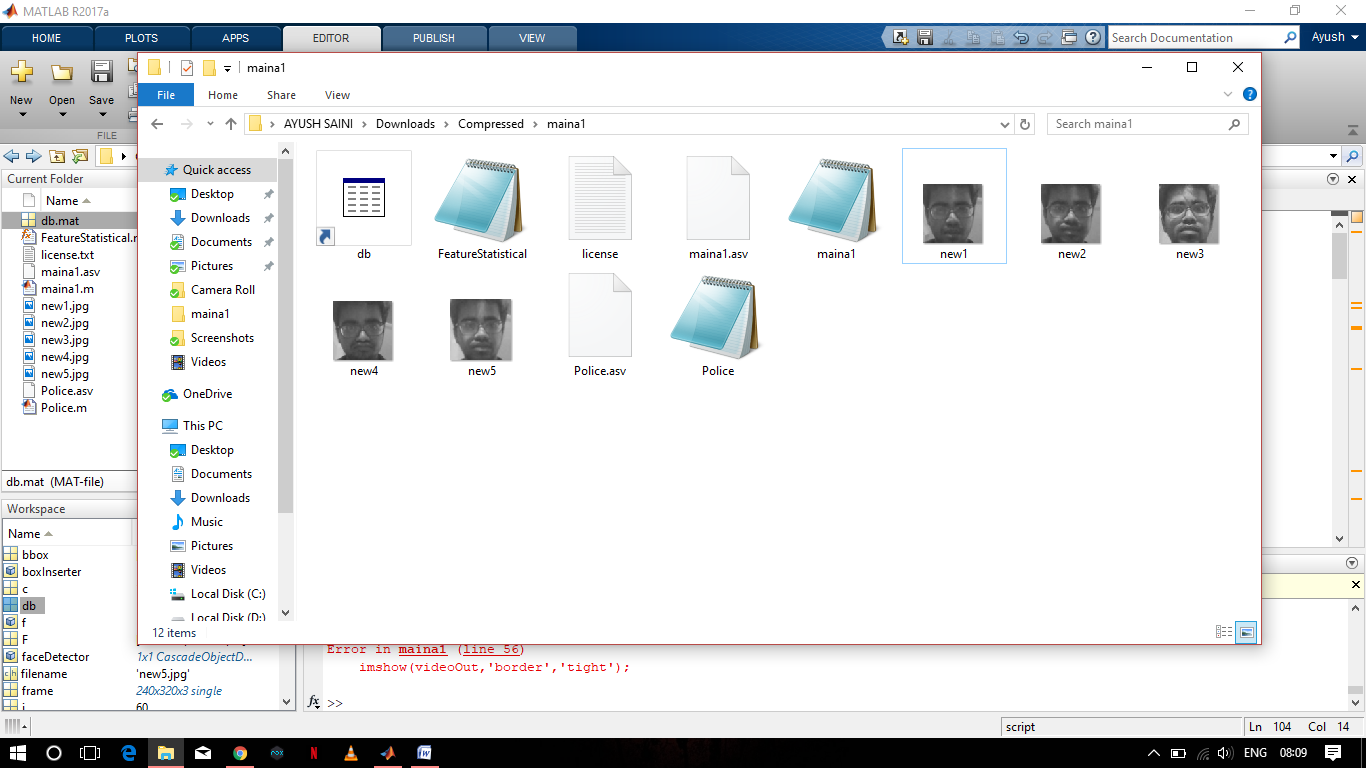
break

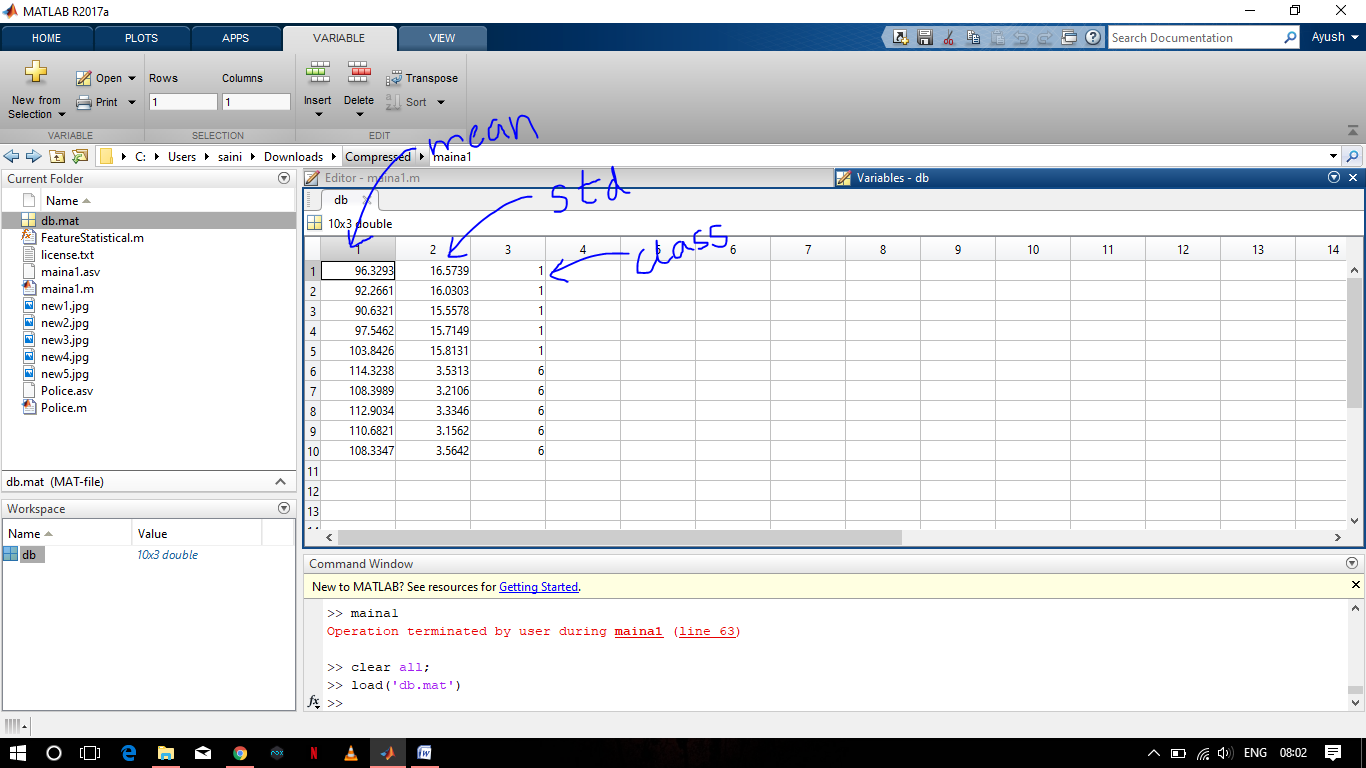
end

pause(0.05)

end

release(obj)

****

****

**Police Detection Code**

clc;

close all;

faceDetector = vision.CascadeObjectDetector();

[fname path]=uigetfile('.jpg','Enter Lost person photo');

fname=strcat(path,fname);

im=imread(fname);

bbox=step(faceDetector,im);

im1 = imcrop(im,bbox);

im2 = rgb2gray(im1);

imshow(im2);

title('Input Image');

Ftest=FeatureStatistical(im2);

load db.mat

Ftrain=db(:,1:2);

Ctrain=db(:,3);

for(i=1:size(Ftrain,1))

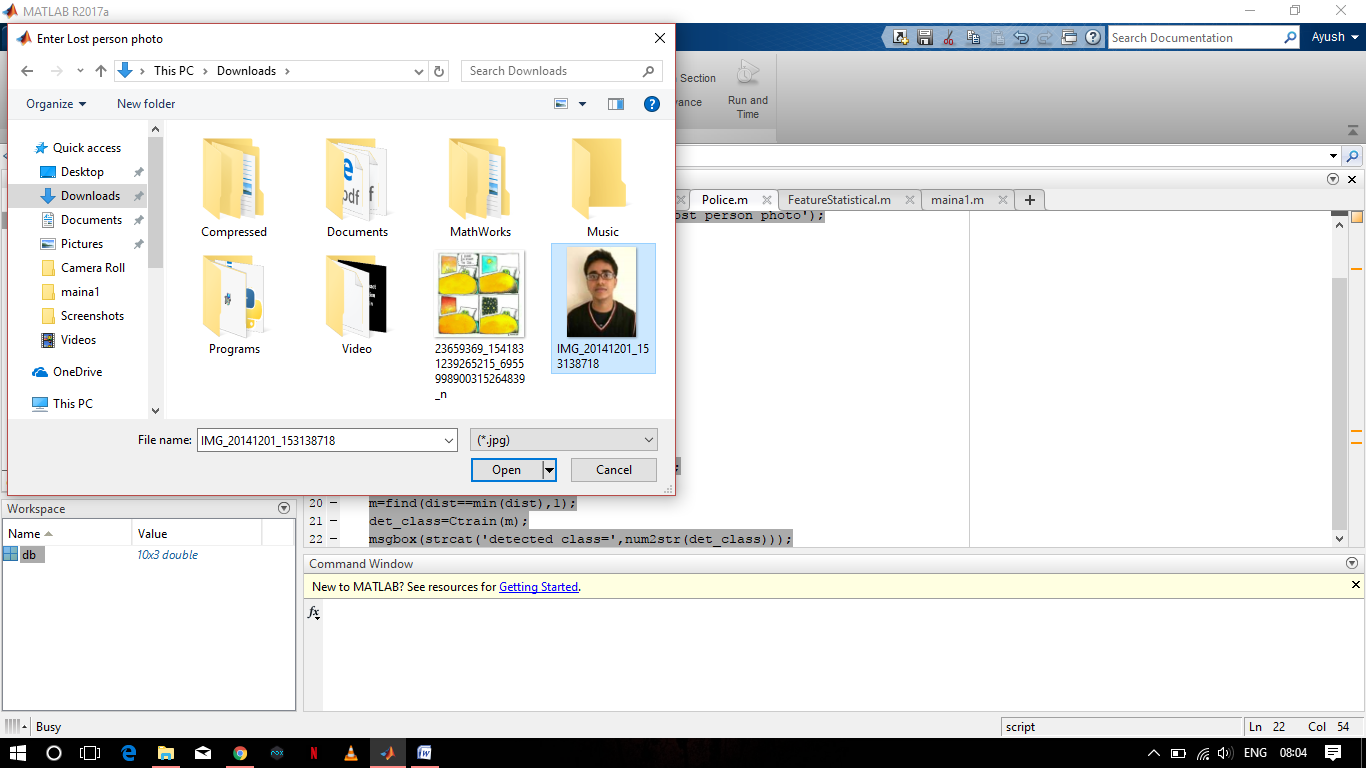
dist(i,:)=sum( abs(Ftrain(i,:)-Ftest));

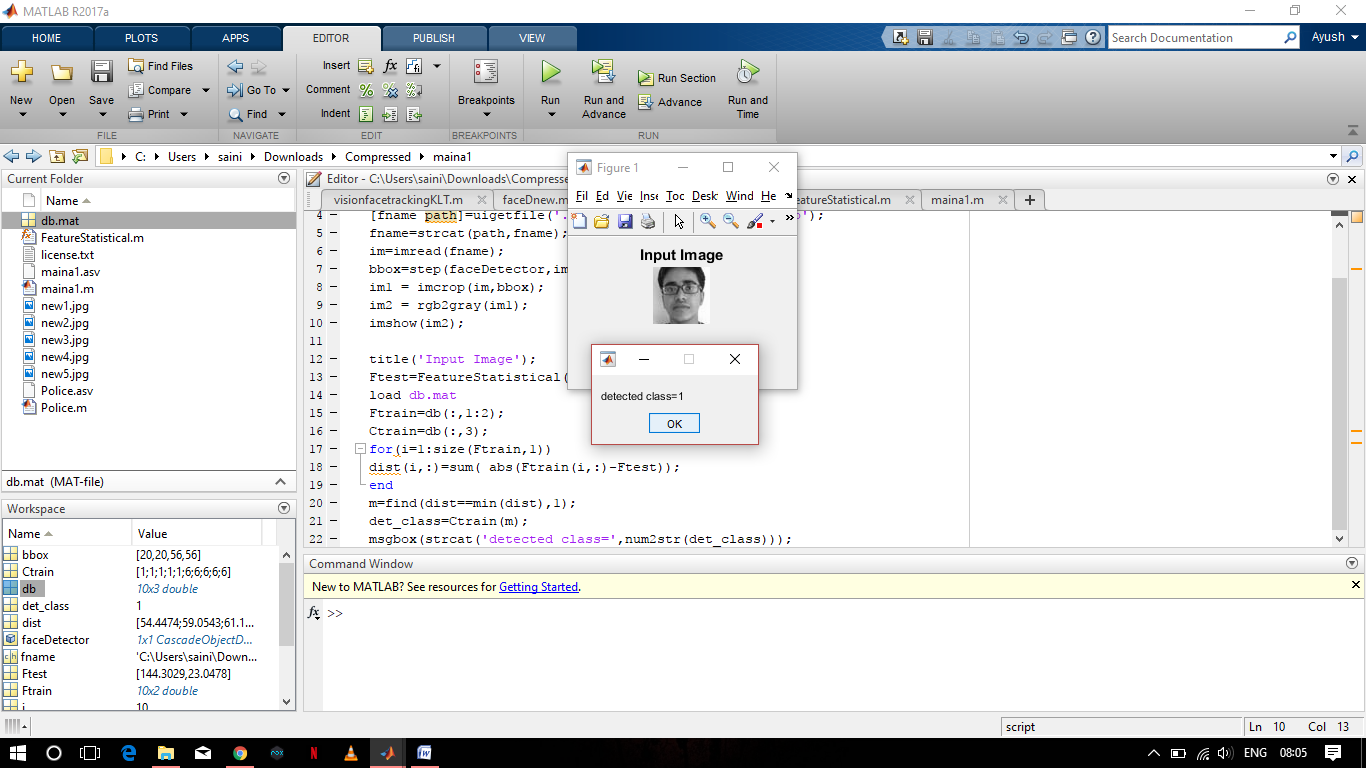
end

m=find(dist==min(dist),1);

det\_class=Ctrain(m);

msgbox(strcat('detected class=',num2str(det\_class)));

****

****

**FUTURE SCOPE**

Now, We have sucessfully recognised the face at the Police Station, we can now implement it with the GPS device and the rest of the system. Police Officials then will be able to get the last location of the person, Name address of the Taxi driver, etc.  
We also have to work on the security of the database system so not to allow anyone else the police offiicials to access the database