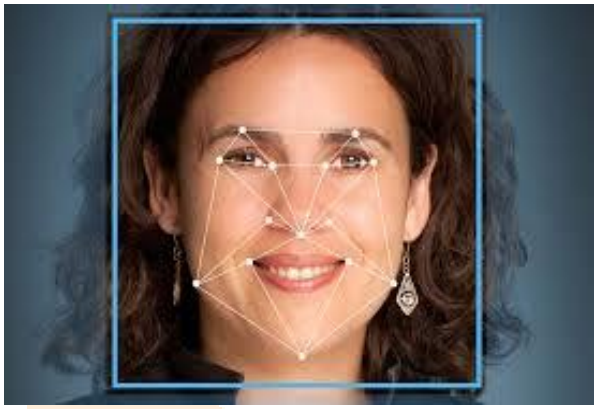
IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

FACE DETECTION USING MATLAB



1.Madhura Pramod Walvekar.2.Yash Dinesh Satpute

3.Keshavi Abhijeet Wakharekar 4. Pranav Deepak Vanik

5.Gauri Satappa Shinde (electronics and Telecommunication Engineering,KITcoEK)

ABSTRACT :

The face is one of the easiest ways to distinguish the individual identity of each other. Face recognition is a personal identification system that uses personal characteristics of a person to identify the person's identity. Human face recognition procedure basically consists of two phases, namely face detection, where this process takes place very rapidly in humans, except under conditions where the object is located at a short distance away, the next is the introduction, which recognize a face as individuals. Stage is then replicated and developed as a model for facial image recognition (face recognition) is one of the much-studied biometrics technology and developed by experts. There are two kinds of methods that are currently popular in developed face recognition pattern namely, Eigenface method and Fisherface method. Facial image recognition Eigenface method is based on the reduction of facedimensional space using Principal Component Analysis (PCA) for facial features. The main purpose of the use of PCA on face recognition using Eigen faces was formed (face space) by finding the eigenvector corresponding to the largest eigenvalue of the face image. The area of this project face detection system with face recognition is Image processing. The software requirements for this project is matlab software

KEYWORDS:

MATLAB, Face detection, Eigen face

1. **INTRODUCTION:**Face detection involves separating image windows into two classes; one containing faces (tarning the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin colour and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height).

The face detection system can be divided into the following steps:

1.Pre-Processing: To reduce the variability in the faces, the images are processed before they are fed into the network. All positive examples that is the face images are obtained by cropping .

2.Classification: Neural networks are implemented to classify the images as faces or nonfaces by training on these examples. We use both our implementation of the neural network and the Matlab neural network toolbox for this task. Different network configurations are experimented with to optimize the results.

3. Localization: The trained neural network is then used to search for faces in an image and if present localize them in a bounding box. Various Feature of Face on which the work has done on:- Position Scale Orientation Illumination

2 LITERATURE SURVEY:

Face detection is a computer technology that determines the location and size of human face in arbitrary (digital) image. The facial features are detected and any other objects like trees, buildings and bodies etc are ignored from the digital image. It can be regarded as a ‗specific' case of object-class detection, where the task is finding the location and sizes of all objects in an image that belong to a given class. Face detection, can be regarded as a more ‗general' case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one). Basically there are two types of approaches to detect facial part in the given image i.e. feature base and image base approach.Feature base approach tries to extract features of the image and match it against the knowledge of the face features. While image base approach tries to get best match between training and testing images.

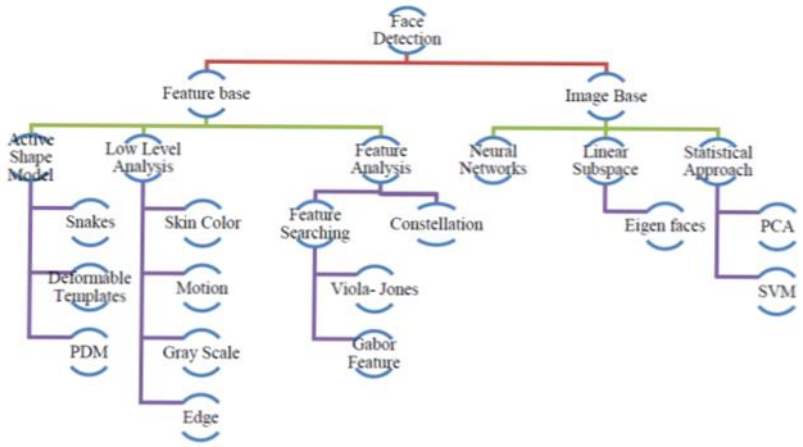


Fig 2.1 detection methods

3.ALGORITHM:

Viola Jones Method:

Paul Viola and Michael Jones presented an approach for object detection which minimizes computation time while achieving high detection accuracy. Paul Viola and Michael Jones [39] proposed a fast and robust method for face detection which is 15 times quicker than any technique at the time of release with 95% accuracy at around 17 fps.The technique relies on the use of simple Haar-like features that are evaluated quickly through the use of a new image representation. Based on the concept of an —Integral Image‖ it generates a large set of features and uses the boosting algorithm AdaBoost to reduce the overcomplete set and the introduction of a degenerative tree of the boosted classifiers provides for robust and fast interferences. The detector is applied in a scanning fashion and used on gray-scale images, the scanned window that is applied can also be scaled, as well as the features evaluated.

Flowchart of algorithm

4.SCRIPT FILE CODE :

```
clc;
[file,path]=uigetfile('*.*','Select an image');
loc=strcat(path,file);
pic=imread(loc);
graypic=rgb2gray(pic);
detectorFace=vision.CascadeObjectDetector('Face');
boundingbox=step(detectorFace,graypic);
detpic=insertObjectAnnotation(pic,'Rectangle',boundingbox,'Face');
imshow(detpic);
%%face detection
cam=webcam("HP TrueVision HD Camera");
im=snapshot(cam);
dete=vision.CascadeObjectDetector();
detectorface.MergeThreshold=200;
imshow(im);
while true
   im=snapshot(cam);
   im2=rgb2gray(im);
   bbox=step(dete,im2);
   pic=insertObjectAnnotation(im,'rectangle',bbox,'face');
   imshow(pic)
end
```

5.HOW DOES IT WORK: The algorithm has four stages:

1. Haar-like features selection.
2. Creating an integral image.
3. Adobos training.
4. Cascading classifiers.

MATLAB has the vision.CascadeObjectDetector system object, which has the viola-jones algorithm used detect faces/objects and it is found in the computer toolbox. The cascadeObjectDetector can be used to detect people's faces, nose, eyes, mouth, and upper body.training vision.CascadeObjectDetector is bundled with several pre-trained classifiers used to detect frontal faces, profile faces, noses, eyes, and upper body.Although, these classifiers are not always sufficient for a particular application. The custom classifier can be trained on an image database through the computer vision toolbox.Run the command below to verify if computer vision toolbox has been installed on your computer.

```
v = ver;
>> setdiff({v.name},'MATLAB')'
```

When this code is executed, it shows all the installed Matlab toolboxes.If you don't have the computer vision toolbox:

- Click on the adds-ons dropdown menu and select get more apps. A new tab is opened in your browser.
- Click on the search box and write computer vision toolbox.
- Once the search is done, select Computer Vision Toolbox Interface for Open CV in MATLAB and click on the download button.
- Once the download is complete, install the package, this will require you to login into your mathwork account. Click on the create account button if you do not have an account.
- Follow the instructions to install the package. For facial or upper body image detection,

1. Create the vision.CascadeObjectDetector object and set its properties.
2. Call the object with the argument as if it were a function.

The image that will be used to detect faces can either be read directly or chosen from the files depending on the user preference. When you add the image directly by using the imshow command, it will display the detected face.When this code is executed, it shows all the installed Matlab toolboxes.

If you don't have the computer vision toolbox:

- Click on the adds-ons dropdown menu and select get more apps. A new tab is opened in your browser.
- Click on the search box and write computer vision toolbox.
- Once the search is done, select Computer Vision Toolbox Interface for Open CV in MATLAB and click on the download button.
- Once the download is complete, install the package, this will require you to login into your mathwork account. Click on the create account button if you do not have an account.
- Follow the instructions to install the package.

For facial or upper body image detection,

1. Create the vision.CascadeObjectDetector object and set its properties.
2. Call the object with the argument as if it were a function.

The image that will be used to detect faces can either be read directly or chosen from the files depending on the user preference. When you add the image directly by using the imshow command, it will display the detected face.

```
Imshow('engineers.jpg')
```

At the moment, you don't need to add the image directly since there could be many images to be used for detection. In this case, you choose the image from the files. For this, you can use the code below:

```
[filename,filepath] = uigetfile('*.*','select an image');
```

You first read the path of the image then read the Image using the imshow command.

```
Filewithpath = strcat(filepath,filename)

Img = imread(filewithpath)
```

The image that we have defined will be saved in the variable Img. Define the face Detector object.

```
faceDetector =  vision.CascadeObjectDetector
```

We then use one of the attributes from the vision.CascadeObjectDetector, i.e. MergeThreshold. This will be used for better detection and accuracy.

```
faceDetector.MergeThreshold = 4;
```

Four is the default value in this case. MergeThreshold can be adjusted depending on the level of accuracy required. The accuracy is higher at the lower values and lowers at the higher value, i.e. the face detection accuracy is higher at 3 than it is at 8.Note that the MergeThreshold is an integer.

```
Boundingboxes = faceDetector(Img)
```

When we execute the code above, it returns an m-by-4 matrix bounding-box. This determines the M bounding boxes containing the detected objects. The detector performs multiscale face detection on the input image, Img. We then introduce conditions for detection. The first condition is when the face is detected. When the face is detected in an image, it should return a bounding box around the detected face.We then insert annotation with the name face in a rectangle. The line width is the thickness of the bounding boxes and it can be changed to any value.Note that the values are in pixel so it determines the thickness of the bounding box in pixel. We then display the image using the imshow command.Here is the code

```
if ~isempty(bboxes)
 Imf = insertObjectAnnotation(img,'rectangular',bboxes,'Faces','linewidth',30);
  imshow(Imf)
  title('detected faces')
```

The second condition is if the face is not detected. We insert text at the location [0 0] which is the x-axis, and the y-axis, with the label ‘no faces detected’. You can change the font size and opacity of the box of this text and display the image. Here is the code

```
else
  position = [0 0];
  label='no face detected';
  Imgn = insertText(Img,position,label, 'fontsize',25,'BoxOpacity',1);
  imshow(Imgn)
end
```

Below is the whole application source code.

```
  Img = imread('building.jpg');
faceDetector = vision.CascadeObjectDetector;
faceDetector.MergeThreshold = 4;
bboxes = faceDetector(img);
if ~isempty(bboxes)
  Imf = insertObjectAnnotation(Img,'rectangle',boundingboxes,'Faces','linewidth',3);
  imshow(Imf)
  title('detected faces')
else
  position = [0 0];
  label='no face detected';
  imgn = insertText(img,position,label, 'fontsize',25,'BoxOpacity',1);
  imshow(imgn)
```

When we run the program, the following image is displayed on our figure window.

This code can also be used to detect eyes, mouths, and/or noses. You only need to change the names, that is replace face with a nose. You would need to vary the MergeThreshold for accuracy.The Viola-jones algorithm is the best algorithm for face object detection in real-time. It is accurate and fast and that makes it efficient when using it in the detection process.

6.IMPROVING FACE DETECTION USING RECONSTRUCTING

Reconstruction cannot be used as a means of face detection in images in near real-time since it would involve resizing the face detection window area and large matrix multiplication, both of which are computationally expensive. However, reconstruction can be used to verify whether potential face locations identified by the deformable template algorithm actually contain a face. If the reconstructed image differs greatly from the face detection window then the window probably does not contain a face. Instead of just identifying a single potential face location, the face detection algorithm can be modified to output many high 'faceness' locations which can be verified using reconstruction. This is especially useful because occasionally the best 'faceness' location found by the deformable template algorithm may not contain the ideal frontal view face pixel area.

potential face locations that have been identified by the face detection system (the best face locations it found on its search) are checked whether they contain a face. If the threshold level (maximum difference between reconstruction and original for the original to be a face) is set correctly this will be an efficient way to detect a face. The deformable template algorithm is fast and can reduce the search space of potential face locations to a handful of positions. These are then checked using reconstruction. The number of locations found by the face detection system can be changed by getting it to output, not just the best face locations it has found so far but any location, which has a 'faceness' value, which for example is, at least 0.9 times the best heuristic value that has been found so far. Then there will be many more potential face locations to be checked using reconstruction. This and similar speed versus accuracy trade-off decisions have to be made keeping in mind the platform on which the system is implemented. Similarly, instead of using reconstruction to check the face detection system's output, the output's correlation with the average face can be checked. The segmented areas with a high correlation probably contains a face. Once again a threshold value will have to be established to classify faces from non-faces. Similar to reconstruction, resizing the segmented area and calculating its correlation with the average face is far too expensive to be used alone for face detection but is suitable for verifying the output of the face detection system.

7.APPLICATIONS: Face detection is a technology that can be applied and implemented in many parts of todays society some areas of applications are :

- **Biometrics** : Using a face as a biometric is proved to be a successful approach since it is the way humans recognize each other.
- **Identification system**: A face could be used to examine if a person exist or not in the list of individuals based on that information the system can allow respectively deny access.
- **Law Enforcement**: Face recognition technology can be used to increase performance of surveillance and law enforcement.
- While face recognition technology is beneficial for verification of personal identity, it does raise privacy issues. Since the technology uses an individual's faceprint, it is often regarded as a breach of one's privacy, safety, and

security. Face recognition using MATLAB can be employed in several cases where security is of utmost concern. From airports and offices to smartphones, facial recognition has become an integral component of many systems and organizations. Although all facial recognition systems use face detection, not all face detection systems are used for facial recognition. Face detection can also be applied for facial motion capture, or the process of electronically converting a human's facial movements into a digital database using cameras or laser scanners. This database can be used to produce realistic computer animation for movies, games or avatars.

Face detection can also be used to auto-focus cameras or to count how many people have entered an area. The technology also has marketing applications -- for example, displaying specific advertisements when a particular face is recognized.

8.Advantages: As a key element in facial imaging applications, such as facial recognition and face analysis, face detection creates various advantages for users, including:

- Improved security. Face detection improves surveillance efforts and helps track down criminals and terrorists. Personal security is also enhanced since there is nothing for hackers to steal or change, such as passwords.
- Easy to integrate. Face detection and facial recognition technology is easy to integrate, and most solutions are compatible with the majority of security software.
- Automated identification. In the past, identification was manually performed by a person; this was inefficient and frequently inaccurate. Face detection allows the identification process to be automated, thus saving time and increasing accuracy.

9Disadvantages: While face detection provides several large benefits to users, it also holds various disadvantages, including:

- Massive data storage burden. The ML technology used in face detection requires powerful data storage that may not be available to all users.
- Detection is vulnerable. While face detection provides more accurate results than manual identification processes, it can also be more easily thrown off by changes in appearance or camera angles.
- A potential breach of privacy. Face detection's ability to help the government track down criminals creates huge benefits; however, the same surveillance can allow the government to observe private citizens. Strict regulations must be set to ensure the technology is used fairly and in compliance with human privacy rights.

10. CONCLUSION: The computational models, which were implemented in this project, were chosen after extensive research, and the successful testing results confirm that the choices made by the researcher were reliable.The system with manual face detection and automatic face recognition did not have a recognition accuracy over 90%, due to the limited number of eigenfaces that were used for the PCA transform. This system was tested under very robust conditions in this experimental study and it is envisaged that real-world performance will be far more accurate.The fully automated frontal view face detection system displayed virtually perfect accuracy and in the researcher's opinion further work need not be conducted in this area. The fully automated face detection and recognition system was not robust enough to achieve a high recognition accuracy. The only reason for this was the face recognition subsystem did not display even a slight degree of invariance to scale, rotation or shift errors of the segmented face image. This was one of the system requirements identified in section 2.3. However, if some sort of further processing, such as an eye detection technique, was implemented to further normalise the segmented face image, performance will increase to levels comparable to the manual face detection and recognition system. Implementing an eye detection technique would be a minor extension to the implemented system and would not require a great deal of additional research.All other implemented systems displayed commendable results and reflect well on the deformable template and Principal Component Analysis strategies.The most suitable real-world applications for face detection and recognition systems are for mugshot matching and surveillance. There are better techniques such as iris or retina recognition and face recognition using the thermal spectrum for user access and user verification applications since these need a very high

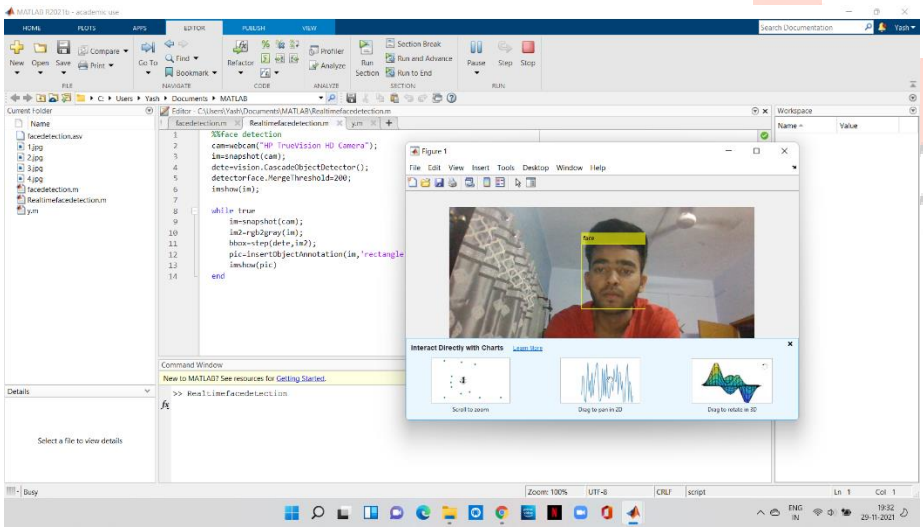
degree of accuracy.The real-time automated pose invariant face detection and recognition system proposed in chapter seven would be ideal for crowd surveillance applications. If such a system were widely implemented its potential for locating and tracking suspects for law enforcement agencies is immense. The implemented fully automated face detection and recognition system (with an eye detection system) could be used for simple surveillance applications such as ATM user security, while the implemented manual face detection and automated recognition system is ideal of mugshot matching. Since controlled conditions are present when mugshots are gathered, the frontal view face recognition scheme should display a recognition accuracy far better than the results, which were obtained in this study, which was conducted under adverse conditions.

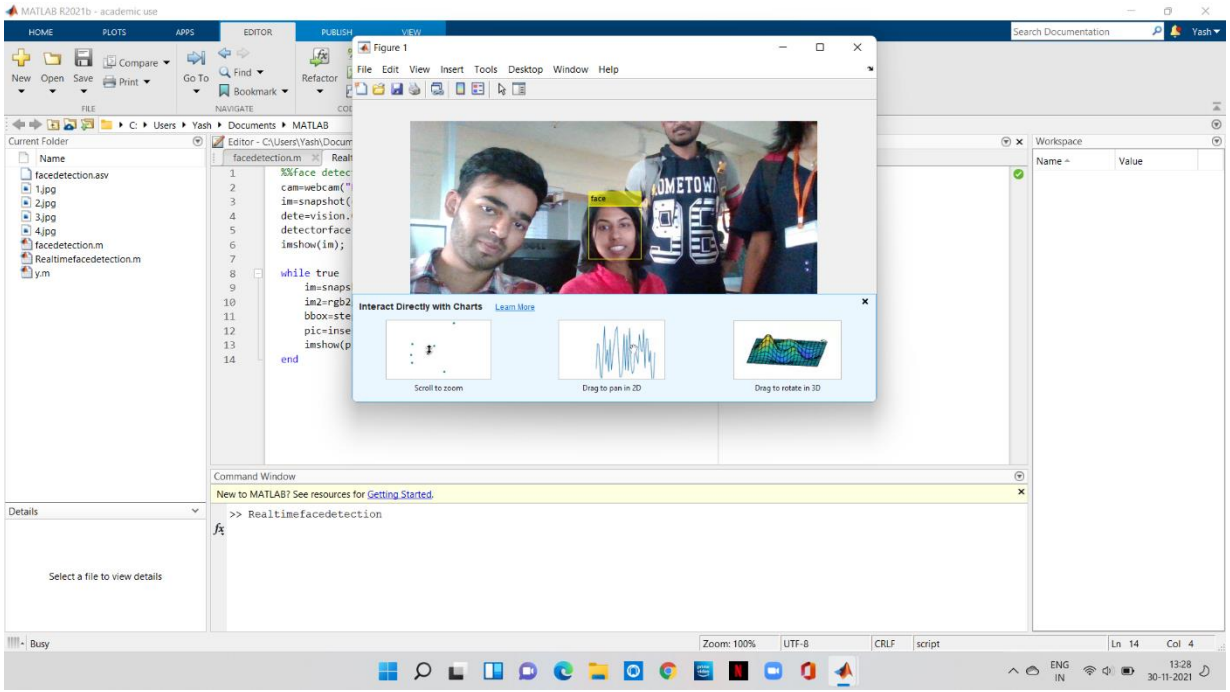
Furthermore, many of the test subjects did not present an expressionless, frontal view to the system. They would probably be more compliant when a 6'5'' policeman is taking their mugshot! In mugshot matching applications, perfect recognition accuracy or an exact match is not a requirement. If a face recognition system can reduce the number of images that a human operator has to search through for a match from 10000 to even a 100, it would be of incredible practical use in law enforcement. The automated vision systems implemented in this thesis did not even approach the performance, nor were they as robust as a human's innate face recognition system. However, they give an insight into what the future may hold in computer vision

REFRENCES:

- https://www.electronicsforu.com/electronics-projects/software-projects-ideas/real-time-face-detection-using-matlab
- https://www.youtube.com/watch?v=R0Cm7zKIJI8
- https://www.section.io/engineering-education/face-detection-matlab/
- https://www.mathworks.com/matlabcentral/fileexchange/13701-face-detection-in-matlab

RESULT :




MATLAB R2021b - academic use
Figure 1
File Edit View Insert Tools Desktop Window Help
face
Interact Directly with Charts
Learn More
Scroll to zoom
Drag to pan in 2D
Drag to rotate in 3D
Command Window
New to MATLAB? See resources for Getting Started.
>> Realtimefacedetection
Select a file to view details