N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU, Belgaum) (AICTE approved, NBA Accredited, ISO 9001:2008 Certified)

NITTE -574 110, Udupi District, KARNATAKA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IMAGE PROCESSING

A MINI PROJECT REPORT

On

"Counting number of persons in the given image"

Submitted by,

Shreyas K Shetty	**4NM18CS181**
Vinyasa VK	**4NM18CS212**

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU, Belgaum) (AICTE approved, NBA Accredited, ISO 9001:2008 Certified)

NITTE -574 110, Udupi District, KARNATAKA

CERTIFICATE

Certified that the project work entitled "Counting number of persons in the given image" is a bonafide work carried out by ***Shreyas K Shetty(4NM18CS181), Vinyasa VK (4NM18CS212)*** *in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belgaum during the year 2021-2022.*

Signature of Guide:

Mrs. Shilpa M Karegoudar

Assistant Professor Gd-I

Dept of CSE, NMAMIT Nitte

TABLE OF CONTENTS

Abstract

Counting a number of persons in a given image automatically is a challenging problem which has received much attention during recent years due to its many applications in different fields. Face detection is one of those challenging problems and up to date, there is no technique that provides a robust solution to all situations. Automatic face detection had been studied worldwide in last 10 years, which has become the very active research area in computer vision and pattern recognition. In this project we will apply Viola Jones Algorithm for human face detection. Once human faces detected we will count the number of persons in a given image.

Introduction

Object detection and tracking are important in many computer vision applications, including activity recognition, automotive safety and surveillance. Presented here is a face detection using MATLAB system that can detect a human face then it will count the number of persons in the given image. Face detection is an easy and simple task for humans, but not so for computers. It has been regarded as the most complex and challenging problem in the field of computer vision due to large intra-class variations caused by the changes in facial appearance, lighting and expression. Such variations result in the face distribution to be highly nonlinear and complex in any space that is linear to the original image space.

Face detection is the process of identifying one or more human faces in images or videos. It plays an important part in many biometric, security and surveillance systems, as well as image and video indexing systems. Although a trivial task for the human brain, face recognition has proved to be extremely difficult to imitate artificially, since although commonalities do exist between aces, they vary considerably in terms of age, skin, color and gender. The problem is further complicated by differing image qualities, facial expressions, facial furniture, background, and illumination conditions. Since skin color in humans varies by individual, research has revealed that intensity rather than chrominance is the main distinguishing characteristic. The recognition stage typically uses an intensity (grayscale) representation of the image.

This face detection using MATLAB program can be used to detect a face. The goal of this project is to detect and locate human faces in a color image. The objective was to design and implement a face detector in MATLAB that will detect human faces in an image similar to the training images.

Implementation

Special functions (syntax and explanation):

This project is implemented using Viola Jones face detection algorithm. Generally, viola jones face detection algorithm has three critical steps, including feature extraction, boosting and multi-scale detection.

Feature extraction:

It is obvious that feature is very significant to any object detection algorithm. Basically, there are a lot of features, such as eyes, nose, the topology of eye and nose, can be used for face detection. In viola jones face detection, a very simple and straightforward feature has been used. Figure 1 shows four different features in viola jones algorithm. Each feature can be obtained by subtracting white areas from the black areas. Here, the area means the summation of all the pixels' gray value within the rectangle. Aiming at calculating these features, a special representation named as integral image has been used. Specifically, integral image of a location is the sum of the pixel values above and to the left of, inclusively. Figure 2 shows one fast way to compute the pixel sum within a rectangle. In Figure 2, the value of integral image at location '1' is the sum of pixels in rectangle A; the value at location '2' is the sum of pixels in rectangle A and B; the value at location '3' is the sum of pixels in rectangle A and C; the value at location '4' is the sum of pixels in rectangle A, B, C and D. Based on this information, the sum of pixels in rectangle D can be easily got from. Therefore, the sum of pixels of any rectangle located at any position can be obtained following this principle very efficiently.

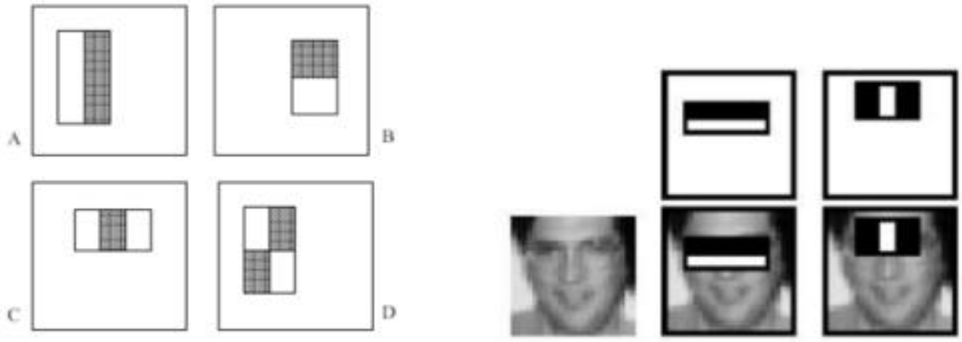

Figure 1: four basic features in viola jones algorithm

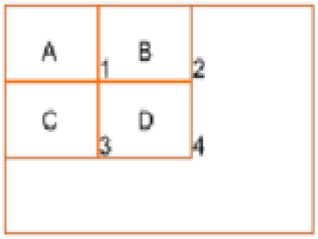

Figure 2: the calculation of pixel sum within a rectangle

Boosting:

The definition of boosting in viola jones face detection algorithm is the combination of several weak classifiers. This boosting idea makes the process of learning to be simple and efficient. Specifically, the boosting works as follows:

1. given a dataset, learn a single and simple classifier first and check where it makes errors;
2. reweight the dataset and give the data where it made errors a higher weight;
3. learn the second simple classifier based on the reweighted dataset;
4. combine the first and second classifier and reweight data where they make errors;
5. keep learning until we get T classifiers;
6. the final classifier will be the combination of all those T classifiers.

Figure 3 shows detail of the principle of boosting.

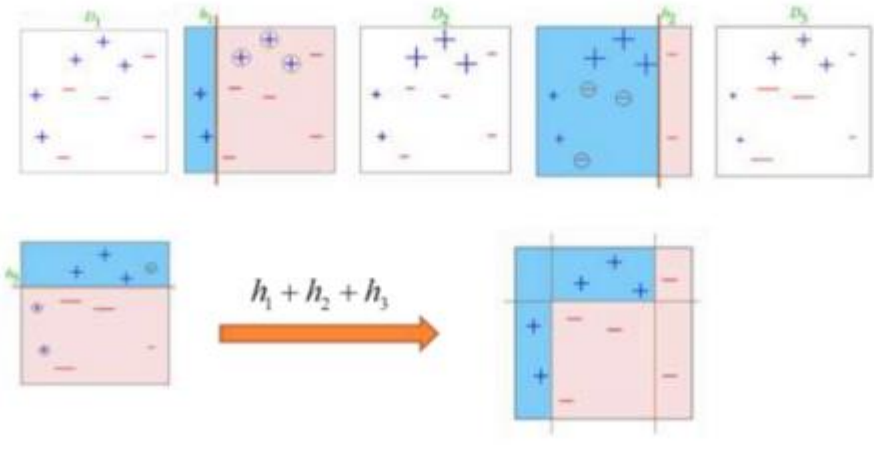

Figure 3: the process of boosting with 3 simple classifiers

Multi-scales detection algorithm:

Another important step in viola jones face detection algorithm is multi-scale detection. It is obvious that we have no idea with the size of face in an image before doing face detection. Therefore, multi-scale detection should be adopted to guarantee that faces with any size can be detected. Features should be calculated at all different scales since the learning and testing are only based on the rectangle features. In this project, we used the scales by the factor of 1.2. In each iteration, the width and height of rectangle will increase to 1.2 times of previous one.

Camshift:

CAMshift is called Continuously Adaptive Mean Shift based on the mean shift algorithm. CAMshift uses the Hue channel to trace objects since by using the Hue channel based on HSV color model, objects with different colors can be recognized. Based on the color information, CAMshift tracks objects faster and consumes relatively little CPU resources. Lower computing resource requirement enable CAMshift to become a one of real-time face tracking algorithms The procedure of CAMshift shown in the Figure 4 includes two important parts which are the histogram and the search of peak probability. The first step of Camshift is to obtain the histogram of tracked object. In the second step, the next frame will be converted into a map of skin color probability based on the histogram. In the third step, the peak probability center will be found based on zero moment and first moment. Finally, CAMshift will check whether it converges. If it is no, it will go to the third step, otherwise, it gets the position of the tracked object in this frame and fetches the next frame to track the object continuously.

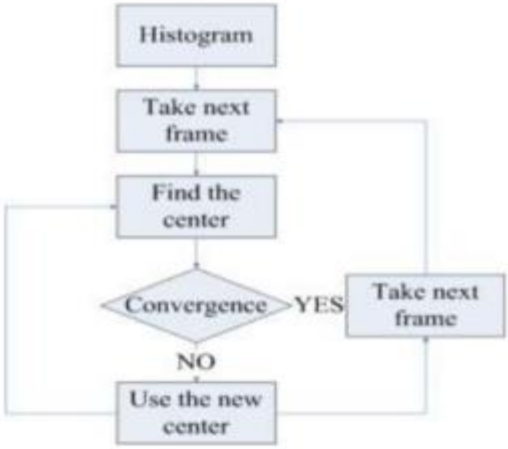

Figure 4 the procedure of CAMshift

Histogram:

The histogram is the tracked object's color probability map. Because in the project we focus on the face tracking, the hue channel of the face shown in the Figure 6. is used to illustrate what is the histogram. In the first step, the whole area of tracked object is scanned and the map which record how many pixels have a certain hue value in the tracked object area is built. And then, CAMshift finds out the peak the number of pixels in a certain hue value and normalizes the map into skin color probability map or histogram shown in the Figure 5. The horizontal bar of the Figure 5 is hue value and the vertical bar is the skin probability. According to the histogram, the popular color on the face is revealed.

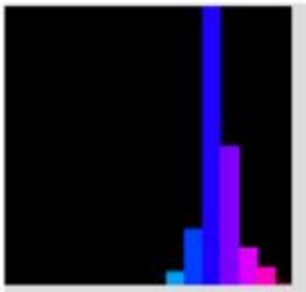

Figure 5 histogram

Figure 6 H channel of the face

Finding the center:

Based on histogram, the skin probability of the next frame can be calculated in this way. In the first, the next frame is converted into H channel which represents the color information of the tracked object (from Figure 7 to Figure 8). And then looking up the histogram, we will get the skin probability for each pixel shown in the Figure 9 according to its hue value. The skin probability of the whole picture is obtained for the next step to find out the center of the face.

Figure7 the next frame

Figure 8 the hue channel of the next frame

Figure 9 the skin probability

Based on the skin probability of the next frame, the peak probability as the center of the face is estimated by the zero moment and first moment. The zero moment and first moment are calculated by the equations shown below.

$$M_{00} = \sum_{x,y} I(x,y) \quad M_{10} = \sum_{x,y} xI(x,y) \quad M_{01} = \sum_{x,y} yI(x,y)$$

In this way, the center of face can be estimated by the following equations.

$$x_c = \frac{M_{10}}{M_{00}} \qquad y_c = \frac{M_{01}}{M_{00}}$$

Finally, the convergence of the center of the face is checked. If the center of the face is very close to the old center, the convergence is reached; otherwise, all the equations should be calculated again based on the new center.

Source code with explanation:

clear all; close all; clc;

clc clears all the text from the Command Window, resulting in a % clear screen. After running clc, we cannot use the scroll bar in the Command Window to see previously displayed text. We can, however, use the up-arrow key in the Command Window to recall statements from the command history. To clear all variables from the current workspace, use clear all, close all closes all figures whose handles are visible

% This is used to read an input image
A = imread('faces.png');

% Create a cascade detector object. The cascade object detector uses the % Viola-Jones algorithm to detect people's faces,
faceDetector = vision.CascadeObjectDetector();

% This will Perform face detection on the input image to get bounding
% boxes
bbox = step(faceDetector, A);

```
% It is used to draw the returned bounding box around the detected face
B = insertObjectAnnotation(A, 'rectangle', bbox, 'Face');
figure, imshow(B), title('Detected faces');

% Finally this will count the number of Persons in the given image
str = num2str(size(bbox, 1));
str = strcat('Number of Persons in the given  = ', str);
disp(str);
```

Screenshot with caption

```
clear all;
close all;
clc;

% Read an input image
A = imread('cricket-team.jpg');

% Create a cascade detector object
faceDetector = vision.CascadeObjectDetector();

% Perform face detection on the input image to get bounding boxes
bbox = step(faceDetector, A);

% Draw the returned bounding box around the detected persons
B = insertObjectAnnotation(A, 'rectangle', bbox, 'Face');

figure, imshow(B), title('Detected Persons');

% Print number of detected persons
str = num2str(size(bbox, 1));
str = strcat('Number of Persons in the given image= ', str);
disp(str);
```

Fig 1. Source code screenshot

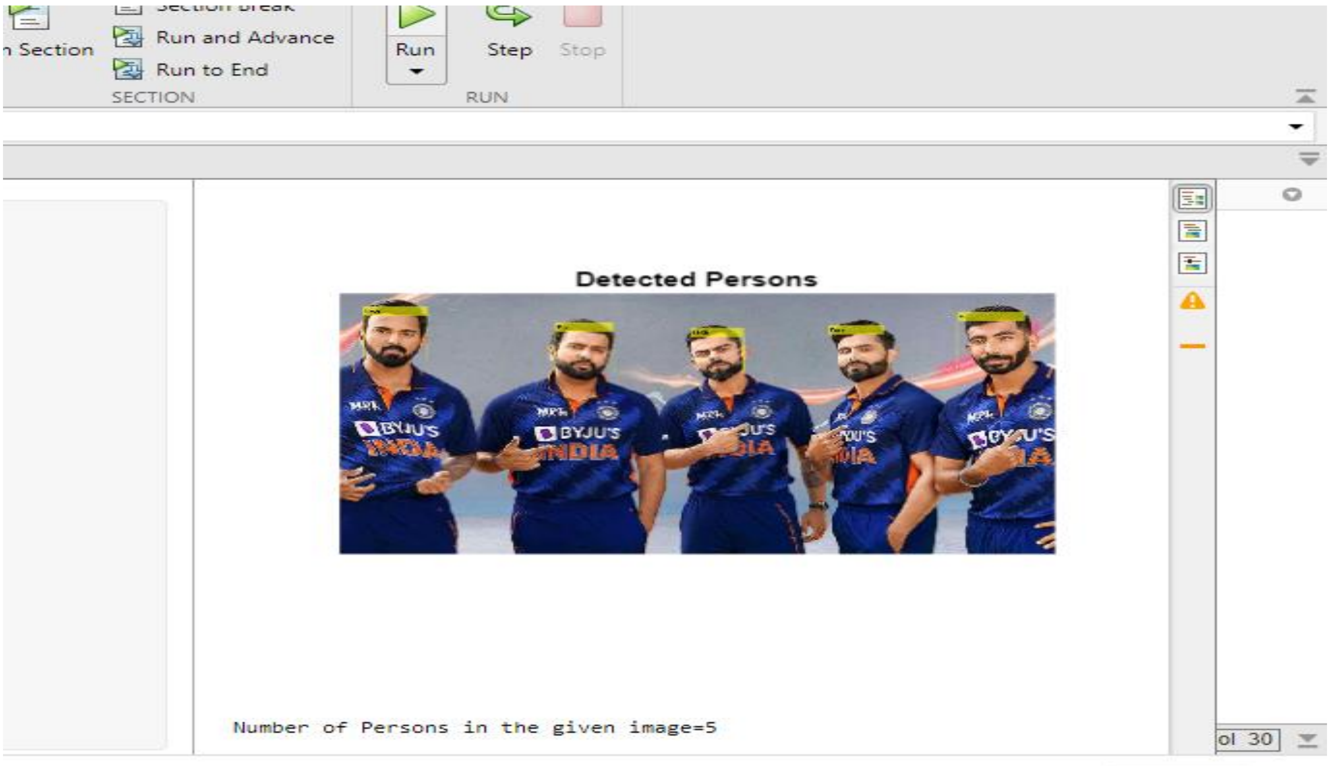

Fig 2. Output 1 detecting 5 persons in an image

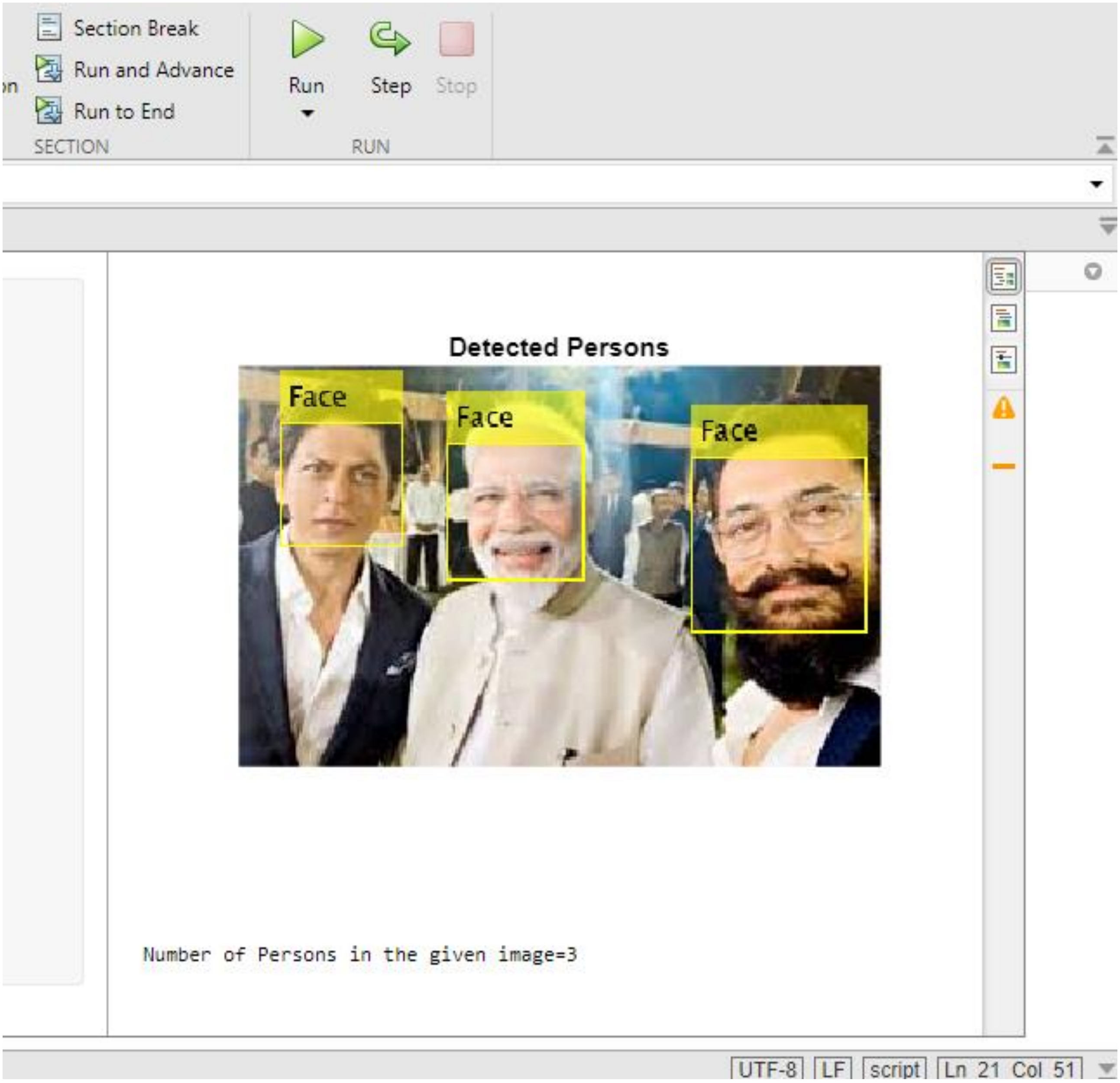

Fig 3. Output 2 detecting 4 persons in an image

Conclusion:

Our project is about introducing a system that uses image processing techniques to solve the problem of crowd management by counting the number of people in an image. We have proposed a counting system, which counts the number of people. The system detects the faces at every interval using Viola-Jones detector. The accuracy of the detection is 77% for low dense, 82% for medium dense and 79% for high density crowd in an image.

References

- https://en.wikipedia.org/wiki/MATLAB
- https://towardsdatascience.com/the-intuition-behind-facial-detection-the-viola-jones-algorithm-29d9106b6999
- https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework
- https://www.researchgate.net/publication/228437591_Real-time_people_counting_system_using_a_single_video_camera