



Vidyavardhini's College of Engineering & Technology
Department of Computer Engineering
Academic Year : 2023-24

Nayan Biramane

Roll no:-21

Aim: To study Detecting and Recognizing Faces

Objective: To Conceptualizing Haar Cascades Getting Haar cascade data
Using Open CV to Perform face detections performing face detection on still
Images

Theory:-

Conceptualizing Haar Cascades:-

Photographic images, even from a webcam, may contain a lot of detail for our (human) viewing pleasure. However, image detail tends to be unstable with respect to variations in lighting, viewing angle, viewing distance, camera shake, and digital noise. Moreover, even real differences in physical detail might not interest us for classification. Joseph Howse, one of this book's authors, was taught in school that no two snowflakes look alike under a microscope. Fortunately, as a Canadian child, he had already learned how to recognize snowflakes without a microscope, as the similarities are more obvious in bulk.

Getting Haar Cascade Data:-

In order to do object recognition/detection with cascade files, you first need cascade files. For the extremely popular tasks, these already exist. Detecting things like faces, cars, smiles, eyes, and license plates for example are all pretty prevalent. You can use Google to find various Haar Cascades of things you may want to detect. You shouldn't have too much trouble finding the aforementioned types. We will use a Face cascade and Eye cascade. You can find a few more at the root



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Academic Year : 2023-24

directory of Haar cascades. Note the license for using/distributing these Haar Cascades.

Using Open CV to perform Face Detection:

The face recognition is a technique to identify or verify the face from the digital images or video frame. A human can quickly identify the faces without much effort. It is an effortless task for us, but it is a difficult task for a computer. There are various complexities, such as low resolution, occlusion, illumination variations, etc. These factors highly affect the accuracy of the computer to recognize the face more effectively. First, it is necessary to understand the difference between face detection and face recognition.

Face Detection: The face detection is generally considered as finding the faces (location and size) in an image and probably extract them to be used by the face detection algorithm.

Face Recognition: The face recognition algorithm is used in finding features that are uniquely described in the image. The facial image is already extracted, cropped, resized, and usually converted in the grayscale.

There are various algorithms of face detection and face recognition. Here we will learn about face detection using the HAAR cascade algorithm.

Performing Face detection on a still image:

The first and most basic way to perform face detection is to load an image and detect faces in it. To make the result visually meaningful, we will draw rectangles around faces in the original image. Remembering that the face detector is designed for upright, frontal faces, we will use an image of a row of people, specifically woodcutters, standing shoulder-to-shoulder and facing the viewer.



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering
Academic Year : 2023-24

Code:-

```
import cv2

imagePath = '/content/a.webp'
img = cv2.imread(imagePath)
img.shape
(4000, 2667, 3)
gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray_image.shape
(4000, 2667)
face_classifier = cv2.CascadeClassifier(cv2.data.harcascades +
"haarcascade_frontalface_default.xml")
face = face_classifier.detectMultiScale(gray_image, scaleFactor=1.1,
minNeighbors=5, minSize=(40, 40))
for (x, y, w, h) in face:
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 4)
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
import matplotlib.pyplot as plt
plt.figure(figsize=(20,10))
plt.imshow(img_rgb)
plt.axis('off')
```



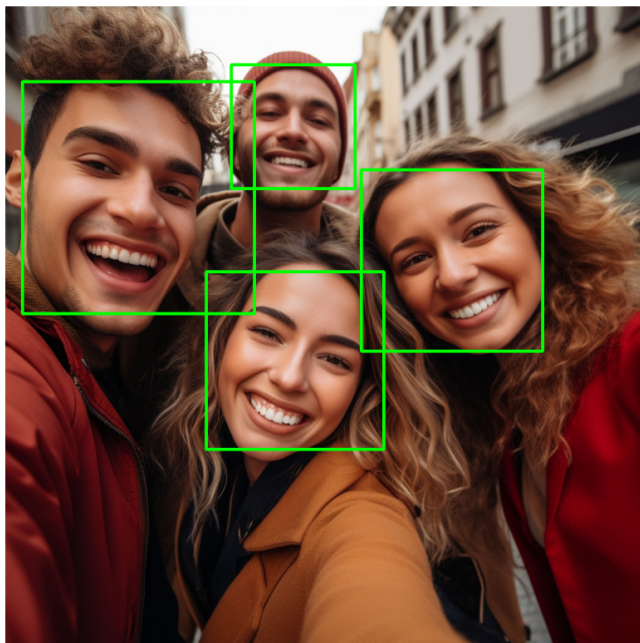
Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering
Academic Year : 2023-24

Input:-



Output:-





Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Academic Year : 2023-24

Conclusion:-

In this experiment we have learned how to detect faces from an image using OpenCV library which is available in python. The OpenCV package comes with pre-trained models for face detection, which means that we don't have to train an algorithm from scratch. More specifically, the library employs a machine learning approach called Haar cascade to identify objects in visual data.

Haar Cascade is a feature-based object detection algorithm to detect objects from images. A cascade function is trained on lots of positive and negative images for detection. The algorithm does not require extensive computation and can run in real-time. We can train our own cascade function for custom objects like animals, cars, bikes, etc.

Haar Cascade can't be used for face recognition since it only identifies the matching shape and size. Haar cascade method can be useful in detecting faces in image but it still needs some major improvements for optimal results.