Experiment No.6
To study Detecting and Recognizing Faces
Date of Performance:21/08/2023
Date of Submission: 04/09/2023

Aim: To study Detecting and Recognizing Faces

Objective: To Conceptualizing Haar Cascades Getting Haar cascade data Using Open CVtoPerformfacedetections performing face detection on still images

Theory:

Conceptualizing Haar Cascades -

Haar Cascades are a machine learning technique used for object detection in images. Theyare particularly valuable for real-time applications due to their speed and efficiency. Theconcept involves breaking down the detection process into stages. Each stage consistsofasimple binary classifier trained to recognize specific image features, such as edgesortextures, using Haar-like features.

During detection, a sliding window moves over the image, and at each stage, a classifierassesses whether the region within the window matches the object's features. If aregiondoesn't match, it's quickly rejected, reducing computational load. This "cascade" effect allows for fast rejection of non-object regions.

Haar Cascades are adaptable, allowing you to fine-tune parameters for accuracyandspeedtrade-offs. They are commonly used in face detection, but their flexibility makes themsuitable for various object recognition tasks.

Getting Haar Cascade Data-

1. To obtain Haar Cascade data for object detection, follow these steps:
2. **Choose or Create a Positive Dataset:**Collect a set of images containingtheobjectyouwant to detect.Create a text file listing the file paths of these positive images alongwiththeir corresponding object annotations (bounding boxes).
3. **Choose or Create a Negative Dataset:**Collect a set of images that do not containtheobject of interest.Create a text file listing the file paths of these negative images.
4. **Install OpenCV:**Install OpenCV, an open-source computer vision library, whichincludes tools for creating Haar Cascades.
5. **Prepare Positive and Negative Samples:**Use OpenCV's tools to create positivesamplesand negative samples from your image datasets.Generate positive sampleswithannotations using the opencv_createsamples command-line tool.Create a text filelistingthe positive samples and negative samples.

6. **Train the Cascade Classifier:**Use the opencv_traincascade command-linetool totrainthe Haar Cascade classifier.Specify parameters like the number of stages, thesizeofpositive and negative samples, and other training options.The tool will runthroughmultiple iterations to improve the classifier's accuracy.

7. **Evaluate and Test the Cascade Classifier:**After training, evaluate the performanceofyour Haar Cascade classifier on test images.Adjust parameters and retrainif necessarytoachieve the desired detection accuracy and false positive rate.

8. **Use the Trained Cascade Classifier:**Once satisfied with the performance, youcanusethe trained Haar Cascade XML file for object detection in your applications.OpenCVprovides functions to load and apply the trained cascade classifier to detect objectsinimages or video streams.

Using Open CV to perform Face Detection:

To perform face detection using OpenCV, first, load a pre-trained Haar Cascadeclassifierfor faces using cv2.CascadeClassifier(). Then, read an image or capture framesfromacamera feed. Use the detectMultiScale() method to locate faces within the imageorframe.Adjust the scaleFactor and minNeighbors parameters to control detectionsensitivityandaccuracy. Draw rectangles around the detected faces using cv2.rectangle(). Finally, displayor save the processed image/frame to visualize or analyze the detected faces. Thisprocessallows for quick and efficient face detection in images or real-time video usingOpenCV'scomputer vision capabilities.

Performing Face detection on a still image:

Performing face detection on a still image involves using computer vision techniquestoidentify and locate human faces within the image. Here's how it's done in a nutshell:

1. **Image Input:** Begin by loading the still image you want to analyze.

2. **Haar Cascade or Deep Learning Model:** You can choose between classicHaarCascade classifiers or more modern deep learning models like Convolutional NeuralNetworks (CNNs). Haar Cascades work by sliding a window over the imageandapplying a series of classifiers to detect facial features. Deep learning models, likethepopular OpenCV DNN module with pre-trained models, provide accurate facedetectionwithout the need for handcrafted features.

3. **Detection:** Apply the chosen model to the image. Haar Cascades will returnboundingboxes around detected faces, while deep learning models may also provide facial landmarks for further analysis.

4. **Visualize and Post-process:** Draw bounding boxes or overlay facial landmarksontheimage to highlight the detected faces. You can also perform additional tasks likegenderor age estimation if required.
5. **Output:** The result is an image with annotated faces or a list of coordinates representingthe location of detected faces in the image.

Introduction

Discover object detection with the Haar Cascade algorithm using OpenCV. Learnhowtoemploy this classic method for detecting objects in images and videos. Exploretheunderlying principles, step-by-step implementation, and real-world applications. Fromfacialrecognition to vehicle detection, grasp the essence of Haar Cascade and OpenCV'sroleinrevolutionizing computer vision. Whether you're a novice or an expert, this articlewill equipyou with the skills to harness the potential of object detection in your projects.

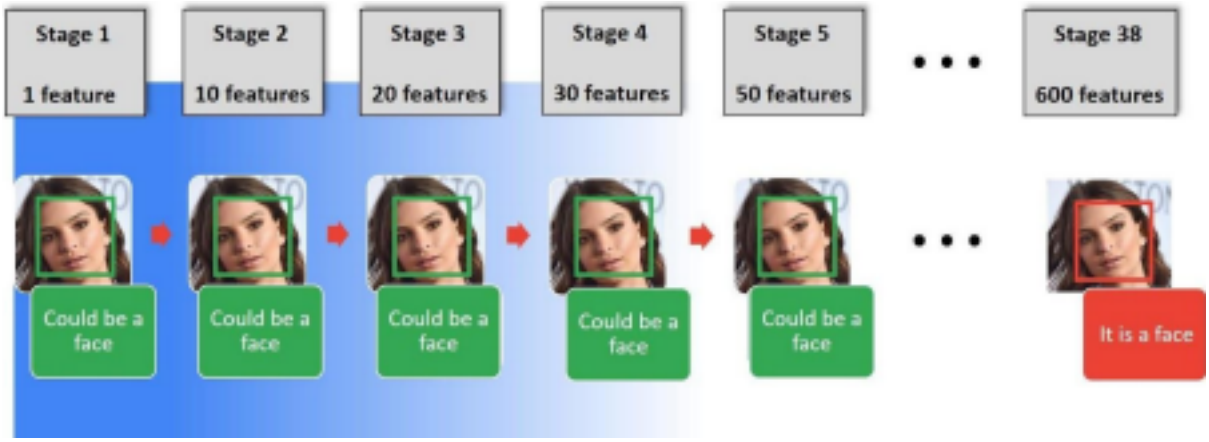


Table of contents

Why Use Haar Cascade Algorithm for Object Detection?

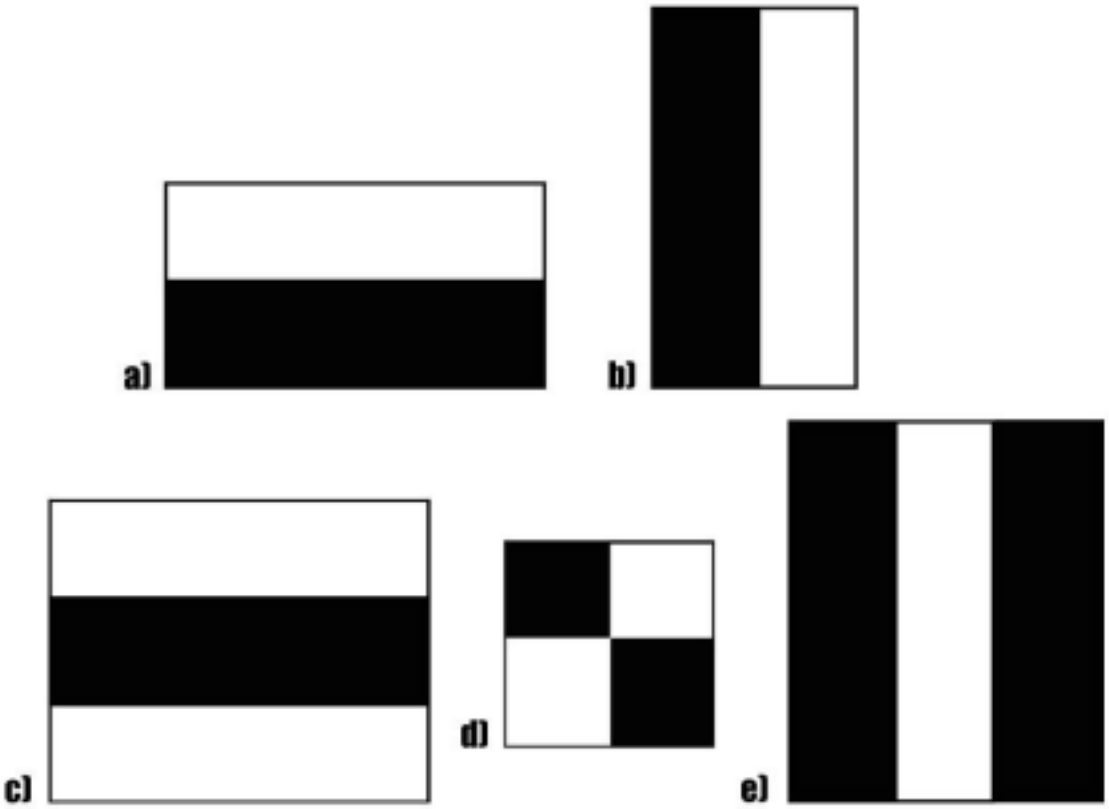
Identifying a custom object in an image is known as object detection. This taskcanbedoneusing several techniques, but we will use the haar cascade, the simplest methodtoperformobject detection in this article.

What is Haar Cascade Algorithm?

Haar cascade is an algorithm that can detect objects in images, irrespective of their scaleinimage and location.

This algorithm is not so complex and can run in real-time. We can traina haar-cascadedetector to detect various objects like cars, bikes, buildings, fruits, etc.

Haar cascade uses the cascading window, and it tries to compute features ineverywindowand classify whether it could be an object.



Haar cascade works as a classifier. It classifies positive data points →that are part of our detectedobjectand negative data points → that don't contain our object.

- Haar cascades are fast and can work well in real-time.
- Haar cascade is not as accurate as modern object detection techniques are. • Haar cascade

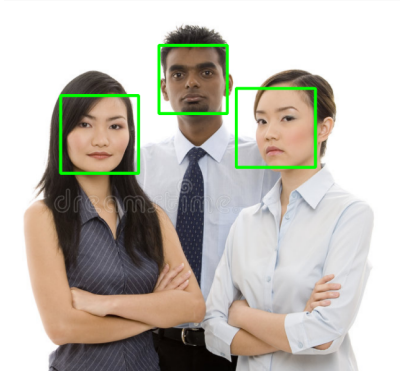
has a downside. It predicts many false positives.

- Simple to implement, less computing power required.

Code:

```
pip install opencv-python

Requirement already satisfied: opencv-python in /usr/local/lib/python3.10/dist-packages (4.8.0.76) Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.10/dist-packages (fromopencv-python)(1.23.5)

import cv2

imagePath = 'input_image.jpg'

img = cv2.imread('mv.jpg')

img.shape

(225, 225, 3)

gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

gray_image.shape

(225, 225)

face_classifier = cv2.CascadeClassifier(

   cv2.data.haarcascades + "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')

(-0.5, 224.5, 224.5, -0.5)
```

Output:



Conclusion :

In conclusion, the study of detecting and recognizing faces is a critical field within computer vision and artificial intelligence. It involves developing algorithms and techniques to locate and identify human faces within images or video streams. This technology finds wide-ranging applications, from security and surveillance to biometric authentication and entertainment. Achieving robust and accurate face detection and recognition remains an ongoing challenge due to varying lighting conditions, poses, and occlusions. Nonetheless, continuous research and advancements in deep learning and Haar Cascadeshave significantly improved the reliability and performance of these systems, making them increasingly valuable in today's digital world.