PANTECH SOLUTIONS®
Technology Beyond the Dreams

AI Master Class series – Day 5

Face Detection & Tracking

Day-5 Agenda.

01.

Face Detection

Face detection & its application

02.

Haar Cascade FrontalFace

Overview about the algorithm

03.

Face detection & Tracking

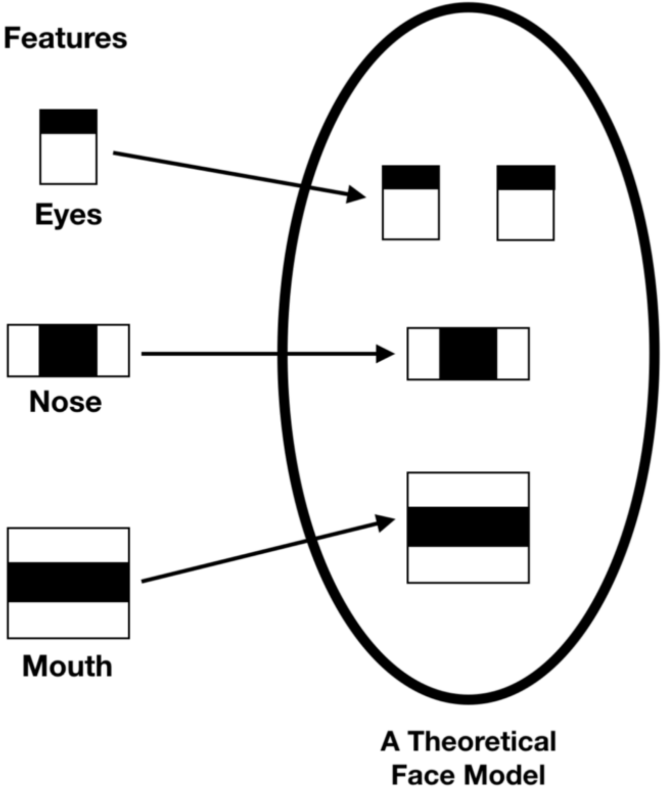
Face detection &
Creating database for
face recogniton

Face Detection.

- Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images.
- With face detection, you can get the information you need to perform tasks like embellishing selfies and portraits, or generating avatars from a user's photo. Because ML Kit can perform face detection in real time, you can use it in applications like video chat or games that respond to the player's expressions

Haar Cascade FrontalFace Algorithm.

- It is based on the Haar Wavelet technique to analyze pixels in the image into squares by function.
- This uses machine learning techniques to get a high degree of accuracy from what is called “training data”.
- This uses “integral image” concepts to compute the “features” detected.
- Haar Cascades use the Adaboost learning algorithm which selects a small number of important features from a large set to give an efficient result of classifiers.



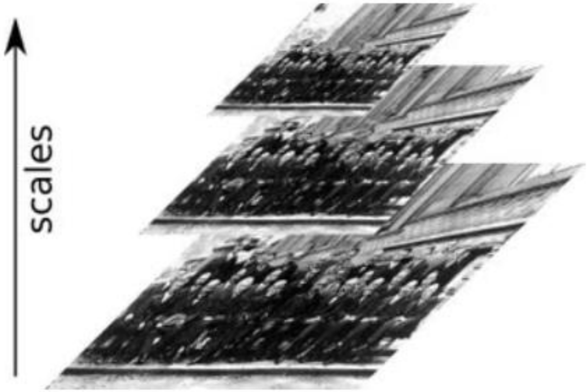
detectMultiScale.

#faces = face_cascade.detectMultiScale(src, scalefactor,minNeighbors)

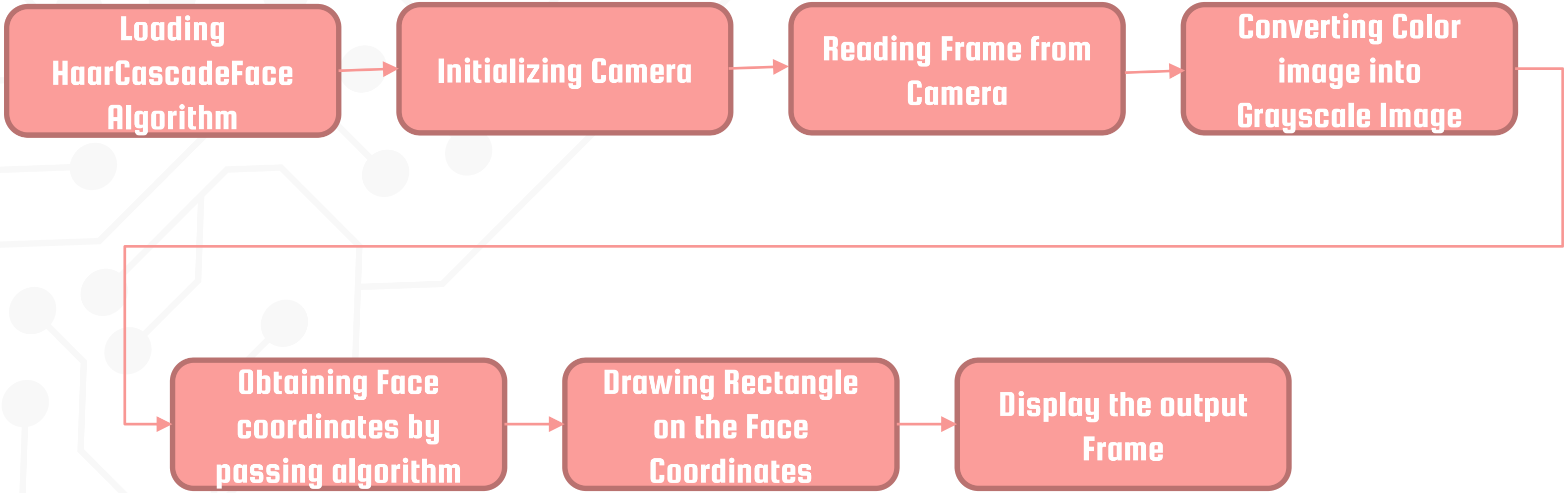
faces = face_cascade.detectMultiScale(gray, 1.3, 4)

scaleFactor — Parameter specifying how much the image size is reduced at each image scale.

minNeighbors — Parameter specifying how many neighbors each candidate rectangle should have to retain it.



Block Diagram – Workflow of Face Detection.



Loading HaarCascadeFace Algorithm

Initializing Camera

Reading Frame from Camera

Converting Color image into Grayscale Image

Obtaining Face coordinates by passing algorithm

Drawing Rectangle on the Face Coordinates

Display the output Frame

Practical session

Face Detect.

```
import cv2
haar_file = 'haarcascade_frontalface_default.xml'
face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0)

while True:
   (_, im) = webcam.read()
   gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
   faces = face_cascade.detectMultiScale(gray, 1.3, 4)
   for (x,y,w,h) in faces:
      cv2.rectangle(im,(x,y),(x+w,y+h),(255,0,0),2)
   cv2.imshow('FaceDetection', im)
   key = cv2.waitKey(10)
   if key == 27:
      break
webcam.release()
cv2.destroyAllWindows()
```

Q & A session

Thanks!

Connect with me on **LinkedIn:**
link in Description

Product & Project:
www.pantechsolutions.net

Course:
Learn.pantechsolutions.net

Tomorrow session

Object tracking based on colour