

Computer Vision Course Project

Face Detection with Python

Galal, Mido, Mona, Omar

May 11, 2018

1 Problem Definition and Motivation

- Problem Definition
- Motivation

2 Algorithms Used

3 Results and Discussions

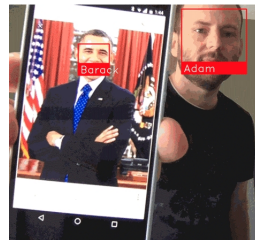
4 Limitations

Problem Definition and Motivation

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images.

The basic benefit of the technology is to detect human faces, regardless of identity of the person.

But a more advanced version of it, coupled with machine learning and training on pre-acquired images, it can detect a person's face and who really is him.

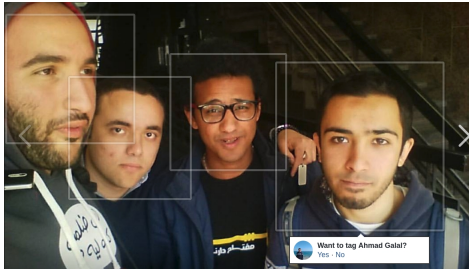


Problem Definition and Motivation



We deal with face detection in our daily life. One simple example is when you're trying to photograph your friends and then your phone camera detects faces and put green rectangles around them.

Problem Definition and Motivation



Another more common example is when you upload the photo you just captured in the previous slide to Facebook, and you find it putting the same rectangles around each face **plus** suggesting a tag for each one.

Algorithms Used

Goal:

Using Haar Cascade Classifiers to detect faces, eyes, smiles, ...etc

Intro:

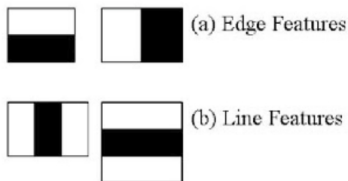
Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Algorithms Used

How it works in the background:

Now we need to extract features from the training images. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

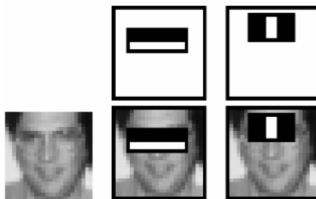
All possible sizes and locations of each kernel are used to calculate lots of features. For each feature calculation, we need to find the sum of the pixels under white and black rectangles.



Algorithms Used

We apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. We select the features that most accurately classify the face and non-face images.

The final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others, they form a strong classifier.



Algorithms Used

Most of any image is a non-face region. So it is a better idea to have a simple method to check if a window is not a face region then we discard it and never process it to save time.

For this they introduced the concept of **Cascade of Classifiers**. Instead of applying all features on a window, the features are grouped into different stages of classifiers and applied one-by-one. If a window fails at a stage, it won't continue to the remaining feature stages. The window which passes all stages is considered a face region.

Algorithms Used

How we interface this in OpenCV:

OpenCV comes with a trainer as well as detector. If you want to train your own classifier for any specific object like a Pagani Zonda or Mido Khalaf for example. you can use OpenCV to create one.

Speaking of the detector, OpenCV already contains many pre-trained classifiers for face, eyes, smiles, ...etc each in an xml file.

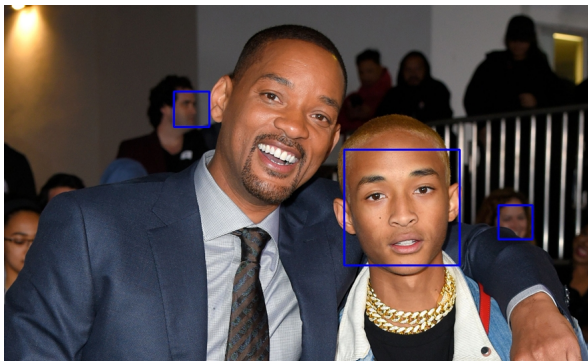
Algorithms Used

The OpenCV Algorithm in a nutshell:

- 1- load the xml files for the objects you want to detect.
- 2- read the image you want to apply the classifier on, or use the video feed from your webcam.
- 3- apply the face detector on the gray-scale version of the image.
- 4- apply the eye and smile detector inside the rectangle of the detected faces.
- 5- plot rectangles of unique colours on every feature you detect.

Results and Discussions

Face Detection



Here we can see that it couldn't detect **Will's** face because it is a little bit tilted

Results and Discussions

Face and Eye Detection



Here, it couldn't detect the RedBull engineer left eye because it is closed.

Results and Discussions

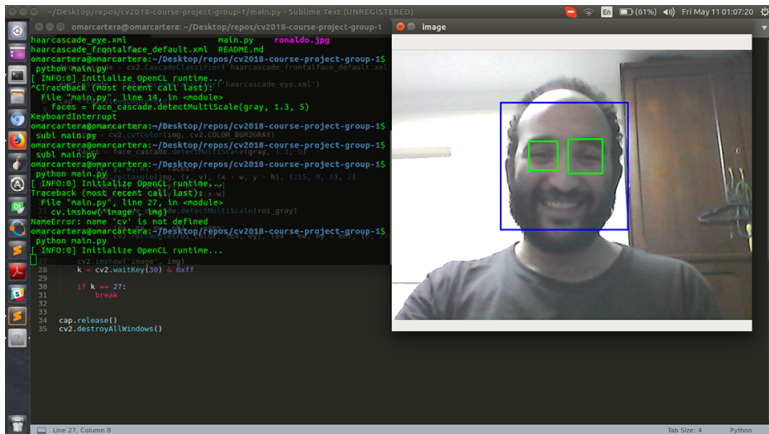
Face and Smile Detection



It is flawless in detecting that smile because it is very obvious.

Results and Discussions

Face and Eye Detection from a Video



Limitations

The limitations we faced were:

- 1- We are using Frontal Face detection, so it will fail to detect side views.
- 2- The detector settings don't always work perfectly with each image, every time you need to tune the **detectMultiScale** *ScaleFactor* and *MinNeighbors*, and then it works well.

THANKS!