

FaceRecognition

December 22, 2017

1 Face Recognition

```
In [41]: import cv2
```

We load the cascade for the face and eyes.

```
In [42]: face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
         eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
```

We create a function that takes as input the image in black and white (gray) and the original image (frame), and that will return the same image with the detector rectangles.

We apply the detectMultiScale method from the face cascade to locate one or several faces in the image.

For each detected face:

- We paint a rectangle around the face.
- We get the region of interest in the black and white image.
- We get the region of interest in the colored image.
- We apply the detectMultiScale method to locate one or several eyes in the image.

For each detected eye:

- We paint a rectangle around the eyes, but inside the referential of the face.

We return the image with the detector rectangles.

```
In [43]: def detect(gray, frame):
         faces = face_cascade.detectMultiScale(gray, 1.3, 5)
         for (x, y, w, h) in faces:
             cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
             roi_gray = gray[y:y+h, x:x+w]
             roi_color = frame[y:y+h, x:x+w]
             eyes = eye_cascade.detectMultiScale(roi_gray, 1.1, 3)
             for (ex, ey, ew, eh) in eyes:
                 cv2.rectangle(roi_color, (ex, ey), (ex+ew, ey+eh), (0, 255, 0), 2)
         return frame
```

We turn the webcam on.

```
In [44]: video_capture = cv2.VideoCapture(0)
```

We repeat infinitely (until break):

- We get the last frame.
- We do some colour transformations.
- We get the output of our detect function.
- We display the outputs.

If we type on the keyboard:

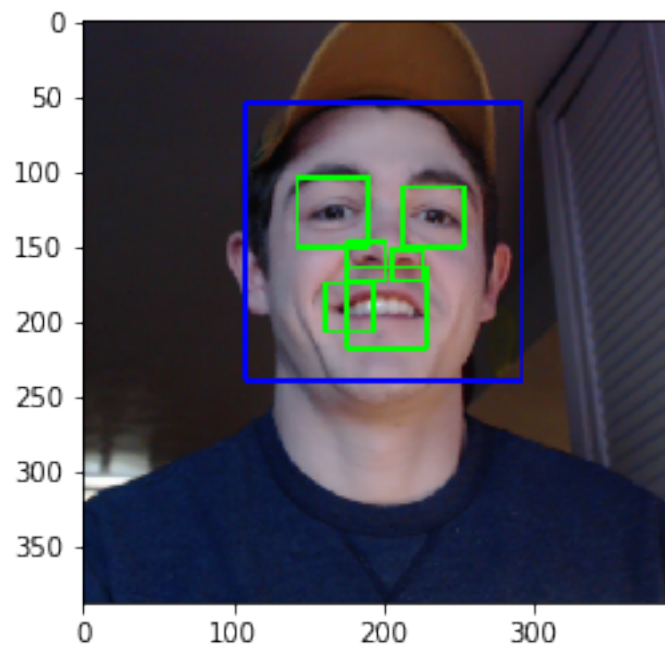
- We stop the loop.

```
In [46]: while True:
        _, frame = video_capture.read()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        canvas = detect(gray, frame)
        cv2.imshow('Video', canvas)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
```

We turn the webcam off and destroy all the windows inside which the images were displayed.

```
In [47]: video_capture.release()
        cv2.destroyAllWindows()

In [48]: import matplotlib.pyplot as plt
        import matplotlib.image as mpimg
        plt.imshow(mpimg.imread('output.png'))
        plt.show()
```



2 Homework Challenge

Many companies today use Computer Vision in their core business to detect emotions. For example, Apple recently bought Emotient, a startup that builds Computer Vision tools to recognize people's feelings (more info here).

Building an AI that sees emotions can be highly valuable in some markets. Imagine if you could detect people's emotions when they are watching movies. That would have a powerful impact on the cinema industry and bring significant added value to companies like Netflix or Time Warner for which you could build powerful recommender systems based on Computer Vision applications able to understand emotions.

The Homework of this first module is to make a simple Computer Vision application that can detect one emotion: happiness. Your mission is therefore to build a Smile Detector.

- Importing the libraries

```
In [2]: import cv2
```

- Loading the cascades

```
In [3]: face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
        eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
        smile_cascade = cv2.CascadeClassifier('haarcascade_smile.xml')
```

- Defining a function that will do the detections

```
In [4]: def detect(gray, frame):
        faces = face_cascade.detectMultiScale(gray, 1.3, 5)
        for (x, y, w, h) in faces:
            cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
            roi_gray = gray[y:y+h, x:x+w]
            roi_color = frame[y:y+h, x:x+w]
            eyes = eye_cascade.detectMultiScale(roi_gray, 1.1, 22)
            for (ex, ey, ew, eh) in eyes:
                cv2.rectangle(roi_color, (ex, ey), (ex+ew, ey+eh), (0, 255, 0), 2)
            smiles = smile_cascade.detectMultiScale(roi_gray, 1.7, 22)
            for (sx, sy, sw, sh) in smiles:
                cv2.rectangle(roi_color, (sx, sy), (sx+sw, sy+sh), (0, 0, 255), 2)
        return frame
```

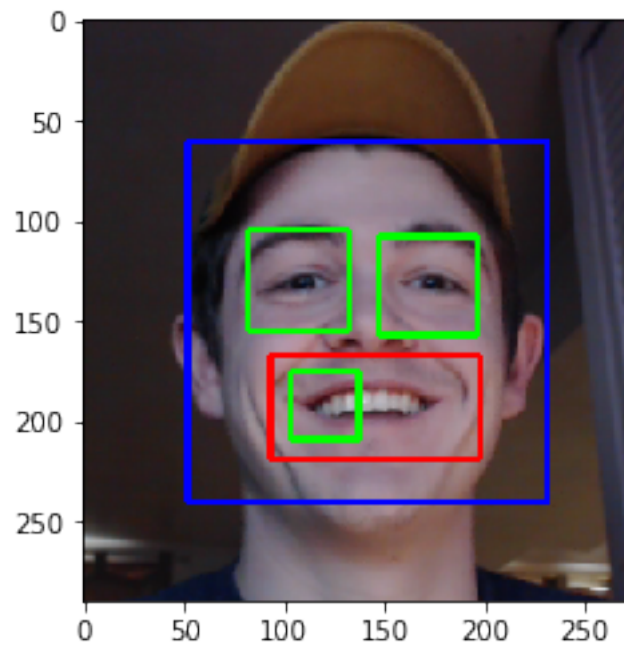
- Doing some Face Recognition with the webcam

```
In [5]: video_capture = cv2.VideoCapture(0)
```

```
In [6]: while True:
        _, frame = video_capture.read()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        canvas = detect(gray, frame)
        cv2.imshow('Video', canvas)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
```

```
In [7]: video_capture.release()
        cv2.destroyAllWindows()

In [8]: import matplotlib.pyplot as plt
        import matplotlib.image as mpimg
        plt.imshow(mpimg.imread('smile.png'))
        plt.show()
```



```
In [9]: plt.imshow(mpimg.imread('noSmile.png'))
        plt.show()
```

