Face Detection Project

By Brian Mallari

Project Description

The aim of this project is to apply a computer vision model on a set of pictures downloaded from the internet into a folder using a Python script in order to detect faces in those pictures. The model as well as the associated .prototxt file were originally from work done by Dr. Adrian Rosebrock, PhD, an expert in computer vision and deep learning. After face detection has been attempted by the Python script, the resulting pictures are then saved into a separate folder. Faces in these resulting pictures are tagged with a box along with a percentage indicating how confident the model believes the tagged item is an actual face.

Technologies Used

- Microsoft Windows 10 Home
- Python 3.8.6, along with the following modules
  - os
  - json
  - requests
  - shutil
  - cv2
  - numpy
- Jupyter Notebook (for initial code development and reference tracking)
- Visual Studio Code (for the remainder of code development)
- Microsoft Word 2016 (for the remainder of reference tracking)
- Command Prompt (for code testing)
- Comcast Xfinity (for internet access)

How to Run the Python Script

Make sure the following files are in the same folder/directory:

1. FaceDetection.py
2. deploy.prototxt
3. res10_300x300_ssd_iter_140000.caffemodel

Open up Command Prompt in Windows, and change the directory to the directory holding the above three files. Once in this directory, type the following command:

```
python FaceDetection.py
```

The Python script should create a folder/directory named folderForPictures, and provided that the computer running this Python script has an internet connection, an assortment of pictures will be downloaded from the internet and then moved into this folder. The pictures in this folder will then be analyzed for the presences of faces.

The Python script will create another folder/directory named folderForPictures_postDetection, save the results of the face detection, and then move the saved results into this new folder.

Over the course of the Python script, the following warning may arise:

```
libpng warning: iCCP: known incorrect sRGB profile.
```

This warning can be ignored. It comes up because the PNG reference library on the computer running the Python script is particular about the information in the ICC profile, which describes the color attributes of the displaying device or viewing requirement.

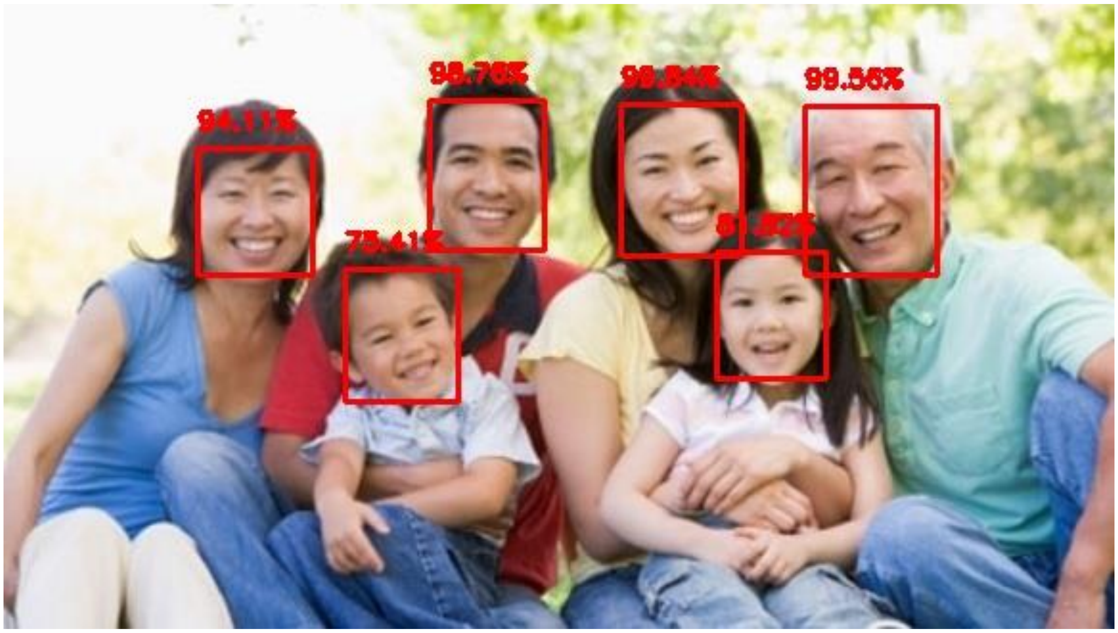
Once the Python script is done running, there should be two folders/directories created:

1. folderForPictures, with pictures before face detection
2. folderForPictures_postDetection, with the saved results of face detection

The results of the face detection process can then be viewed at leisure.

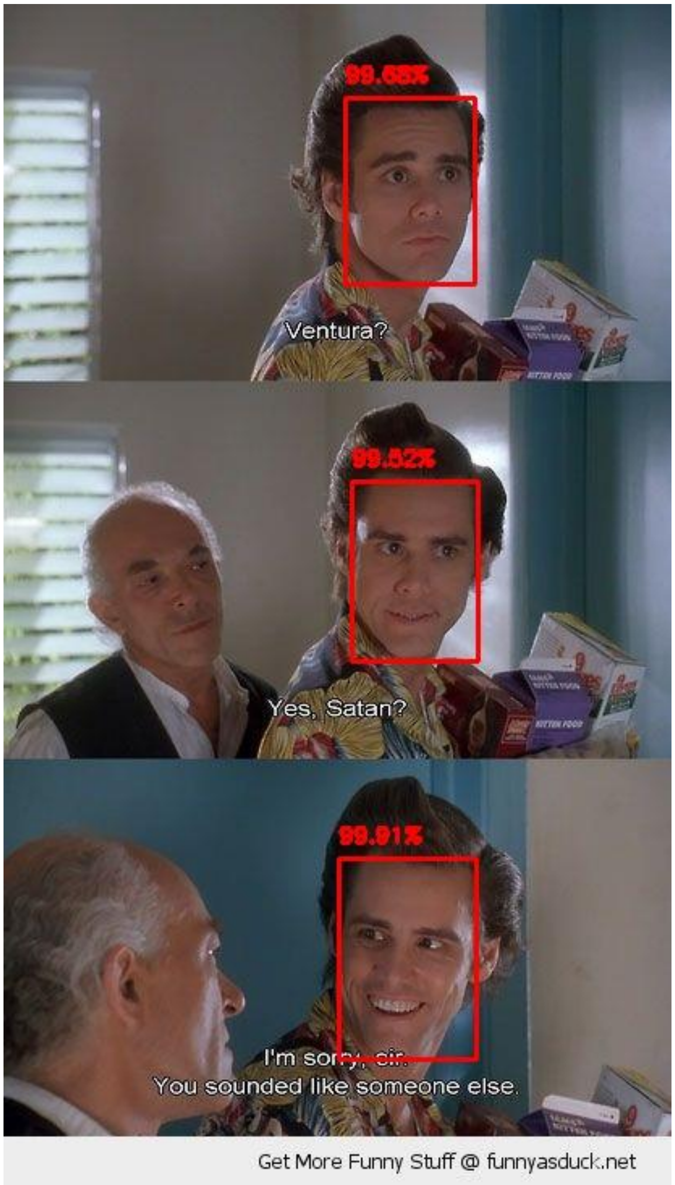
Additional Remarks

Many of the pictures after face detection will have all of the faces detected.



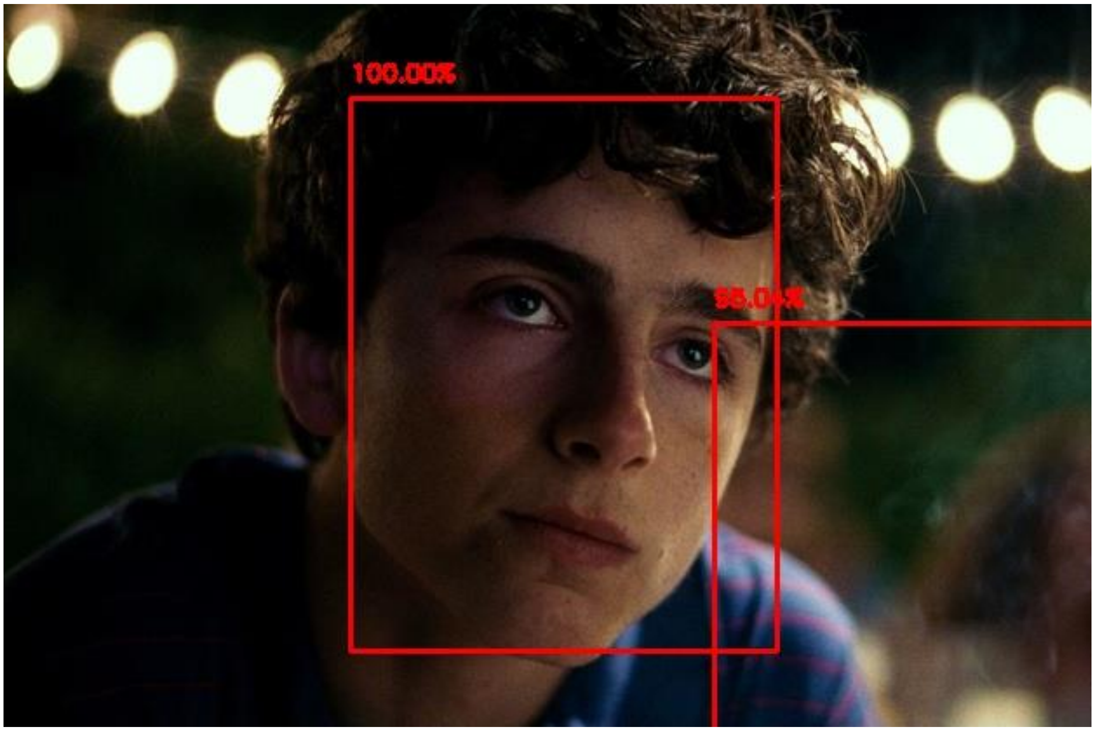
faceDetectionPicture_5_post_detection.jpeg

However, some of the pictures will have a face that was not detected, thus indicating a false negative.


99.65%
Ventura?
99.52%
Yes, Satan?
99.91%
I'm sorry, sir.
You sounded like someone else.
Get More Funny Stuff @ funnyasduck.net

faceDetectionPicture_58_post_detection.jpeg

Moreover, some pictures will have a “non-face” identified as a face, thus indicating a false positive.



faceDetectionPicture_29_post_detection.jpeg

Nevertheless, face detection can still be a useful technology in certain applications, such as counting the number of people entering a retail space, or identifying faces to blur in a picture for the sake of privacy.

References

See the document Face Detection References.pdf for references used for this project.