Final Project
Face Detection and Blur

04/16/2021

Jacob Zachariah
101010004
jacobzachariah@cmail.carleton.ca

Abstract:

My final project is a code that can detect a face in an image using Haar cascade and then apply a Gaussian filter to blur the x,y,h,w coordinates to blur out the faces.
I would not say my problem is challenging in terms of computer vision, but as a beginner to OpenCV and learning Gaussian filters, I think this level of project was perfect for me to learn and implement course materials.

Introduction:

My code uses the Haar cascade to identify a face in the image given, and then uses a Gaussian filter to blur the face. This would be a good candidate to be placed into a vision application in terms of privacy. This method can be used on videos or photos where the user wants to blur out their face.
It was challenging to me because I've never experienced working in OpenCV and this is the first Vision related course I've taken in Carleton.

Background:

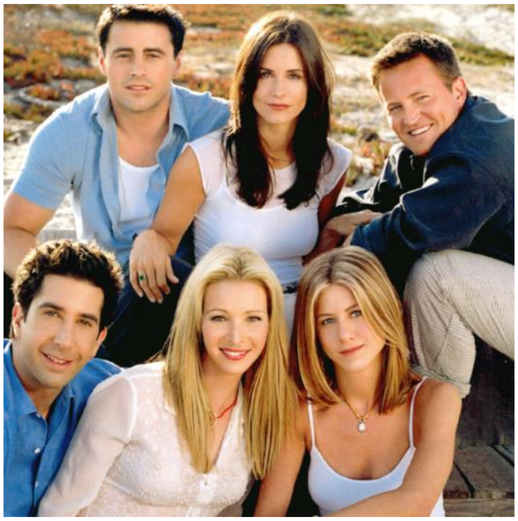
https://github.com/opencv/opencv/tree/master/data/haarcascades
Most of the project is based on the Haar cascade Frontal Face xml file that I got from the GitHub.
I learned that Haar cascade is a machine learning-based approach that is used to train certain classifiers. They use Positive and Negative images, positive are images that we want the classifier to identify, and negative are images of everything else that we do not want identified.
Then the gaussian filter was used from the lectures in class.

Approach:

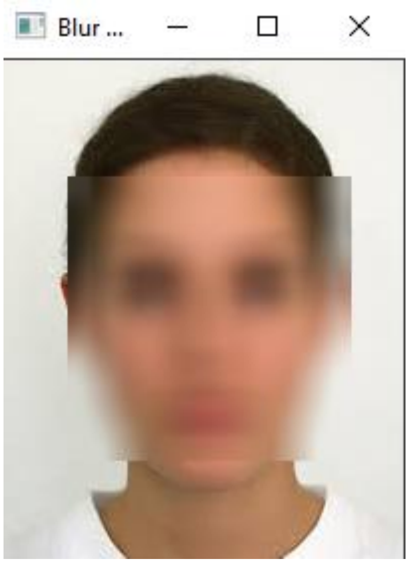
My code first loads in the cascade file which uses Haar cascade. Then it reads in an input image (I have included 2 images, please change the line7 from test1.jpg to test2.jpg for testing). Since the Haar cascade cannot detect faces that are not in grayscale, my code then converts the image to grayscale. Then it urns the cascade to detect faces. I had to set the minNeighbours value manually as the default value was too high and would either detect empty spaces as faces or miss faces entirely. Then, once we have our x, y, w, h coordinates of the faces, we can use them to form a Gaussian filter in them, in my case, I used a blur filter.

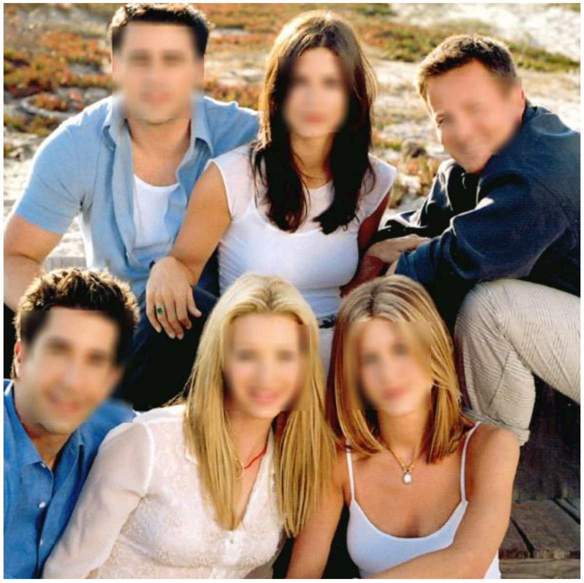
Results:

My final output is of an image with faces that has been blurred. Here are the two input images:



And here are the two output images:





List of Work:

This was a solo project.

GitHub Page:

https://github.com/JacobZ97/COMP4102_FinalProject