JANUARY 2023

FACE SEARCH

KNOW WHO IS THAT UNKNOWN PERSON IN IMAGE

Project Report
Prepared by: Aman Bharti
B.Tech. Electronics And Communication Engineering
IIT (ISM) Dhanbad

Motivation

This occurred to me while scrolling through my news feed that there were a lot of unknown people in many of the photos. Some of the photos also included a few people I already knew and followed posing with those 'strangers'. Happens quite often with most of us. Most of the times, we simply ignore or try to google (and retreat). We don't really have any idea what to do in such a scenario! After 10-15 minutes of Google searching and/or Google dorking[1], we sometimes even achieve our goal; but nonetheless, not in a straightforward way. I had learned OpenCV recently and having mostly gone through a lot of MOOCs in the summer (doing nothing but learning stuff), I thought why not give it a try!

The way to go

The Google reverse image search comes in handy when you can't tell Google what you want, in words. The case is similar here. We know the face, not the name. We can use it. However, just running an image search for the entire image wouldn't be much fruitful, as it would most likely return you the same source from where you got the image!
We can, however fine tune the task and force it to search for a particular face in the image by using OpenCV and its Cascade Classifier. Searching for only the face turned out to be giving better results. We also need to upload the cropped image automatically to Google and open a new browser window with the results. A little bit of research showed it to be possible without any API, since Google has a static link to which one can simply send a POST request (using requests in python) to upload an image. The returned response contains a header named 'Location' which is the URL of search results found by Google. We can then use python's builtin webbrowser module to open the URL in the user's browser.
That's all we'll need.

Implementation

The script is a command line application. Once installed, you just have to do 'facesearch path/to/image' to run the script on the image. OpenCV is an open-source computer vision library originally written in C++. It is widely used for computer vision related tasks. For the curious, computer vision is the field of AI that deals with making the computers 'see', i.e., work with images and derive useful data from it. OpenCV was used to detect the faces in the given image, which were then cropped out using Numpy, a python library for performing advanced mathematical stuff in python. Mostly used for working with matrices. The cropped faces are then padded appropriately and shown to user in a window, using OpenCV. OpenCV provides a dedicated function (cv2.setMouseCallback) to track the Mouse events on a window. It returns us the event name and the (x, y) coordinates of the event. Using the x, y coordinates of the click event, we can figure out which face user has clicked on. This allows for a convenient point-and-click operation.

For the convenience of dragging any image right off the internet and onto the terminal to get it searched, the script checks if the path provided to it is an internet URL and if it is, uses urllib to GET the image for further operations.

At last, to search for the selected face, the cropped out face is saved temporarily in the current directory and then uploaded to the Google's server using the static link. Once the image is uploaded and the response is received back, the location header is read and passed to webbrowser module, which then opens a new browser window or a new tab in an existing session with the search results. The script then cleans up the generated image and exits.