Facial Recognition and Web Scraping

Cristina Giraldo

Gregg Legarda

George Washington University

**Table of Contents**

[Chapter 1: Introduction 3](#_Toc17115370)

[Chapter 2: Data 4](#_Toc17115371)

[Summary of dataset 4](#_Toc17115372)

[Smart Question 4](#_Toc17115373)

[Chapter 3: Methods 5](#_Toc17115374)

[Face detection using haar cascade 5](#_Toc17115375)

[Face detection using facial recognition library 6](#_Toc17115376)

[Web Scraping 7](#_Toc17115377)

[Chapter 4: Results 7](#_Toc17115378)

[Chapter 5: Discussion 9](#_Toc17115379)

[Chapter 6: Conclusions and Recommendations 9](#_Toc17115380)

[Reference 10](#_Toc17115381)

# **Chapter 1: Introduction**

At a basic level, facial recognition works by obtaining geometry by scanning the face and recognizing patterns such as the distance between eyes, the size of the nose and mouth, and so on. With that information, the computer can create a virtual map of the face and is then able to perform a match against other faces to identify the appearance of the person who is being captured through a camera or a digital image (Bala & Watney, 2019). Nowadays, facial recognition has become a subject of great importance in various fields. For example, it is used in law enforcement to identify criminals, in social media to recognize friends, in technology to drive cars without a driver, in smart devices to unlock the device with just the look of the face and in surveillance to control communities. Given all these fascinating applications, we are interested to understand how this technology works by using facial recognition in real time and implementing web scraping to obtain basic information about Instagram users' accounts and to present such information on the facial recognition.

# **Chapter 2: Data**

## **Summary of dataset**

The dataset used for this project is a collection of personal pictures of the team members. The dataset lists about 1 to 2 pictures of each person mentioned above along their Instagram user account when face recognition library is used. To clarify, when the Haar-like feature algorithm is used more pictures were necessary. The dataset is divided in subfolders, which each is named with Instagram’s user of each person.

## **Smart Question**

Is it possible to identify and obtain basic information about one or more people on the video in real-time according to digital images and Instagram’s users stored in the dataset? The purpose of this analysis is to determine whether web scraping and facial recognition can be used to obtain information about users in real-time and how this can be impactful.

# **Chapter 3: Methods**

The analysis conducted in this report is based on the video and images captured by a webcam in real time and the users' account on Instagram. It is worth remembering that the dataset has a sample of two to three images per each team member.

To respond to the SMART question, some analysis was carried out on the dataset, which included the application of two algorithms for facial recognition (Haar-like feature based and HOG detection method), and web scraping. For the analysis, encoding and detection, we made use of PyCharm software using dlib and face\_recognition libraries. For reading the web cam, OpenCV and imutils libraries were used. For web scraping, Instagram’s API was used. For the Haar algorithm a .yml and .xml file were need it. Finally, for visualization Plotly Dash was used.

## **Face detection using haar cascade**

This algorithm proposes the use of a cascade classifiers. It uses many positive (images with faces) and negative pictures (images without faces) and addition this algorithm combines many weak classifiers with a threshold and define many features to do the recognition (Viola & Jones, 2001).

## **Face detection using facial recognition library**

**Encoding.** To encode images, the face encoding method from the face recognition library was used. This library uses a deep neural network to recognize a face. In this process the image undergoes a transformation where the picture is converted into numbers to be read by a computer. It is worth clarifying that each image will produce different information unless the face is the same.

Since in this project dlib library was used, it was not necessary either to create our own neural network or to train the algorithm. Instead, we used the pre-train model contained in the dlib library, which helped to classify our own images correctly.

To convert our images to numbers, we used the face\_encoding method. This method allowed us to get the representation of the images in different arrays of 128 elements, and finally the information is stored in the encodings.pickle file to be used in the recognition afterwards.

**Histogram of Oriented Gradients (HOG).** The purpose of a HOG, which is a type of “feature descriptor,” is to generalize an object which will produce a similar outcome to the same feature descriptor when the object or the face is viewed under different conditions. A HOG thus makes classification simpler (McCormick, 2013).

HOG uses a single vector to represent a complete image. It is normally used with SVM classifiers. Each HOG descriptor that is computed is fed to a SVM classifier to determine if the object was found or not (Dalal & Triggs, 2005).



*Figure 1.* Gradient

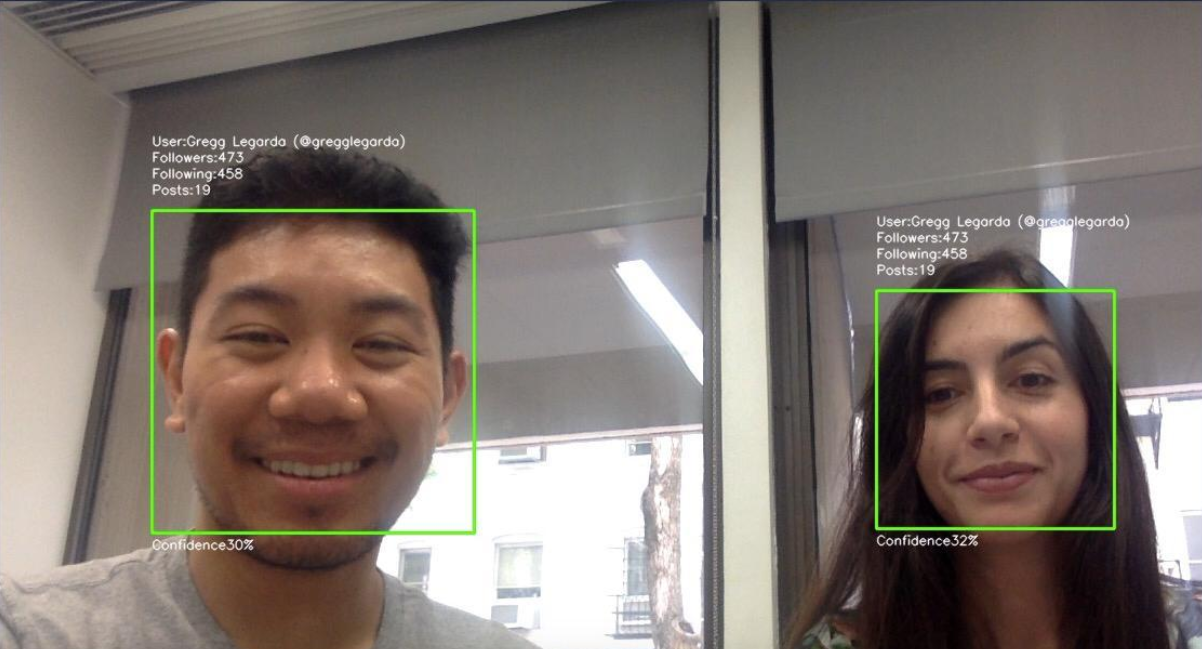
## **Web Scraping**

This technique is commonly used to extract data from different web sites by using an API to facilitate the extraction of information. For this project we made used of this technique to get information of an user, then clean the data and finally, we present the user’s data on the screen as part of the facial recognition and give an answer to our smart question.

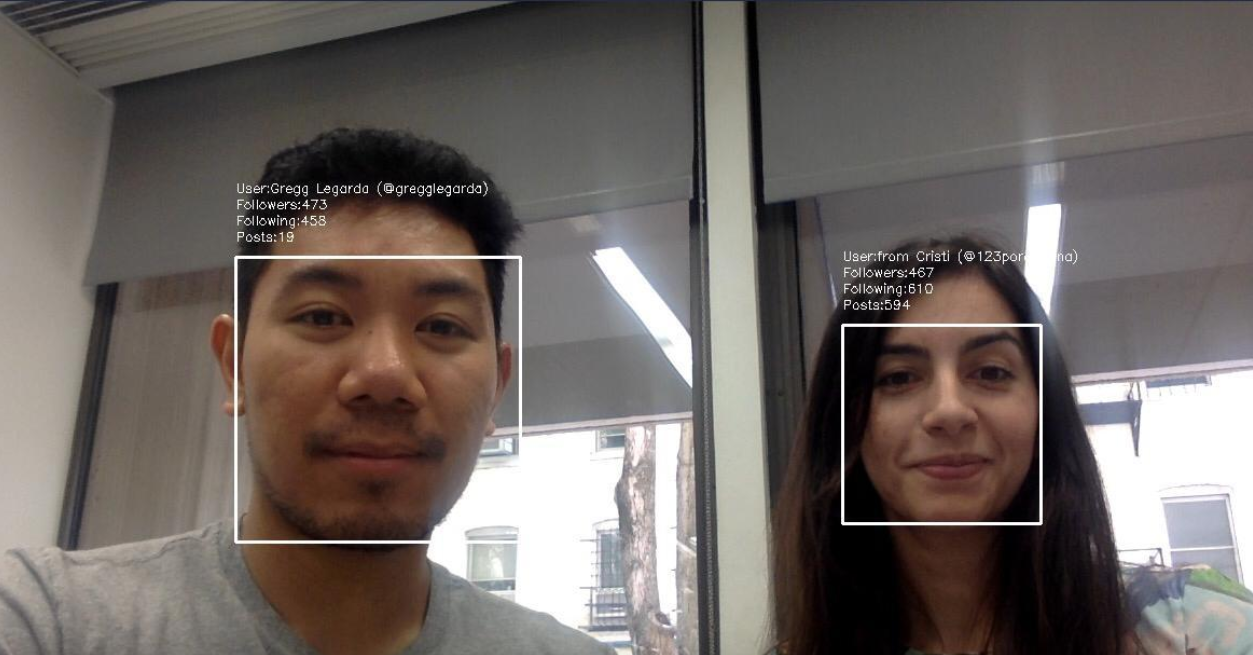
# **Chapter 4: Results**

For the haar algorithm about 23 pictures were trained. Most of the images belonged to Gregg. After the algorithm was executed, we could notice that the accuracy was not precise (the result can be seen on figure 2) and the algorithm works under certain conditions, for example just for frontal face. However, Haar provided a good speed on the video and the recognition.

For the HOG algorithm fewer digital images were necessary to train, and the facial recognition work well under different conditions and different positions (see figure 3). Thus, the HOG method could predict an individual correctly in about 80% of the time while the Haar algorithm had gave us a confidence of about 35% due to very few images were used.



*Figure 2.* Facial recognition using Haar-like feature based



*Figure 3.* Facial recognition using Histogram of Gradients (HOG)

# **Chapter 5: Discussion**

To recap, the objective of this paper is to determine if it is possible to identify and obtain basic information about one or more people on the video in real-time according to digital images and Instagram’s users stored in the dataset and how this can be impactful.

From the results obtained in each algorithm and which the results can be observed in Figure 2 and 3 we were able to identify the users and obtain basic information from Instagram such as followers, following and amount of posts.

In summary, the answer to the S.M.A.R.T. question appears to be that it is possible to get the information of a user based on their Instagram’s user and with that information draw conclusions about someone. By using other API to do web scrapping would be possible to get even more information and do an analysis of someone. However, it is worth to mention that we do not do a matching face due to the complexity and the time that would take to build this algorithm.

# **Chapter 6: Conclusions and Recommendations**

Our project used different techniques to do facial recognition such as Haar and face\_recognition and for the dataset we used our own pictures to train the different algorithms .

According to our result from comparing the accuracy rate among the 2 methods used for this project, we can tell that both codes have advantages and disadvantages. First, Haar provides speed while face\_recognition library provides accuracy.

Second, analyzing the Haar recognition we realize that it is necessary to train a big number of digital images to get a precise identification of the individual on the camera. Third, HOG code does not need to train a lot of image due the pretrain algorithm that has built in, however in real time facial recognition using hog is significantly slow.

We also test the algorithms with Asian faces and few times we were successful. After doing some research we found out the facial recognition library has a limitation regard to Asian faces and the limitation is due to the fact the pre trained model was done over occidental faces.

Another limitation presented is that sometimes the algorithms can get confused between people with similar features. Humans are very good at recognizing faces but these algorithms that we used has a long way to surpass this human ability. Sometimes the machine will make mistakes and identify a face with a different name. One solution to improve the recognition is by decreasing the threshold. In our trials this change did not improve the accuracy, but probably it was due the lack of information to train the algorithm.

# **Reference**

Bala, N., & Watney, C. (2019, June 19). What are the proper limits on police use of facial recognition? Retrieved from <https://www.brookings.edu/blog/techtank/2019/06/20/what-are-the-proper-limits-on-police-use-of-facial-recognition/>

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05), 886-893. doi:10.1109/cvpr.2005.177

McCormick, C. (2013, May 09). HOG Person Detector Tutorial. Retrieved from <https://mccormickml.com/2013/05/09/hog-person-detector-tutorial/>

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. doi:10.1109/cvpr.2001.990517

Sources:

https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/

https://towardsdatascience.com/real-time-face-recognition-an-end-to-end-project-b738bb0f7348

https://www.youtube.com/watch?v=jG3bu0tjFbk

http://scipy-lectures.org/advanced/image\_processing/

https://www.youtube.com/watch?v=PmZ29Vta7Vc

https://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html

<https://www.promptcloud.com/blog/how-to-scrape-instagram-data-using-python/>

There are two pre trained algorithms used in the project. HAAR cascades and HOG detection.

HAAR uses 1million image???

While HOG uses a pretrained data set of ---

We collected our own data by taking frontal facial images of ourselves

For HAAR, we used a .yml file to store our trained faces so we can run it in the main

There are limitations of the data set since sometimes it can get confused between people with similar features. Humans are very good at recognizing faces but these algorithms that we used has a long way to surpass this human ability. Sometimes the machine will make mistakes and identify a face with a different name