Facial Recognition and Web Scraping

Cristina Giraldo

Gregg Legarda

The George Washington University

Table of Contents

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 the 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 transportation 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

Description 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 team member along with their Instagram user account when the facial recognition library is used. To clarify, when the Haar-like feature algorithm is used, more pictures were necessary. The dataset is divided into subfolders, each of which 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' accounts 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 needed. Finally, for visualization Plotly Dash was used.

Face Detection Using Haar Cascade

This algorithm proposes the use of cascade classifiers. It uses many positive pictures (images *with* faces) and negative pictures (images *without* faces). In addition, this algorithm combines many weak classifiers with a threshold and define many features to perform the recognition (Viola & Jones, 2001). Figure 1 represents how the algorithm proposed by Viola and Jones works.

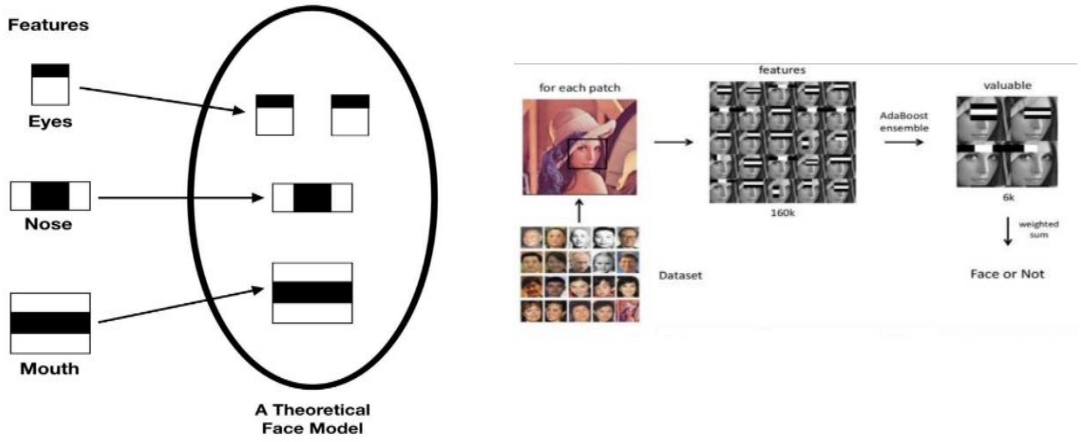

Figure 1. Viola-Jones Algorithm

Face Detection Using Facial Recognition Library

This library provides an easy way to implement facial recognition in real time with accuracy. But what it makes this library powerful is the fact that it is implemented along with dlib library. Dlib uses a deep neural network called deep metric learning which is popular for the use in face recognition. Thus, dlib helps to perform the facial recognition by implementing a ResNet network with 29 convolutional layers and training three million of images (King, 2017). The trained model become in the pre-trained model for facial recognition library that is why use of this library allowed as to perform facial recognition accurately. Although, facial recognition library provides the possibility to perform a CNN to perform the facial recognition, in this project we decided to use HOG given that our computers did not have the power to process a convolutional network. Even tough we got good result at the process of recognizing the person on the screen.

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. Each image will produce different information unless the face is the same.

Since in this project a 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-trained 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 an SVM classifier to determine whether the object was found (Dalal & Triggs, 2005). In the figure 2 can, be observed how HOG calculates the distribution according to small cells (Greche & Es-Sbai, 2016). In addition, figure 3 shows how the gradient is presented on the computer.

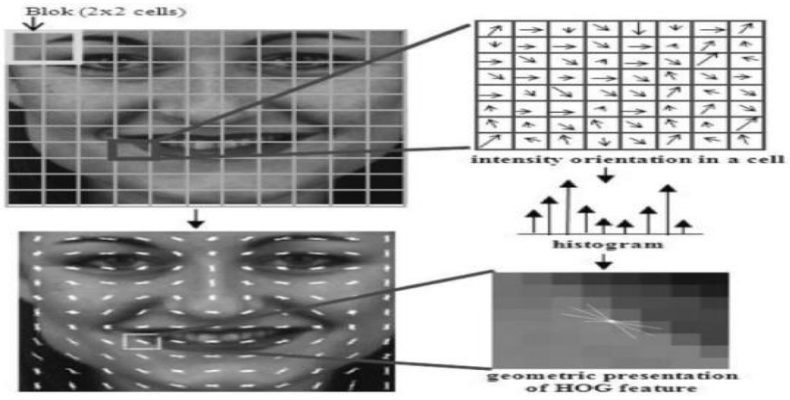

Figure 2. Histogram of oriented gradient extraction from face

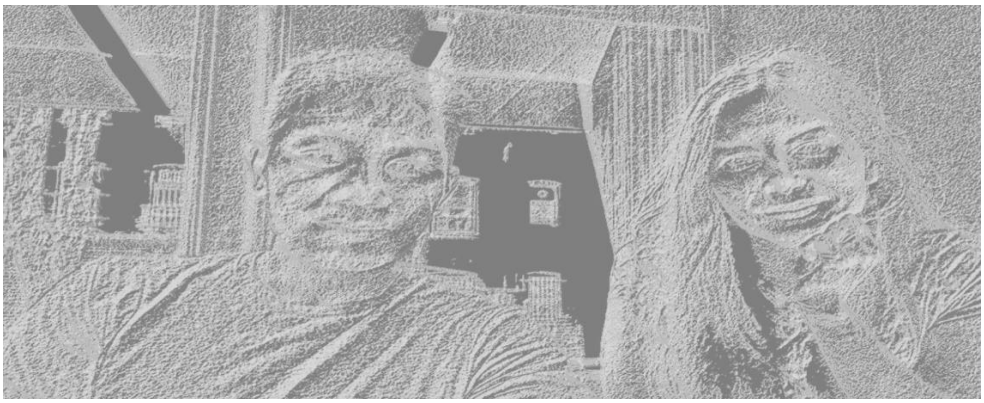

Figure 3. Gradient visualization from the computer

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 use of this technique to get information of a 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 noticed that the accuracy was not precise (the result can be seen on figure 4) and the algorithm worked only 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 worked well under different conditions and different positions (see figure 5). 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 limited number of images used.

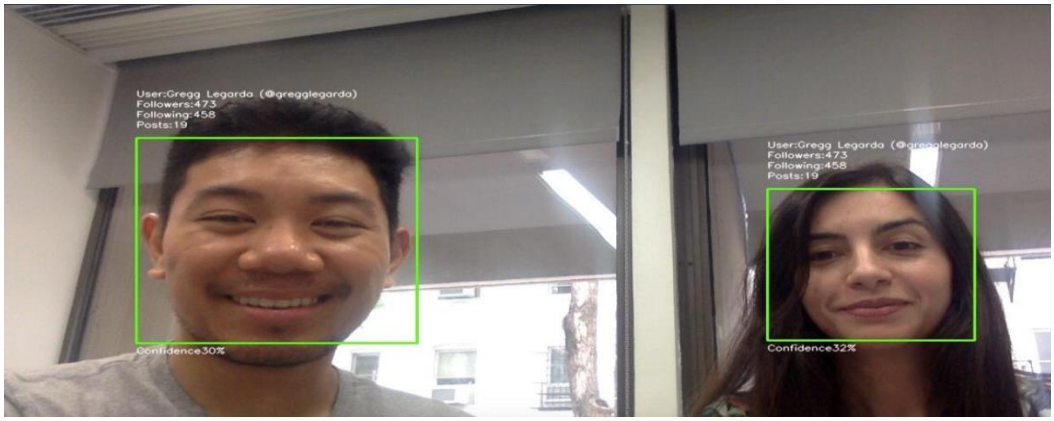

Figure 4. Facial recognition using Haar-like feature based

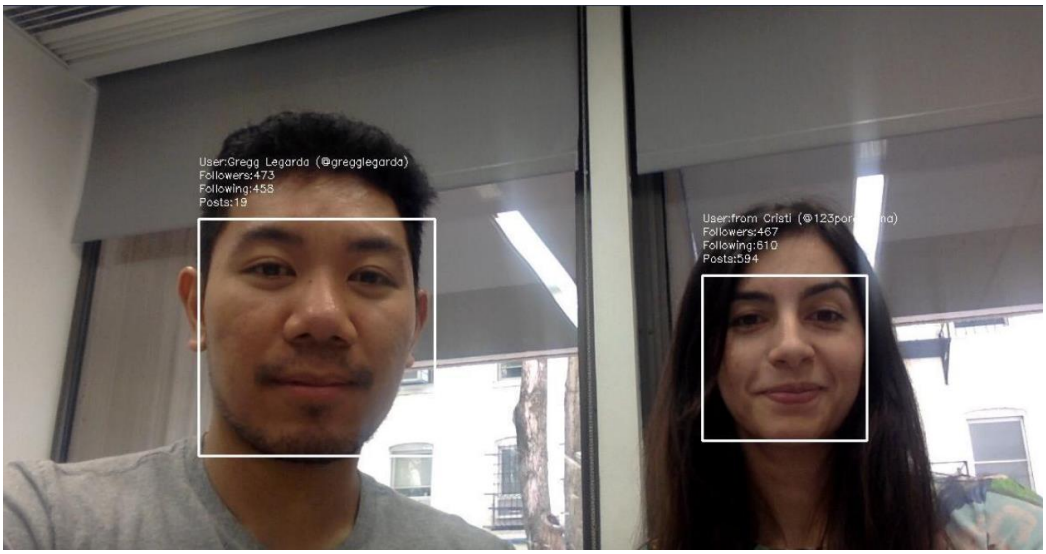

Figure 5. Facial recognition using Histogram of Gradients (HOG)

Chapter 5: Discussion

To recap, the objective of this report 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 results can be observed in Figure 4 and 5, we were able to identify the users and obtain basic information from Instagram such as followers, following users and number 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' user profile and, with that information, draw conclusions about a person. By using other API to do web scraping, it would be possible to obtain even more information and thus do a more thorough analysis. However, it is worth

mentioning that we did not do a matching face online 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 perform 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 results from comparing the accuracy rate among the 2 methods used for this project, we determined that both codes have advantages and disadvantages. First, Haar provides greater speed while, face_recognition library provides greater accuracy.

Second, analyzing the Haar recognition, we realize that it is necessary to train Haar algorithm with a large number of digital images to get a precise identification of the individual on the camera. Third, HOG code does not need to be trained on many images due to its pretrained, built-in algorithm. However, in real time, facial recognition using HOG is significantly slower.

We also tested the algorithms with Asian faces, and we were successful only sporadically. After doing some research, we discovered that the facial recognition library has a limitation regarding to Asian faces and the limitation is due to the fact that the pre-trained model was performed using largely occidental faces.

Another limitation presented is that sometimes the algorithms can get confused between people with similar features. Humans are very good at distinguish between similar faces but the 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 on the HOG algorithm to make the algorithm stricter. In our trials, this change did not improve the accuracy, but probably it was due the lack of information to train the algorithm.

References

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

Greche, L., & Es-Sbai, N. (2016). Retrieved from https://www.semanticscholar.org/paper/Automatic-system-for-facial-expression-recognition-Greche-Es-Sbai/6dc82921719319deb1a193febb0388b2f32cecb0

King, D. (2017, February 12). High Quality Face Recognition with Deep Metric Learning. Retrieved from http://blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html

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