Facial Recognition

Cristina Giraldo

Gregg Legarda

George Washington University

Table of Contents

[Chapter 1: Introduction 3](#_Toc16712860)

[Chapter 2: Data 4](#_Toc16712861)

[Summary of dataset 4](#_Toc16712862)

[Smart Question 4](#_Toc16712863)

[Chapter 3: Methods 5](#_Toc16712864)

[Encoding 5](#_Toc16712865)

[Histogram of Oriented Gradients (HOG) 6](#_Toc16712866)

[Web Scraping 6](#_Toc16712867)

[Chapter 4: Results 11](#_Toc16712868)

[Chapter 5: Summary and Conclusions 13](#_Toc16712869)

[Reference 15](#_Toc16712870)

# **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**

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

The data used for this project is a collection of personal pictures of the team members and some extra photos that can be taken during the presentation if the individuals accept to participate in it. The dataset lists about 2 to 3 pictures of each person mentioned above along their Instagram user account. The dataset is divided in subfolders named with Instagram’s user of each person.

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

## **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**

We first tried to detect faces using pictures before moving on to the live webcam feed

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 encoding, 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. Finally, for visualization Plotly Dash was used.

## 

## **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 images. 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).

## **Web Scraping**

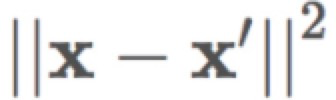
## **Web Scraping**

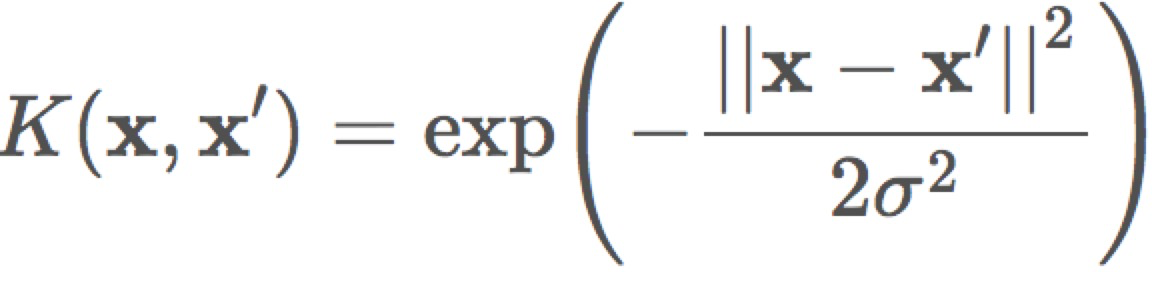
**Experimental Setup**

We implemented two methods to do the facial recognition. The methods that we implemented were Haar and HOG.

**Support Vector Machine with RBF kernel**

support vector machines is an algorithm that determines the best decision boundary

between vectors that belong to the given group and vectors that do not belong to it. Here, we choose to use the SVM RBF Kernel for the purpose of comparing the result with the other two methods. The RBF kernel is defined as below. The recognized as the [squared Euclidean distance](https://en.wikipedia.org/wiki/Euclidean_distance#Squared_Euclidean_distance) between the two feature vectors.

 *Figure 2.* RBF kernel

# **Chapter 4: Results**

By using first method Haar we wanted to find out how accurate was the algorithm and we found out that the algorithm was not very accurate. However, Haar provided better speed on the video and the recognition. On the other hand, HOG needed fewer digital images to train. Also, HOG under different conditions and different positions presented a better accuracy and almost 99% of the time the algorithm could predict an individual correctly.

# **Chapter 5: Summary and Conclusions**

Our project used text mining techniques to draw meaning out of the written online reviews. Unlike normal data mining, most of the text mining data is unstructured with a content that can be valuable. However, it requires to implement several steps of preprocessing to extract the meaningful information.

In our project, we use different preprocessing methods, such as removing common words, stop words, rare words, lemmatization, stemming, and spelling correction. At the end, we reduced our vocabulary cluster so that features produced for the classification model would be more accurate.

According to our result from comparing the accuracy rate among the three models, we can tell that the linear vector machine had a better performance. Following are some explanations about the reasons to lead this outcome. First, the text has a lot of features and the linear kernel is good when lots of features are used. It is doesn’t really help to increase the dimensional space like using the RBF Kernel. Second, analyzing the text related information means that it take more time. Training with linear kernel is faster. Third, comparing with the other kernels, the linear need less parameters to optimize when train the data.

We want also to discuss the improvements that have been made for the better outcome. In case of the linear vector machine with lasso penalty, the model could predict a 50%. If we consider that the data it is only 10,000 instances, we would expect that the model increase the percentage of prediction with more data. We can also conclude that the model is better with 1-3 N-grams, compared to 1-1 or 1-2. We have also improved this model by using chi-squared to improve the feature selection.

Last but not least, by re-evaluating the observations from the sample dataset, we found out most of the comments are subjective that people may use their own “rating system” to give the their own hotel rating. Also, people are more likely to give the 4+ or 5 score even they have slightly positive hotel experience. As result, we can see some limitation by nature on predicting using the “1 to 5” rating system. To get a better outcome, we went back to the preprocessing stage and did the test to use different scales on our hotel rating column. As we can see in our comparison by taking SVM Kernel as an example, if we rescale to the negative (for ratings from 0 to 3), Moderate (for ratings from 3 to 4) and positive (for ratings from 4 to 5 ), our accuracy rate will significantly increase from 46.20% to 74.36%.

# **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/>

García, S., Luengo, J., & Herrera, F. (2015). Data preprocessing in data mining. Cham: Springer.

<https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html>

A. N. (n.d.). *Support Vector Machines*. Lecture

<http://cs229.stanford.edu/notes/cs229-notes3.pdf>

Jain, S. (2019, March 11). Ultimate guide to deal with Text Data (using Python) - for Data Scientists and Engineers. Retrieved from <https://www.analyticsvidhya.com/blog/2018/02/the-different-methods-deal-text-data-predictive-python/>

R. (n.d.). *An Idiot’s guide to Support vector machines (SVMs)*. Lecture.

Text Analytics for Beginners using NLTK. (n.d.). Retrieved from <https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk>

Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. J.R.S.S.B. 58, 267-288

A. (2016, February 24). *Topics in sparse Support Vector Machines*. University of Arizona.

<https://www.math.arizona.edu/~wammonj/talks/topics-sparse-svms.pdf>

J., S., T., & R. (n.d.). *1-norm Support Vector Machines*. Stanford University

<https://papers.nips.cc/paper/2450-1-norm-support-vector-machines.pdf>

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/