Social Media Image Analysis with Google Cloud Vision

Paine Leffler, Jiaming Zhao, and Eleanor Leffler

***Abstract*— This is a final project proposal for Digital Archeology which will explain the motivation, data, milestones, and expected outcomes. For this project, we will be anaylzing images extracted from public social media accounts, such as Facebook, Twitter, Instagram, and LinkedIn. To do this, we will be using the Google Cloud Vision API, social media APIs, Node JS API, and data vizualization tools.**

I. INTRODUCTION

Social media has revolutionized the way information is shared on the Internet. Because sharing is so quick and easy, pictures and personal information have become abundantly available. Additionally, online presences have become much more relective in personality of the individual. However, judging an individual by his or her social media can be biased or incorrect. As much as individuals do not want to be judged by the pictures or data about them online, many people still do. Also, an image in memory can seem to be meaningless to humans, but with further analysis, there are hidden patterns that can provide clues to what the images contain. The main goal of this project will be to analyze that information and provide labels for a specific individual.

II. MOTIVATION

Individuals are publishing less textual information and more pictorial information by taking hundreds of pictures every week and posting them to some sort of social media. While humans can easily recognize attributes of pictures, analyzing hundreds or thousands of pictures becomes time consuming. Utilizing Google Cloud Vision, this project allows users to quickly understand how their social media might be perceived by outside observers. For example, employers often use social media to judge the behavior of a candidate. This project gives candidates a tool to determine how an employer might perceive him or her. This project will help people understand how their public images might affect their reputation.

III. DATA

The data will be derived from social media APIs. These social media APIs include Facebook, Twitter, Instagram, and LinkedIn. Images from these sources will be submitted to the Google Vision API for processing. We then receive the labels and scores for each image, classify, and categorize them. From this we can give breakdowns of an individual's interests and image content.

IV. MILESTONES

This is a breakdown of the milestones in the order to be completed for this project.

A. Google Cloud Vision API

The Google Cloud Vision API will be used to process a specific image at a given URL link. Using a link will eliminate the need for downloading and uploading images. This method will allow for quick data analysis and will speed the transfer of data from social media sources to the Google Vision API.

B. Social Media Image URL Extraction

Social media APIs will be used so that a user can retrieve a given quantity of the most recent image URLs posted on that social media. For example, we use Twitter-js-client npm package to request Tweets from public users. Then we parse the JSON result from the Twitter API, pull the media URLs, and use them for analysis. We will also analyze the profile picture and background picture of the user to give us more information.

C. Large Scale Vision API Processing

Many retrieved image URLs need to be efficiently processed by the Google Cloud Vision API. The Google Cloud Vison API is advantageous because it only requires the URL link to analyze the image. It will reduce time in downloading images from social media sites. We will send the URL links to the Google Cloud Vision API and the response will contain labels and other important image information. The result includes labels relative to the image and the confidence of each label.

D. Data Needs Classification and Grouping

The scores and labels given by the Google Cloud Vision API need to be summed and indexed to give weight to scores against the entire group of images classified. Also, specific labels need to be grouped and organized by larger encompassing categories. There is a case that an individual does not have a large number of images. If we just consider the sum of confidence, we will get relatively low score for that individual compared to others. To combat this issue, we will use the equation (sum confidence)/ (number of images) as the confidence score.

E. Result Caching

The scores and labels given by the Google Cloud Vision API need to be stored and keyed off of the URL link in order to save processing of repetitive images. When we retrieve the latest 1000 images from the same individual, there may be just 10 new images posted after the last retrieval. We do not want to waste time re-analyzing all 1000 images again. Instead, We just need to anlyze those new images. We would

store the URL link associations to labels and scores so that we only need to process an image once. When we get an URL link, we will check if it exists in our database. If it not, then we submit it to Google Cloud Vision API for analysis.

F. Node JS API

To optimize API calls, we will build an API wrapper that will accept usernames and return media URL links. It will also have an endpoint that will accept URL links and check our database to see if the image has been processed yet. Then it will submit the links to the Google Cloud Vision API if needed. With this method we have more real-time feedback on the runtime and progress of the image analysis. For example, if we submit 1000 pictures as one API request, we have one giant process without a progress indicator. But by submitting them in sections or individually, we have a frame of reference of runtime and can relay this back to the user with a loading bar or animation. The API will be written in Express JS and will manage all of the routes.

G. Data Visualization

The data will be put into Chart.js to graphically represent our user's data by categories and percentages. We will show labels as a pie chart to show the top percentages of all images available from the user. We can also show every specific score breakdown for each label given to the images of the user. Possible stretch goals would be to indicate negatively connotated labels or categories such as drugs or alcohol. This way the application can be proactive in helping users adjust their online profile.

H. User Interface

The tool will need to be designed so that a user can easily search and find their associated labels for their images on social media. The front end will be built on React JS and the backend will service API requests in Node JS. The User Interface will mainly feature a search bar that can allow specifications of usernames to each social media.

V. EXPECTED RESULTS

We want to get labels that exactly fit the interests of the individual. We can classify these people into groups based on similar interests. The outcome of this project will be a tool to assist people in classifying an individual's pictures and categorizing them into interests, locations, or content. From this, individuals can draw conclusions from a percentage of images with given labels. Individuals could analyze his or her public web presence.

REFERENCES

[1] https://cloud.google.com/vision/docs/apis
[2] https://developer.twitter.com/en/docs
[3] https://developers.facebook.com/docs/javascript