

Google Cloud Vision API Usage Report

Eloy Gil Guerrero
eloy.gil@est.fib.upc.edu

June 2017

1 Introduction

Google Cloud Vision API¹ enables developers to understand the content of an image by encapsulating powerful machine learning models in an easy to use REST API. It quickly classifies images into thousands of categories (e.g., "sailboat", "lion", "Eiffel Tower"), it makes possible to detect individual objects and faces within images, and finds and reads printed words contained within images. You can build metadata on your image catalog, moderate offensive content, or enable new marketing scenarios through image sentiment analysis. Cloud Vision API can analyze images uploaded in the request or integrate with your image storage on Google Cloud Storage.

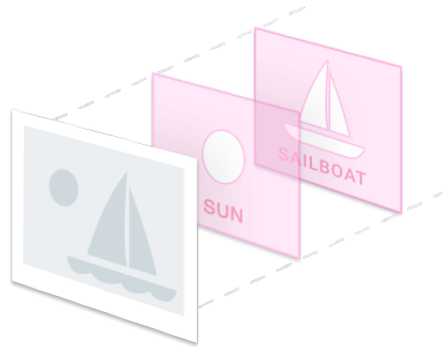


Figure 1: Powerful image analysis

¹Google Cloud Vision API homepage: <https://cloud.google.com/vision/>

2 Work done

During this introduction to Google Cloud Vision API usage there are several steps followed since the setup of the environment until the actual testing of the features tool.

2.1 Cloud Platform sign up

There is a free trial of this service, but to fully enjoy the benefits of Google's Cloud platform it looks like the best option is to get a business trial, with \$300 worth of free credit for this year. That amount is far more than enough for the expected needs on this subject's scope. Once inside the management console, it is possible to create a new project, and enable an API that is the main tool where the credit is going to be spent. Finally, it is needed to set up some credentials that will be the way we authenticate to enable the communication.

2.2 Environment setup

The base language is Python and those packages and almost every needed module was already installed. There is a GitHub repository with scripts and images that are ready to be used, there is only need to *clone* it. A new requirement was the setup of a *Redis Server*. As in this case it is used Ubuntu as operating system this can be done with a simple *apt-get install* command and get it installed and launched in seconds. An important detail is to set the path to the authentication key generated for our API as a specific environment variable (GOOGLE_APPLICATION_CREDENTIALS) that will be the contact point of the code with the API.

2.3 Running the example

The proposed example will be useful to check that everything is working fine and feel the power of this tool: It consists in executing a Python script that, given a test image data set folder analyzes each picture to extract text and it is indexed in the local Redis Server. After this, if everything went fine it should be possible to query using keywords we know that are present in the text of those images to see that they were detected and classified properly. It is important to remember that *stopwords* are filtered, so if we query one (i.e.

"*up*") the response will be "No hits" and could be kind of misleading. At this point there were suggested some changes to the official repository because the example results were not matching with what it shows, after opening an issue and investigating what was wrong. This were just some minor issues that could make this hands on harder to inexperienced users.

2.4 Code explanation

The example runs a script that is explained in detail: From the beginning, where it takes place the building of the Vision API Client, the processing of a request, until an inverted index is built using the image OCR results. This part is easier to understand if you are familiar with Python programming.

3 Conclusion

During this first contact with the Cloud Vision API it is demonstrated that it is possible to configure and launch in minutes a working application that analyzes images in the cloud. The examples are run in a local machine, but moving this to a server and create a business around it is the way to go during this era. Depending of the interest it is possible to ask for different answers and the way the data obtained is processed and used enables a new world of new opportunities, with uses like detecting inappropriate content or easily detect broad sets of objects, among others, that are used in a transparent way everyday by everyone.