

Golf_SwingML

By Luis R. Alvarez Jr



Tiger Left elbow angle: 187 deg



Luis Left elbow angle: 189 deg

Golf SwingML

Off the tee, in the year 1997 the average PGA tour driving distance was 267 yards. At the same time Tiger Woods is outperforming his field by averaging 294 yards ([Woods](#)). As of August 2021, the tour average driving distance has caught up to Mr. Woods. Today a normal drive on the men's tour is a whopping 295 yds. With the number one player hitting an average distance of 320 yds ([DeChambeau](#)). If the pros can increase their driving distance, why can't you?

This project uses Python and Google's Video Intelligence Application Interface (API) to analyze golf swing video footage to better understand how the best in the world hit the long ball. Specifically, Google Cloud (GC) offers services that allow programmers to leverage cloud storage and machine learning to annotate video with labels. The GC API produces a Java Script Object Notation (JSON) file that contains a dictionary object that labels coordinates on the video and estimates what the point is, such as eyes, nose, shoulder, elbow and wrist. These annotations are then processed using python, pandas, matplotlib, and math libraries to produce a Comma Separated Values (CSV) file that contains a time series of golf swing angles.

Google Cloud Bucket Setup

To setup a Google bucket sign into a google cloud account and browse to the CloudStorage menu. Click on Create Bucket, type a globally unique and permanent name, select region location, select standard storage class for data, and then select uniform access control. This last step allows for permissions to the resource via Identity Access Management (IAM). Next, browse to Create Project, click create; note the project name, organization, and location are setup here. Before the API can access the bucket and video mp4 file configure permissions by browsing to Service Accounts and selecting Create Service Account. Type a service account name, grant the role of owner and select done. Select the service account just created and click on Keys, Add Key, and Create. Check the downloads folder and there will be a unique alphanumeric JSON file that will allow remote access to the cloud resource. Store this key in the same directory used to start virtual environment.

Create [bucket](#), Configure [Service Account](#), and [Project](#)

The screenshot shows the Google Cloud Platform console interface. At the top, there's a navigation bar with the Google Cloud Platform logo, a dropdown menu for 'golfswingml', and a search bar. Below this, the 'Bucket details' page for 'alvarez-golfswingml-bucket' is displayed. The page has tabs for 'OBJECTS', 'CONFIGURATION', 'PERMISSIONS', 'RETENTION', and 'LIFECYCLE'. Under the 'OBJECTS' tab, there's a list of objects. The first object is 'Golf_Swing_Clean.mp4', which is 506.4 MB in size, of type 'video/mp4', and was created on 'Jul 17, 2021, 6...'. The storage class is 'Standard', and it was last modified on 'Jul 17, 2021...'. The public access is 'Not public', and it uses 'Google-managed key' for encryption.

Name	Size	Type	Created time	Storage class	Last modified	Public access	Encryption	Retention
Golf_Swing_Clean.mp4	506.4 MB	video/mp4	Jul 17, 2021, 6...	Standard	Jul 17, 2021...	Not public	Google-managed key	-

Async API Request

Google's VideoIntelligence_v1 Application Interface (API) parses video and applies a machine learning model to detect a person in the video. Specifically, this project analyzes a video of me at the driving range hitting 50 golf balls with a driver club on a sunny day in Iowa. The API **VideoIntelligenceServiceAsyncClient** ([google cloud client libraries](#)) uses the `annotation video` method to process a preconfigured request file that contains all the arguments necessary to asynchronously parse the video to detect a person and their movements.

To setup the request, first it is necessary to configure the virtual environment with an access token. Run the command `export GOOGLE_APPLICATION_CREDENTIALS="alpha-numeric_key.json"` with the credentials pointing to the key file downloaded earlier and now saved in the virtual environment. Once the credentials have been setup create file named `request.json` and copy the code below as it contains the parameters that are used to trigger the annotation by the google service. The script can be found in the Google API [documentation](#). The two most important parameters in this script are the `inputUri` and `features`. The first parameter `gs://alvarez-golfswingml-bucket/Golf_Swing_Clean.mp4` is the cloud storage bucket and video file to be processed by the annotation algorithm. The second parameter denotes the type of annotation which in this case is `PERSON_DETECTION`. Note, the google API also has methods that can run facial recognition, track objects, transcribe videos, and much more.

Contents of `request.json` file: source: [Cloud Video Intelligence API](#).

```
{
  "inputUri": "gs://alvarez-golfswingml-bucket/Golf_Swing_Clean.mp4",
  "features": ["PERSON_DETECTION"],
  "videoContext": {
    "personDetectionConfig": {
      "includeBoundingBoxes": true,
      "includePoseLandmarks": true,
      "includeAttributes": true
    }
  }
}
```

Using the terminal proves to be the fastest way of processing the video and has the least number of dependencies. When compared to setting up the google cloud Software Development Kit (SDK) a lot less work is needed to process a video. However, since this project is only running batch processing it is not necessary to setup a full data pipe-line with the SDK. With more time and effort, the docs clearly explain how this can be accomplished programmatically using languages like python, java and node.js. From the terminal run the following curl command to send a `POST` request to the `annotation_video` method.

Send video [annotation request](#)

```
curl -X POST \
-H "Authorization: Bearer $(gcloud auth application-default print-access-token) \
-H "Content-Type: application/json; charset=utf-8" \
-d @request.json \
https://videointelligence.googleapis.com/v1/videos:annotate
```

Stumbling Block: “What is...name?”

The process will take less than a minute to register with a `name`. While trying to complete this project programmatically using python and jupyter notebook this name detail was elusive. Much of the API reference materials reference a name-string that can be used to retrieve the parsed data via a python [coroutine](#). After stumbling on not being able to find the string-name for quite some time I regrouped by researching how to perform this operation using the Command Line. From here I was able to find that the project name-string is returned by the `videos:annotate` method and in my case it was configured with information I knew except for the unique 24 digit operation id. In order to retrieve the information, the API again provides a shell script that can be run after it is configured with the specific information to retrieve the annotations.

Response name-string

```
{
  "name": "projects/PROJECT_NUMBER/locations/LOCATION_ID/operations/OPERATION_ID"
}
```

Configure Response command:

- PROJECT_NUMBER: found in google cloud project
- LOCATION_ID: data center location selected during bucket creation e.g., us-west1
- OPERATION_ID: unique 24-digit code to identify the long running operation

GET response command

```
curl -X GET \
-H "Authorization: Bearer "$(gcloud auth application-default print-access-token) \
https://videointelligence.googleapis.com/v1/projects/82932726978/locations/us-west1/operations/17480310946357676047 > ML_out.json
```

Given more time I believe I can write a python program to automate video annotation using the Google Cloud’s VideoIntelligence_v1 API. Also, an interesting evolution of this project is to use the beta version of this API and annotate video in real time to get instant feedback on golf swing.

Summary access token, POST, and GET commands

```
(my_Virtual_Env) (base) luisalvarez@Luiss-iMac my-folder-name % export GOOGLE_APPLICATION_CREDENTIALS="golfswingml-b92ada60eb03.json"
(my_Virtual_Env) (base) luisalvarez@Luiss-iMac my-folder-name % curl -X POST \
-H "Authorization: Bearer "$(gcloud auth application-default print-access-token) \
-H "Content-Type: application/json; charset=utf-8" \
-d @request.json \
https://videointelligence.googleapis.com/v1/videos:annotate
{
  "name": "projects/82932726978/locations/us-west1/operations/12834128508130435308"
}
(my_Virtual_Env) (base) luisalvarez@Luiss-iMac my-folder-name % curl -X GET \
-H "Authorization: Bearer "$(gcloud auth application-default print-access-token) \
https://videointelligence.googleapis.com/v1/projects/82932726978/locations/us-west1/operations/12834128508130435308 > ML_out.json
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload    Total   Spent    Left  Speed
100  427    0  427    0    0    747    0 --:--:-- --:--:-- --:--:--   746
(my_Virtual_Env) (base) luisalvarez@Luiss-iMac my-folder-name % ls
ML_out.json      golfswingml-b92ada60eb03.json  my_Virtual_Env  request.json
(my_Virtual_Env) (base) luisalvarez@Luiss-iMac my-folder-name %
```

Data Analysis with Python and or Jupyter Notebook

After collecting the annotations results the data is processed with Jupyter Notebook. Specifically, the data is analyzed by mapping two vectors originating from the left elbow. One vector points to the left shoulder and the other to the left wrist. The angle of the elbow is calculated using high school geometry. The notebook contains code that pulls out the 2D coordinates for each point on the arm during a swing and along with time series data to a CSV file for further analysis. The code can be refactored to calculate other interesting angles, such as the position of legs, hips, eyes, and nose. However, this analysis focuses on the left arm and elbow during the takeoff and downswing during a drive. The analysis will also compare a pro golfer and me a weekend warrior golfer.

The file `Luis_best_swing.csv` contains the `left_elbow_angle` for a 200 yard drive that can be considered the best drive out of 50. The golf swing can be broken down into two phases after the address of the ball. The takeaway phase is the wind up of the golf club. At address my left arm is at its straightest and is about 187 degrees at the elbow and is basically straight. Midway through the takeaway the left arm starts to bend at the elbow significantly with an angle of about 201 degrees. At the top of the swing just before the downswing phase my left arm is about 263 degrees. The downswing is the fastest part of the swing and takes place in tenths of a second. A good high-speed camera captures more detail. Right before contact the left elbow is at a respectable 207 degrees and ends in 197 degrees just past the point of contact.

At most driving ranges there are markers that allow for the estimation of the distance the golf ball travels. In this particular day, my average swing was about 140 yards and my best driving being about 200 yards. In order to understand how to hit the ball further I analyzed some high-speed footage of professional golfer Tiger Woods. At address both Tiger and I have our arms in similar positions with our left elbow angle close to 190 degrees. However, that is where the similarities end. Once Tiger starts his takeaway it is very evident that his left elbow angle is constantly close to 190 degrees. In contrast, my left arm bends at angles greater than 200 degrees. One startling surprise was observed midway through Tiger's swing.

Tiger Vs Luis



Tiger Left elbow angle: 135 deg

Luis Left elbow angle: 201 deg

The pro golfer's arms exhibit excellent flexibility. The left arm's wrist drops below the left elbow and shoulder almost as if the golfer was double jointed and registers a 135 degree left elbow! At this stage my arms are at a 201 degree angle. My left arm forms a magnitude vector that opens up while Tiger's opens down. This flexibility along with control and power are likely why the pro can hit the ball on average 290 yards.

Based on the annotated video, Tiger keeps his left elbow almost straight if not locked through his swing. Right before contact, Tiger's arms stay at a steady 185 degrees. The same is gracefully true after hitting the ball. The main takeaway from Tiger's swing is that he is able to generate energy by keeping his left elbow as straight as possible and even has a period where he forms an acute angle with this joint before he unwinds. This swing can aptly be described as a pendulum that starts in his shoulder and is extended by the head of the graphite club. Furthermore, in an attempt to emulate this golfer, I attempt to keep my left elbow as straight as possible while slowly progressing through the swing. Needless to say, the task is very challenging and gives the sensation of almost falling over. I think I will leave the long ball to the pros.