# Manual to use the Video Analysis Project

1. Below are the details of each of the folder of this project:
2. **Dataset Folder:** All the videos will be stored in this folder. Each folder inside it contains a video with a CSV file having YouTube video ID and playlist ID, and video title.

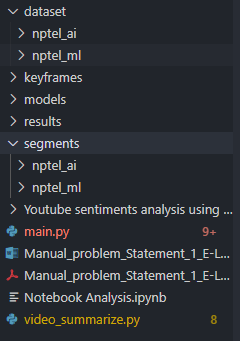
# Segment folder: All the videos segment will be stored in this folder with the same hierarchy as the Dataset folder.

# Keyframe Folder: If enabled in the saving using cv2 in code, it will load all the keyframes, which will only be available in this file.

# Results Folder: All the results, like all keyframes, detection, and the average of all the summarization, will be in this folder in the CSV file.

# YouTube's sentiments analysis using API v3: This contains the Jupyter notebook files that go from fetching data to preprocessing and sentiment model analysis for the videos we have put.

1. The next step is to run the Video summarization. You can refer to Notebook-Analysis.ipynb file.
2. We need to create the same folder structure in the ‘segment/’ folder as we have in the ‘dataset/’ folder because each video segment goes into those folders only. Similarly, we can create multiple folders to create a result of the number of video files.



# The below steps specify the flow of the code:

1. Refer to the Notebook Analysis.ipynb for a quick description of the tasks and what we could have written separately and what dependency needs to be installed.

The above step will handle 1,2,3 task number, the Result folder will contain the output.

1. Once done, we can move to task 4 that is sentiment analysis in the folder: Youtube sentiments analysis using API v3. The flow of this folder is:
   * + GETTING API DATA.ipynb -> to get all the video ID mentioned in the dataset file in CSV file format. Convert into Dataframe and save Dataframe in local for further analysis.
     + Data Cleaning.ipynb -> To clean the data and making the data ready to make a model out of it.
     + Sentiment Analysis.ipynb -> To make the model of the data we got from previous steps. Once done, this is the last task 4.

**Additional information:**

# Details about Haar Cascade algorithms:

Haar cascade is defined as a machine learning object detection algorithm used to identify objects, and the concept has been introduced in early 2001.

The cascade is trained over many positive and negative images to detect the objects, whether it is the face, hand, body, and even objects, etc.

Haar feature considers adjacent rectangular regions at a specific location in detection widows, sums up the pixel intensities in each region, and calculates the difference between these sums to detect the objects.

In exercises we are using 6 haar cascade algorithms:

1. haarcascade\_upperbody.xml
2. haarcascade\_fullbody.xml
3. haarcascade\_frontalface\_alt.xml
4. lbpcascade\_frontalface.xml
5. haarcascade\_lowerbody.xml

As the name suggests, each cascade algorithms detect the objects.

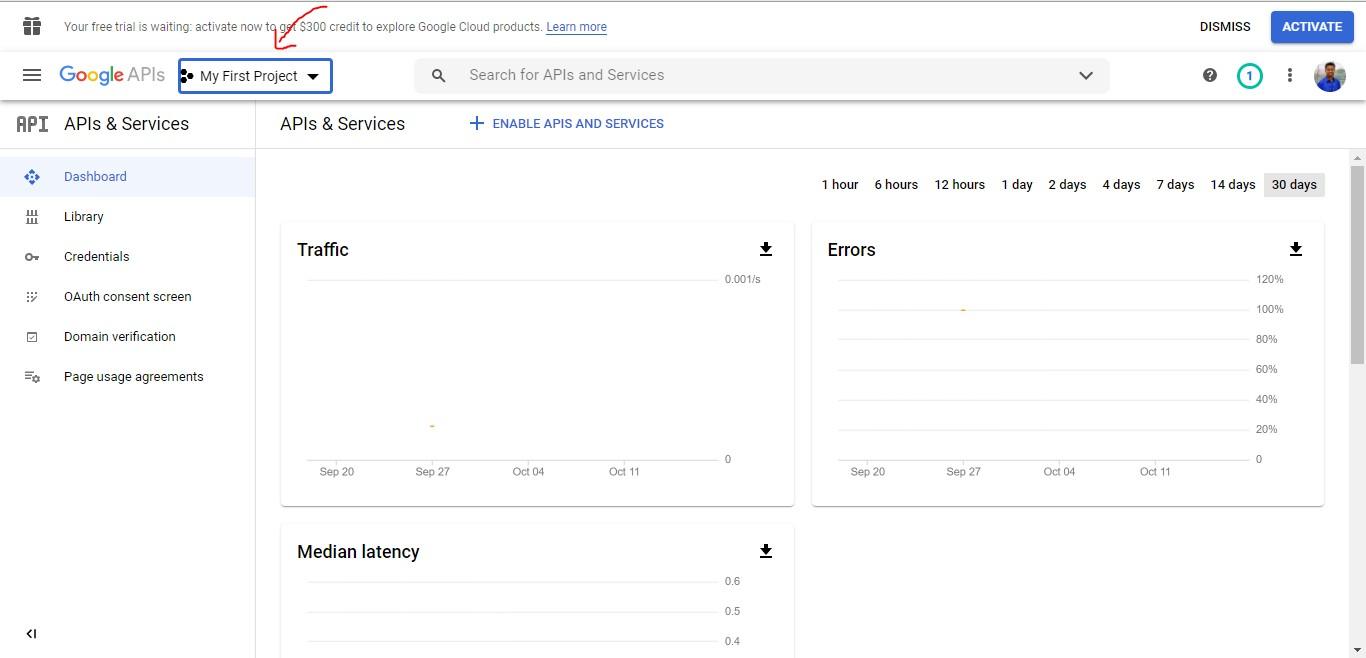
We are also using one pretrained model to detect text:

1) frozen\_east\_text\_detection.pb

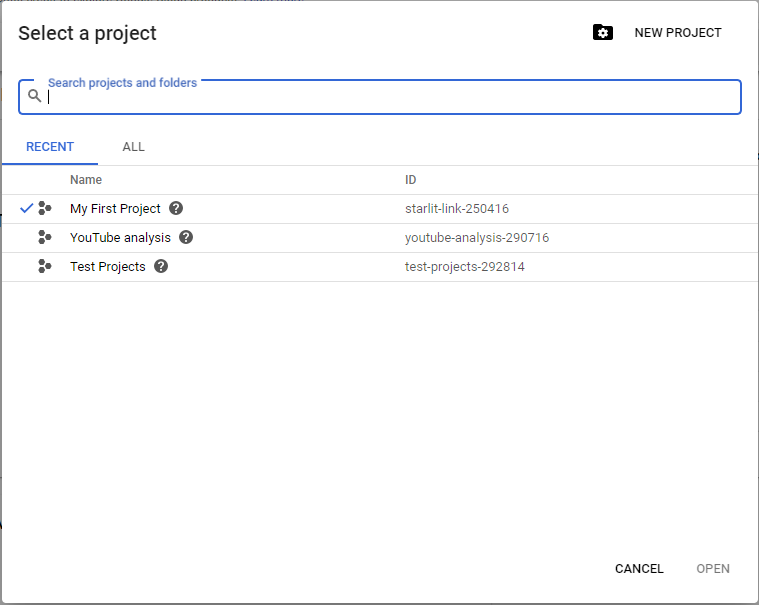
It is based upon the deep learning library that based upon the novel architecture and training pattern, it is have a very accurate text recognition capacity and running at real time 13 FPS on 720p images.

Task 4: To get the YouTube comments and replies, you need to generate your YouTube API token key:

1. Create your Google account
2. Go to google developer console: <https://console.developers.google.com/>
3. Log in with your Google account and select on the top left corner drop-down menu to create a new project:

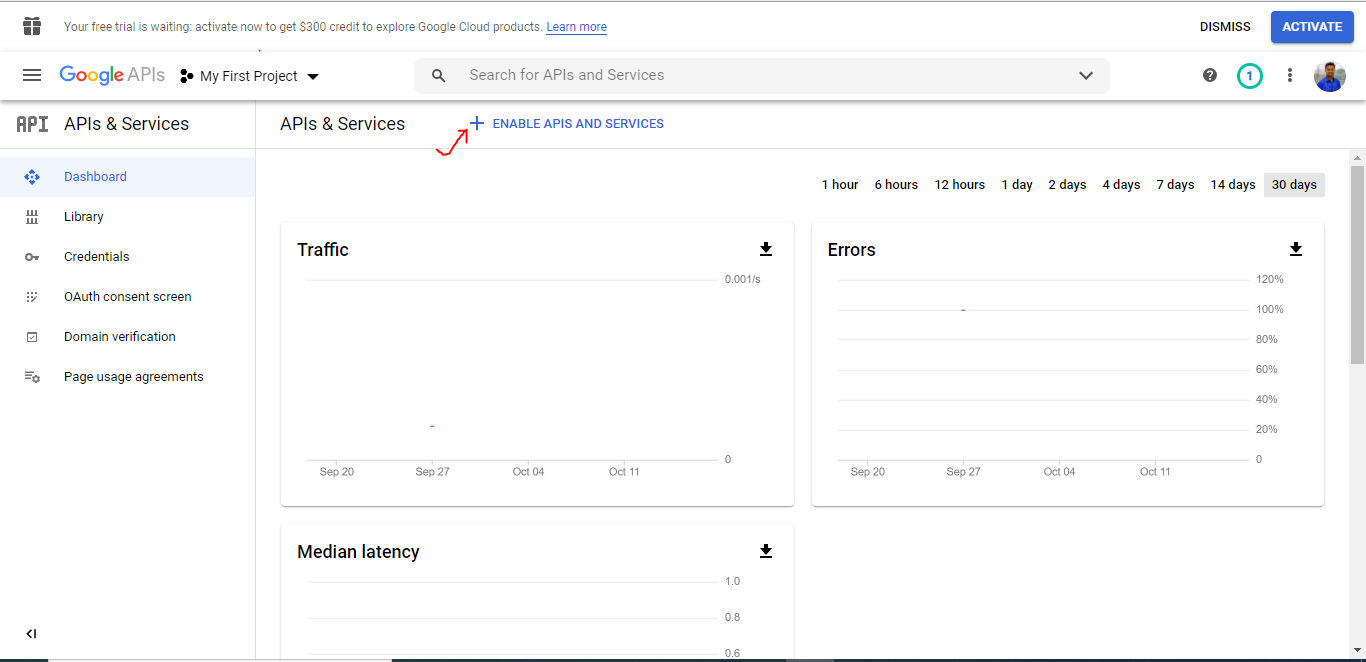


1. On the next screen, click on create a new project and name your project. Let's say, “Youtube Video Analysis.”

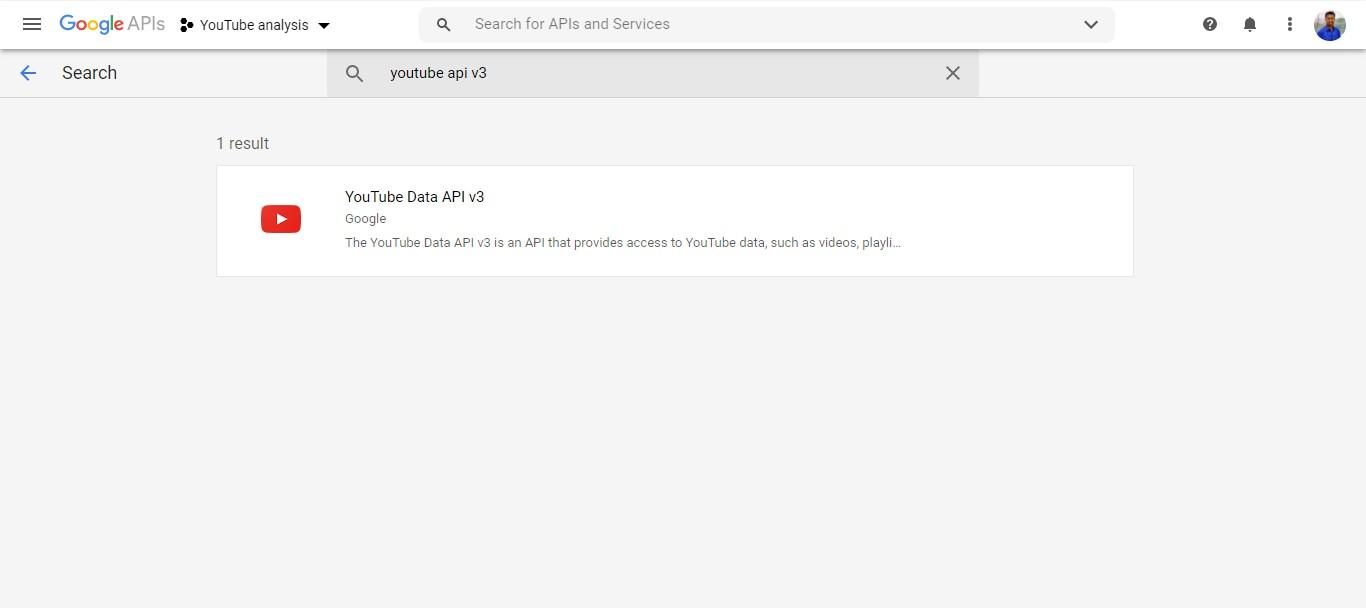


Just name it and ignore the other details.

1. Select the project from the drop-down menu you have created if you are not in the same project. And click “Enable API and Services.”

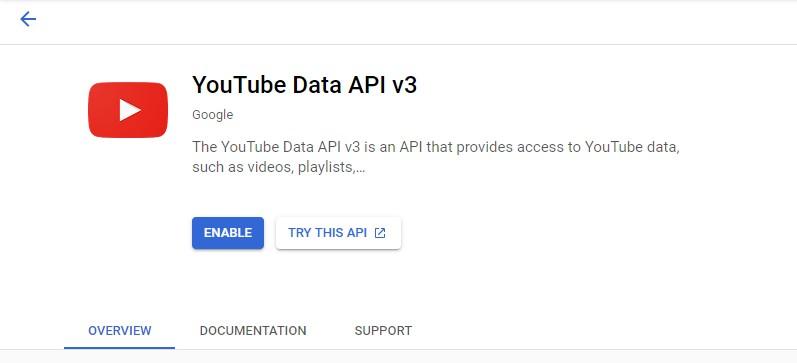


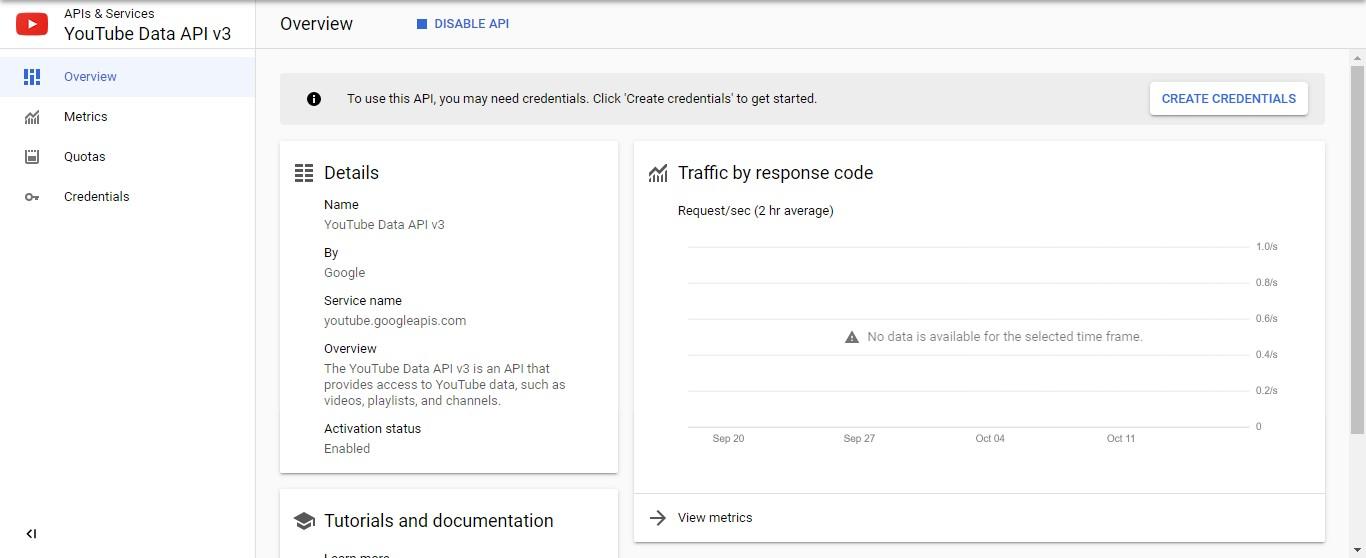
Now search your API in the search bar “youtube API v3.”



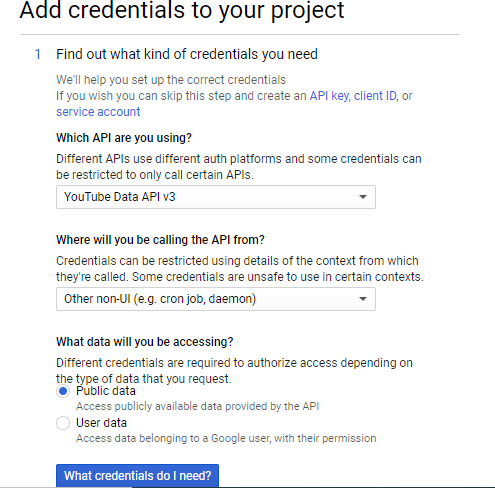
Select the result:

1. Click on the Enable button:



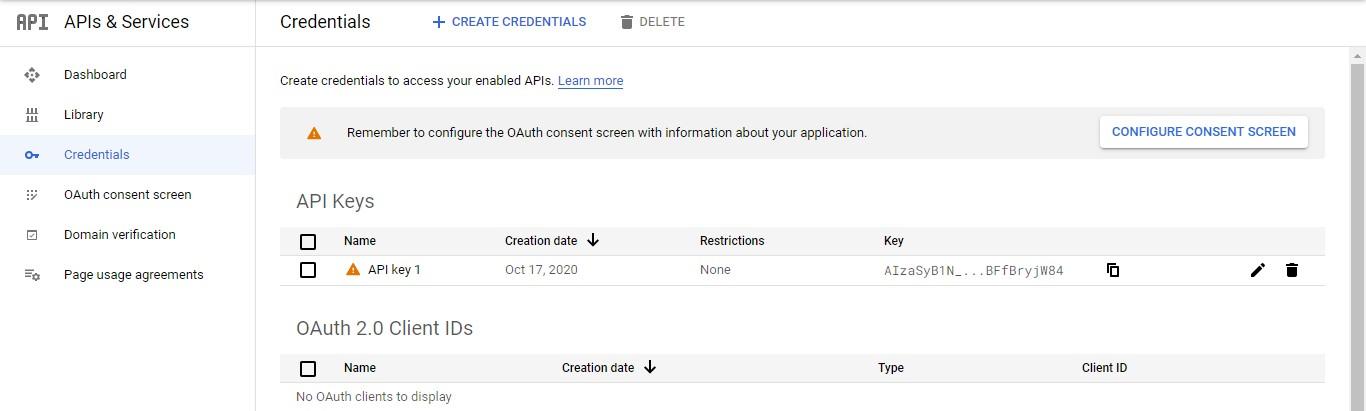
1. In the next step, select the create credential button:
2. Give the details
   1. Which API -> YouTube Data API v3
   2. Where you are calling API -> Other Non-UI
   3. What data will you be accessing -> Public Data

Once done, click “what credentials do I need.”



1. After this part, you will get your generated API, copy your API Key, and save in a secured folder. You can click done after it.

You can get the key to in case you lost it in the Credentials option in API and services.



Next, you can place your key and use it to run your project.

In this task, we are also using some libraries like demoji, langdetect. Some idea of these libraries are:

1. Demoji: It has a set of functions that accurately identify emojis in the text. We can use it to remove emojis and replace it with null values.
2. Langdetect: It detects language based upon the google language detection algorithm. We can use it to detect language English and remove other languages.
3. TextBlob: More details you can get from here: <https://textblob.readthedocs.io/en/dev/> It is an API for the NLP task, part of speech tagging, sentiment analysis, noun phrase extraction, and more.

We can determine the polarity of the text. If the polarity is > 0, it means it is a positive comment or replies else vice versa.