NEURAL NETWORK PROGRAM
USER MANUAL

[55 Water Street 9th FL, New York, New York, 10004]

[hsleenet@gmail..com]

1. INTRODUCTION

This document lets you know how to use our program step by step.
Welcome to the Neural Network Team!

2. GENERAL FEATURES

On our program, You can

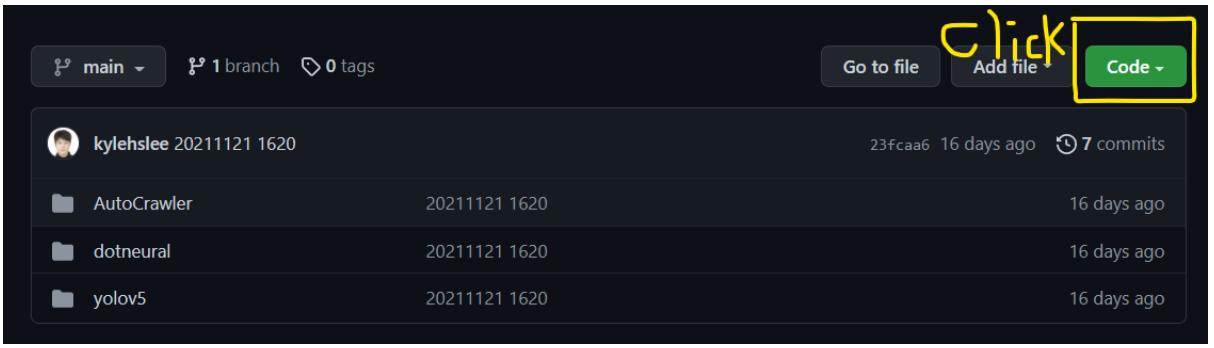
- Search assets, which are already stored on our database server.
- Retrain our model with new assets and detect those assets on street images if not on our server.

3. PREREQUISITE

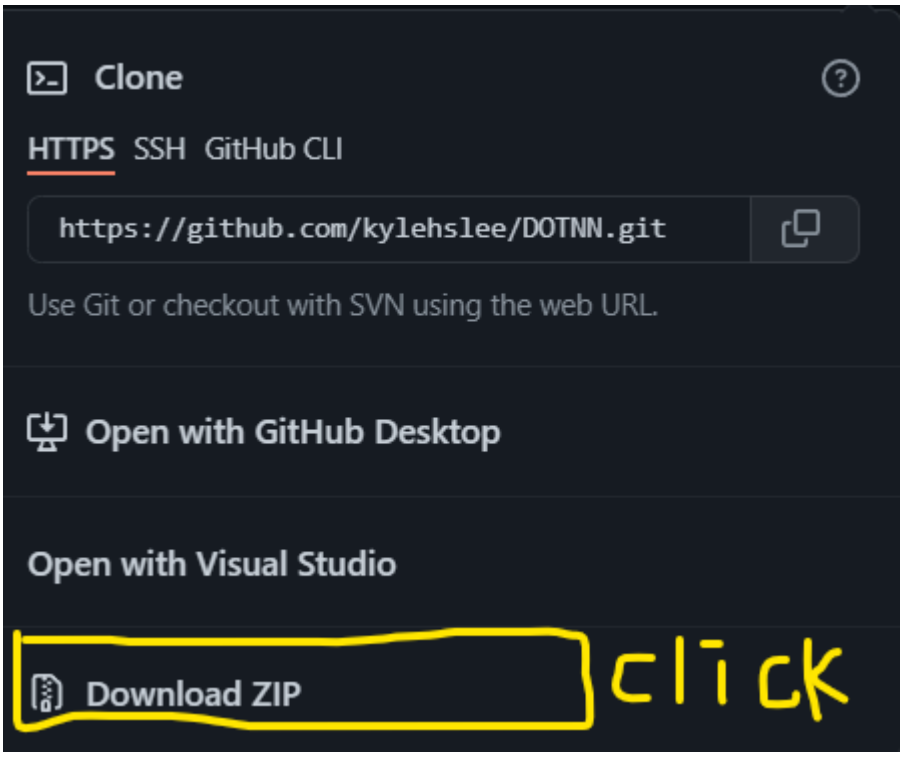
1. Only Windows operations are supported. (No Macbooks, iMacs, Linux and so on)

2. Make sure you have installed Google Chrome on your computer. It will be needed for our autocrawler. https://www.google.com/chrome/

3. Go to https://github.com/kylehslee/DOTNN

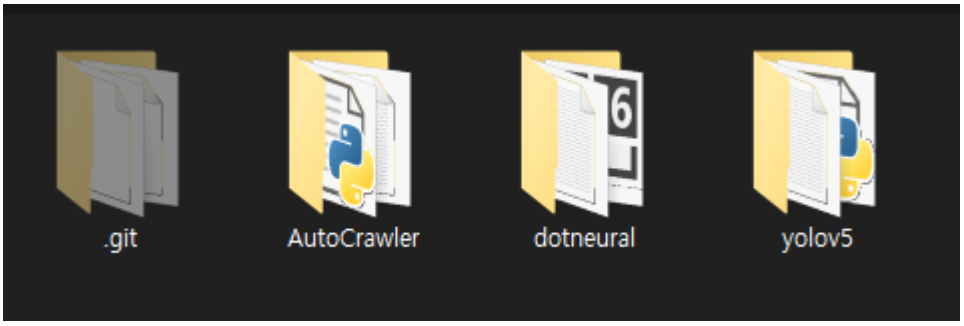


Click the “Code” green button.



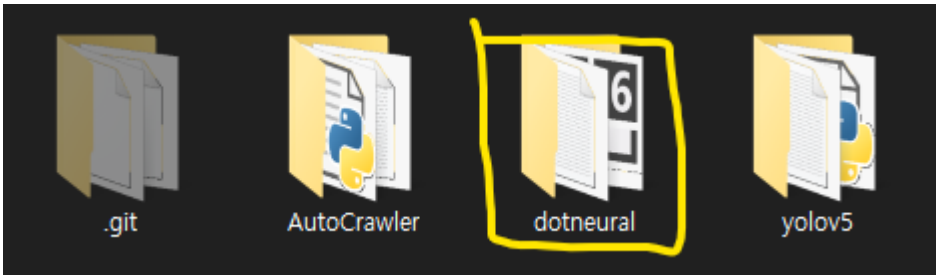
Click the “Download ZIP”. Then, you will successfully download our program.

4. Extract the zip file.

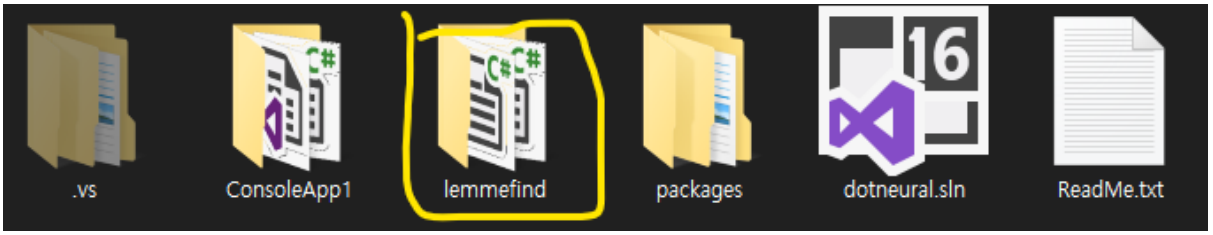


Now we are ready.

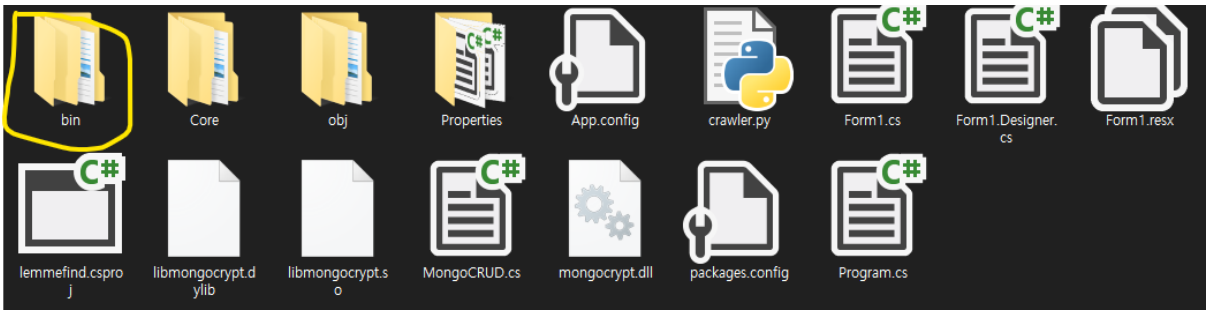
4. START PROGRAM



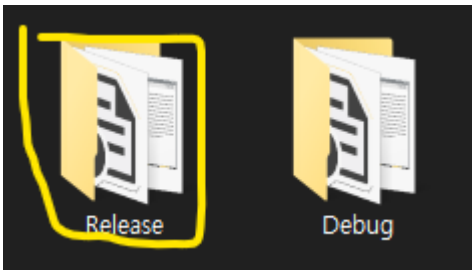
Click the “dotneural”



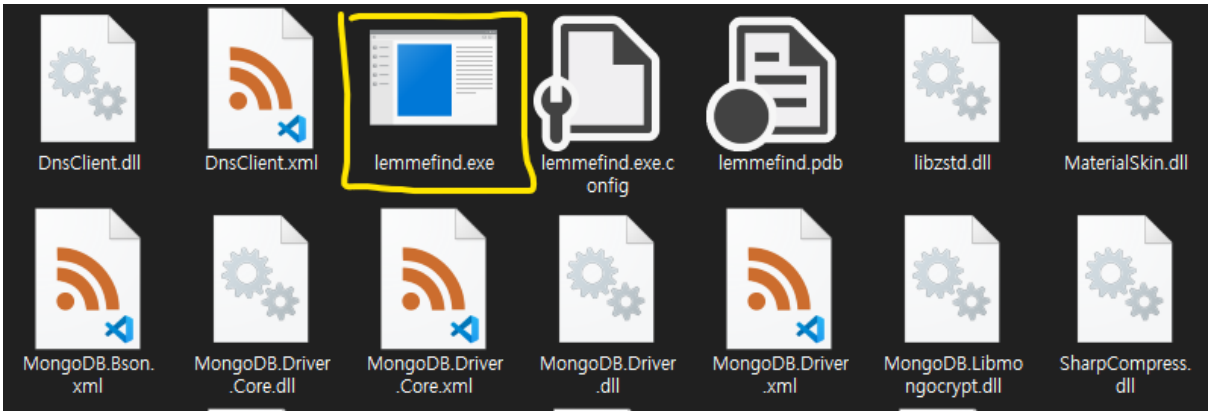
Click the “lemmefind”



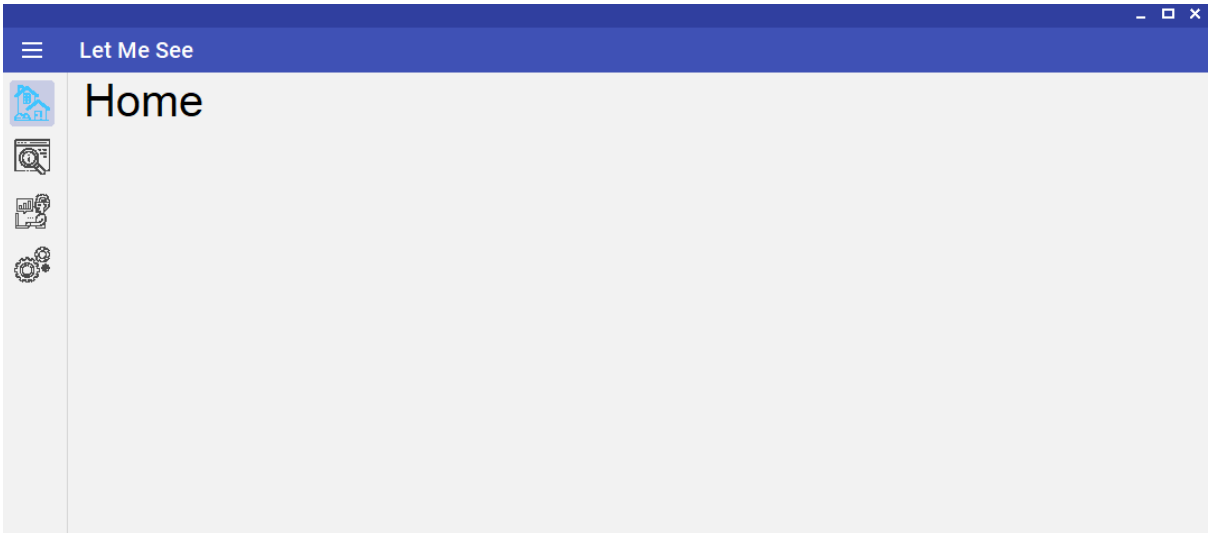
Click the “bin”



Click the “release”

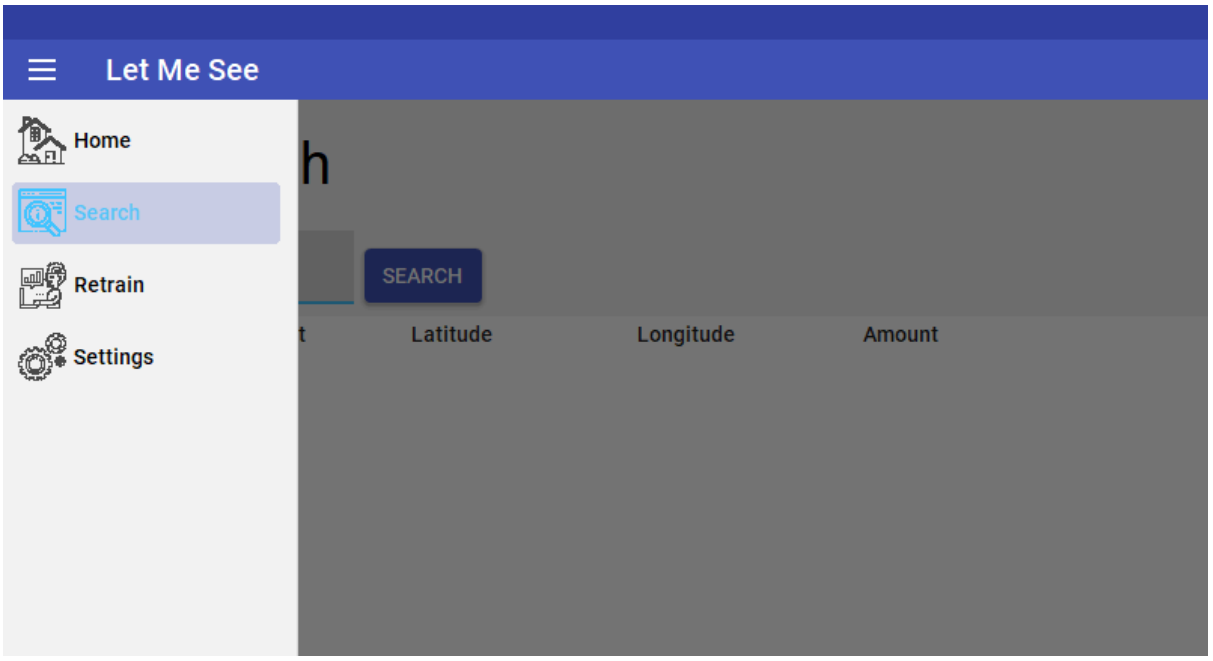


Launch “lemmefind.exe”.

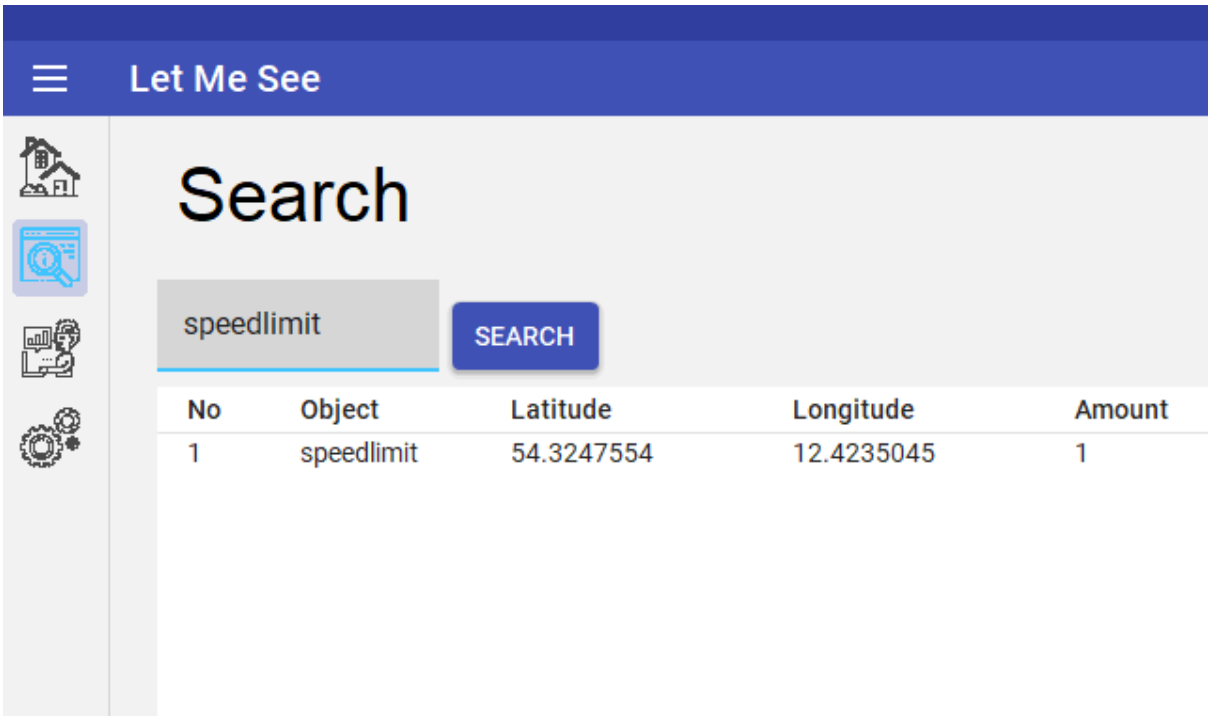


This is how you launch the program.

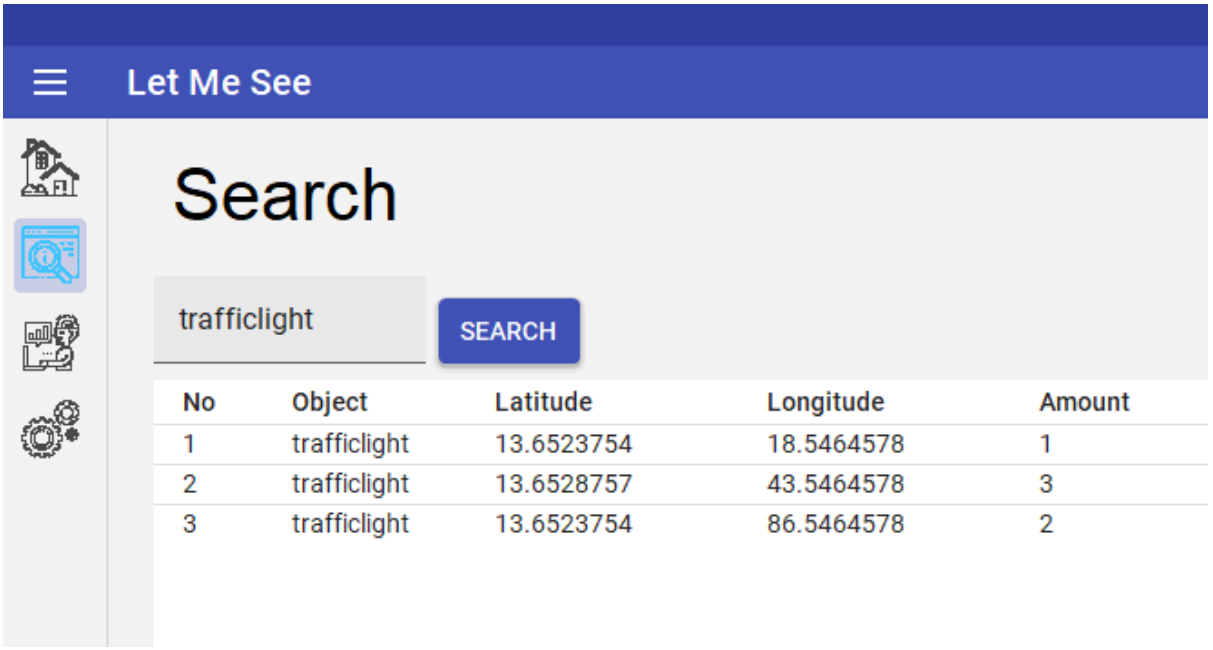
5. SEARCH EXISTING ASSETS



Go to the “Search” tap.



You can search assets which are already stored on our database. It seems like there is only one data for a speed limit sign. You can get latitudes and longitudes of assets. “Amount” column lets you know how many objects there are on that latitude and longitude.

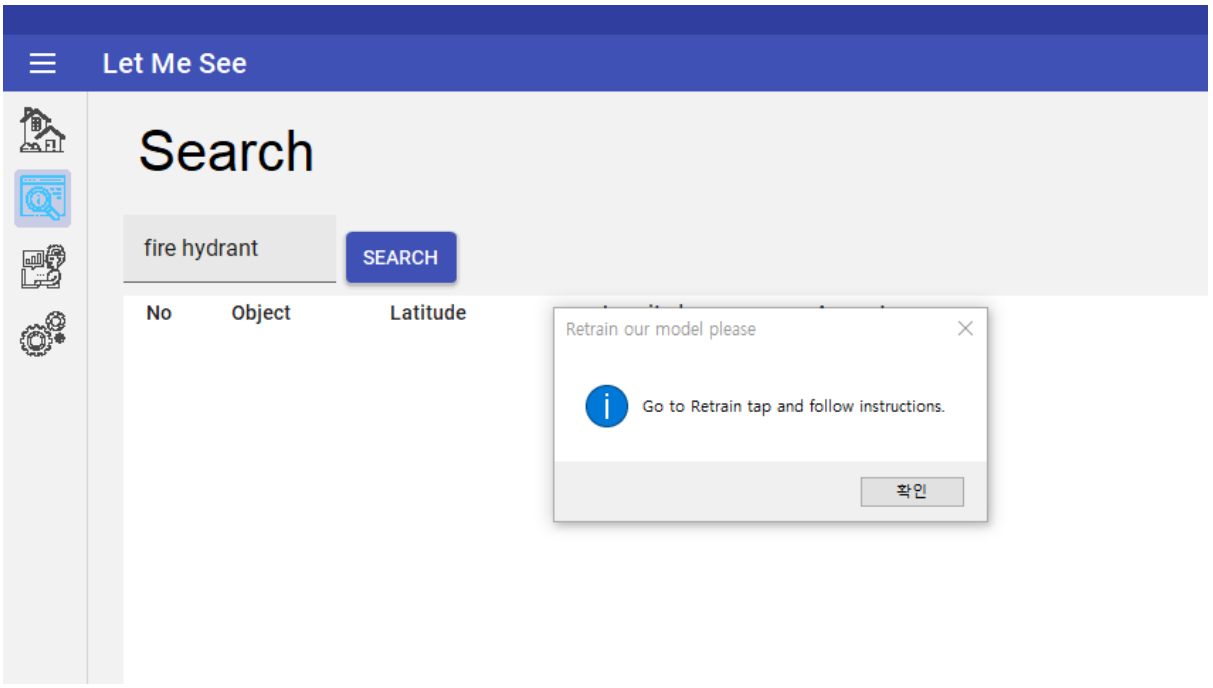


The results which are shown on the result box are demo data, so it might look weird. What I would like to tell you here is that when you see No 2 traffic light, its amount is 3, which means on that location there are three traffic lights detected.

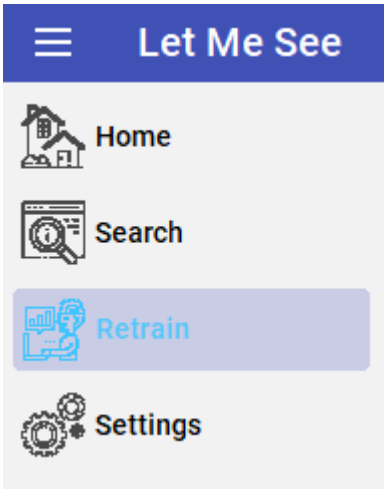
6. RETRAIN OUR MODEL

if there was not data that you were looking for, you can retrain our model with objects you like to find on the street and detect them on real street images.

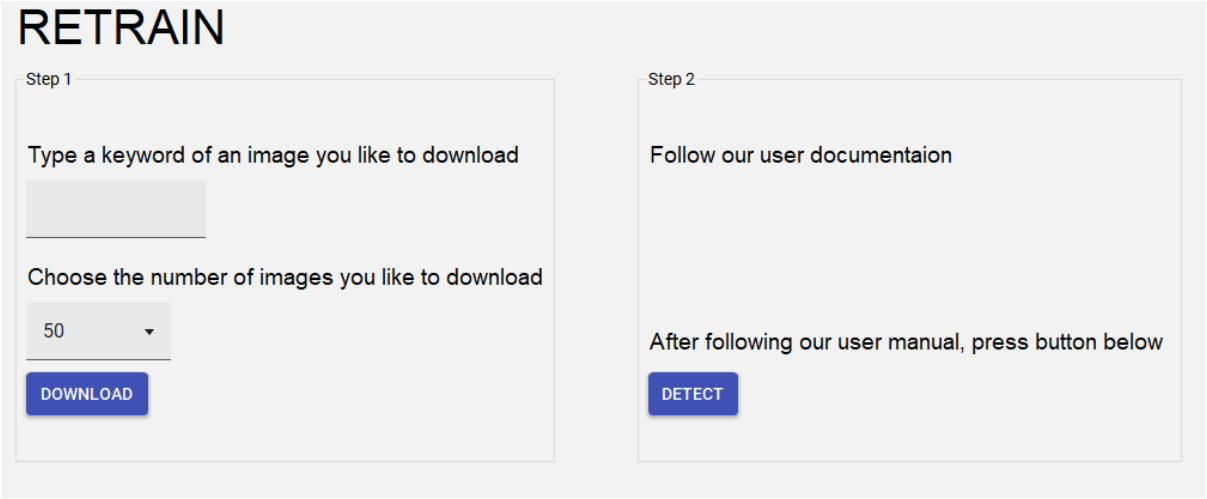
Let's say you would like to find information on "fire hydrants". You might type like this.



However, you would not see data because our database does not have such data. Don't worry.

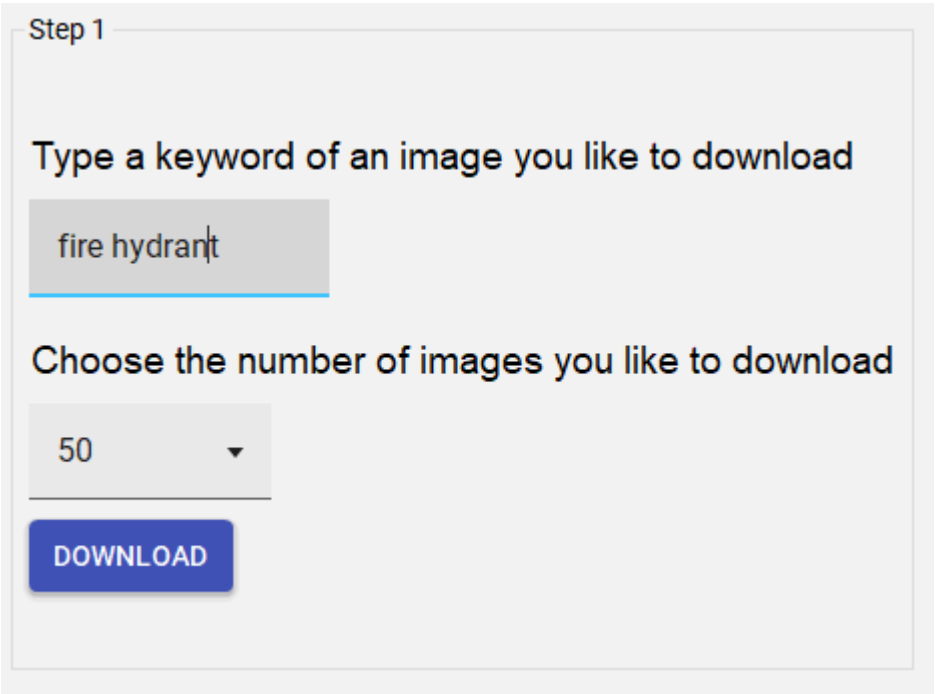


Go to the "Retrain" tap.



You will end up being here. From now on, it might be a little bit complicated to follow, but I will do my best to explain specifically.

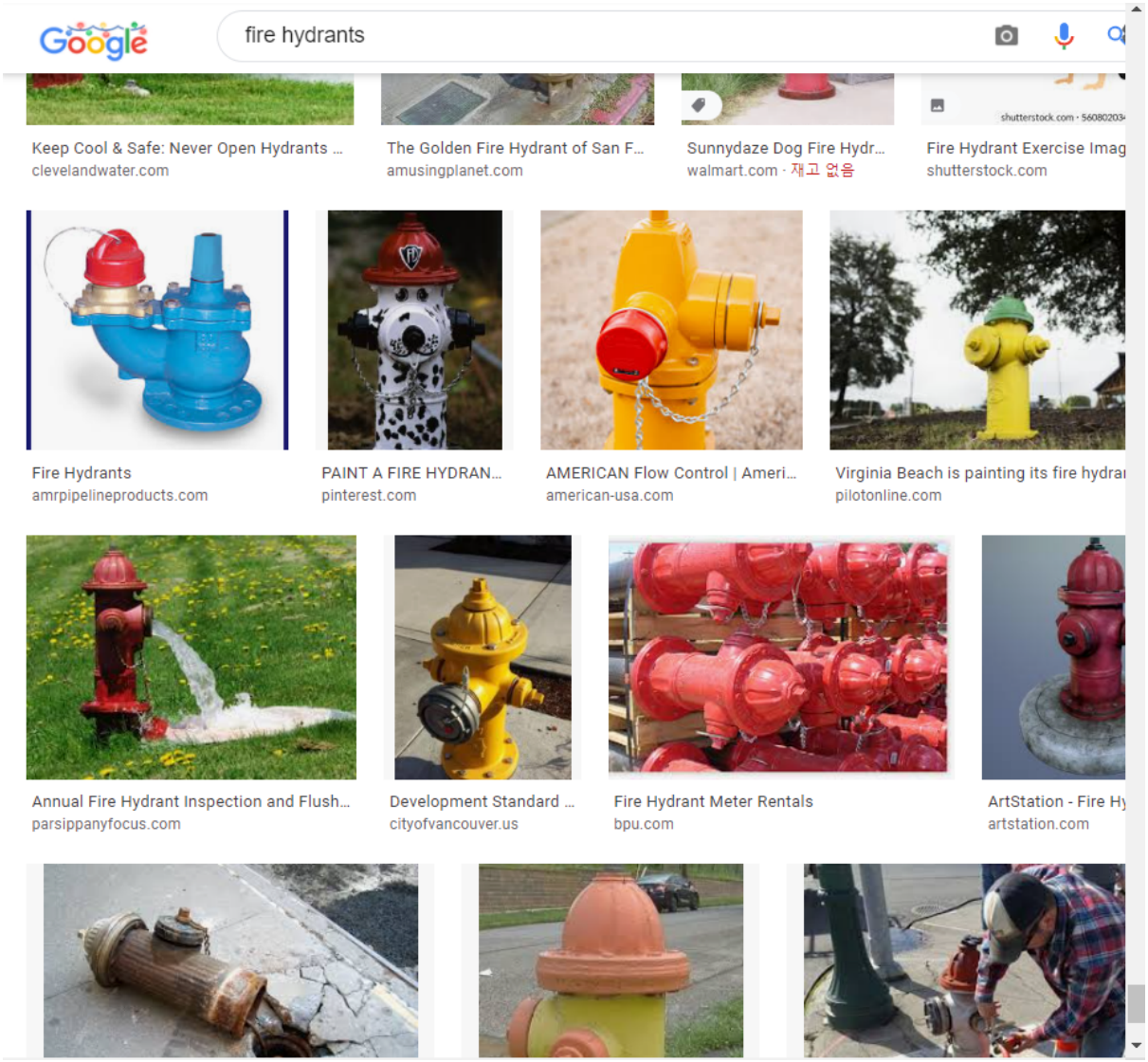
What you are looking for is a “fire hydrant”. If you have images of a fire hydrant, then you are good. However, if you do not, then you need to download images of it using our crawler. How? just type a keyword like below.



That’s it. Click the “download” button, then wait for about 1 minute.

Images will be downloaded in the “ AutoCrawler/download ” folder.

While our program downloads images, two things will come up on your screen.



The first thing is google chrome. It will look like it is doing something alone automatically.

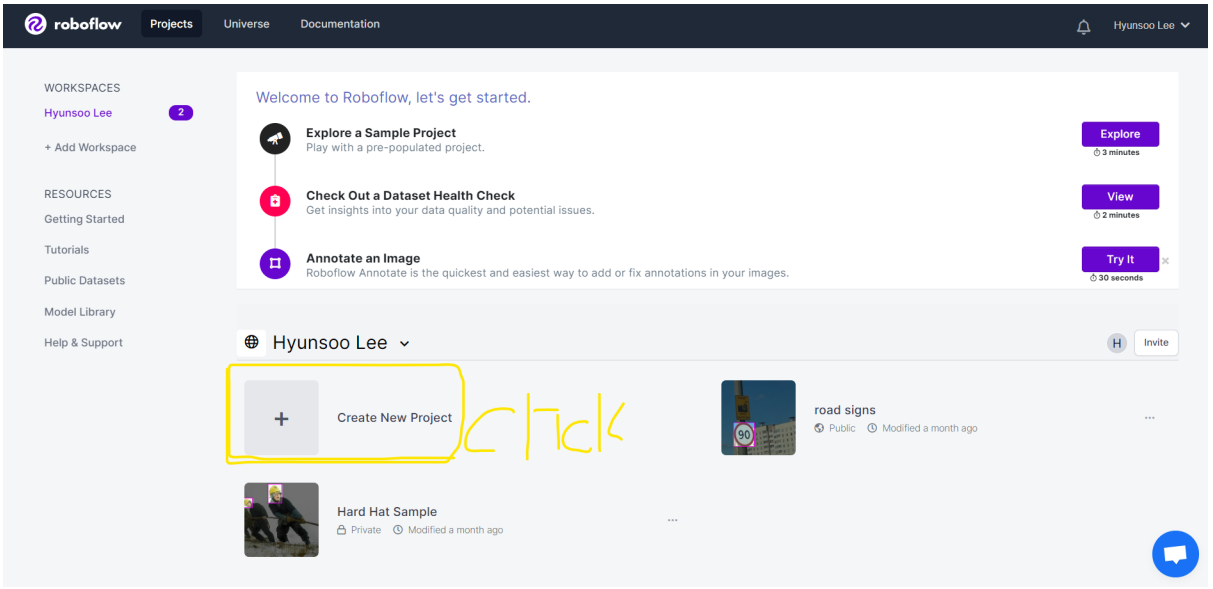
The second thing is a command prompt.

All you need to do is just wait for it to finish downloading. It will take about 1 minute for 50 images. After it finishes, those two screens will disappear automatically.

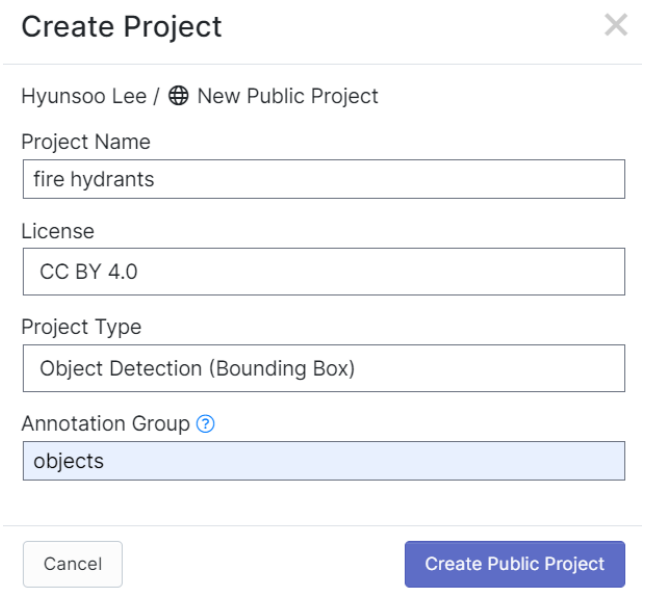
6.1. ADD ANNOTATIONS ON IMAGES

Go to the Roboflow website https://roboflow.com and sign in if you have an account, else you need to sign up. It is for free.

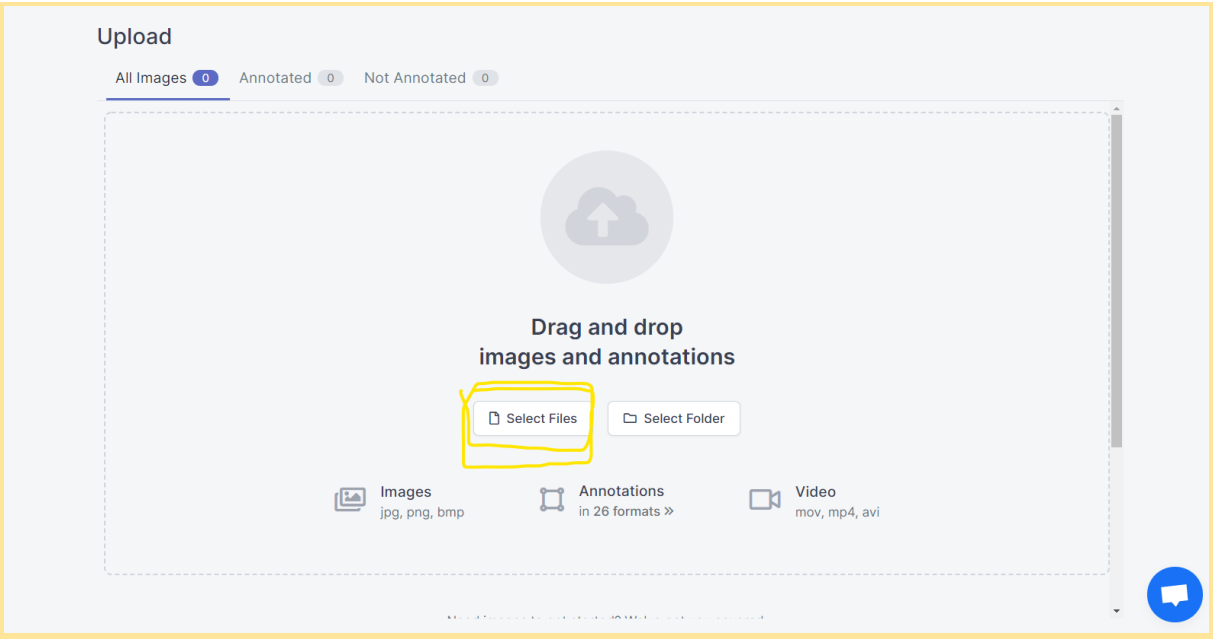
After you sign in,



Create new project

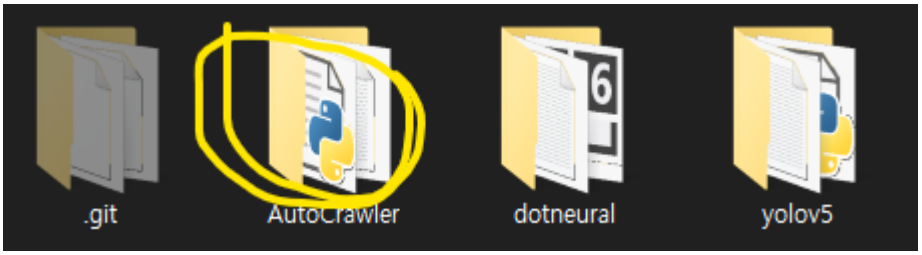


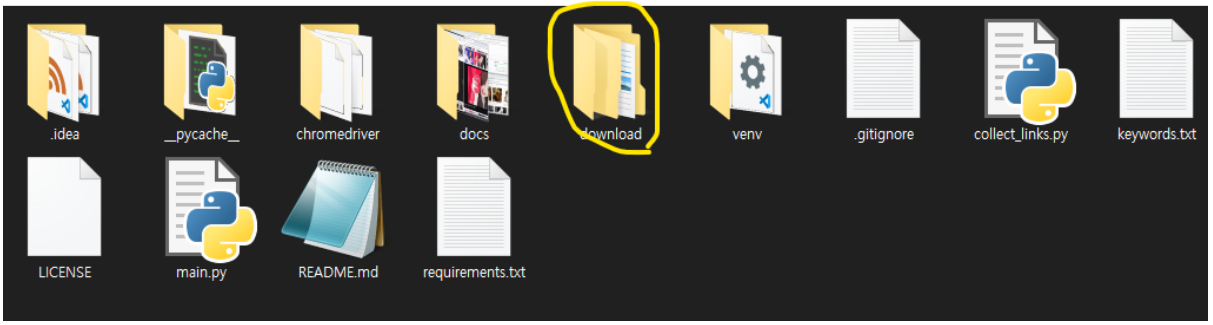
Set project name and annotation group.

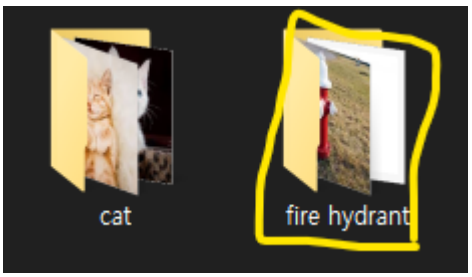


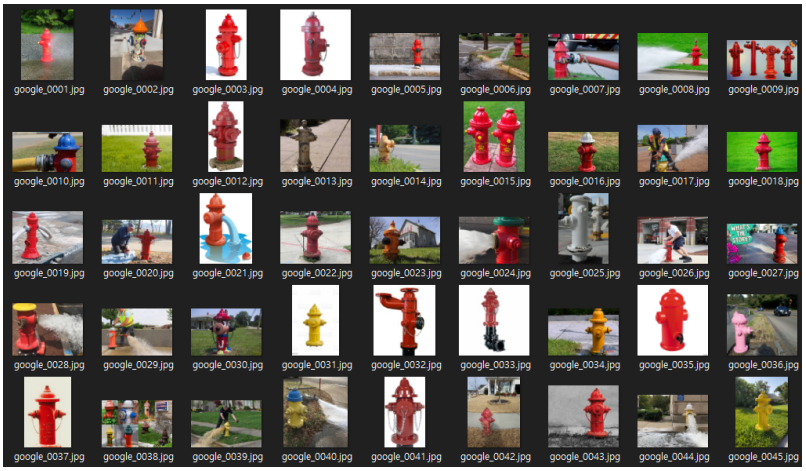
Click the “Selected Files” to load our fire hydrant images

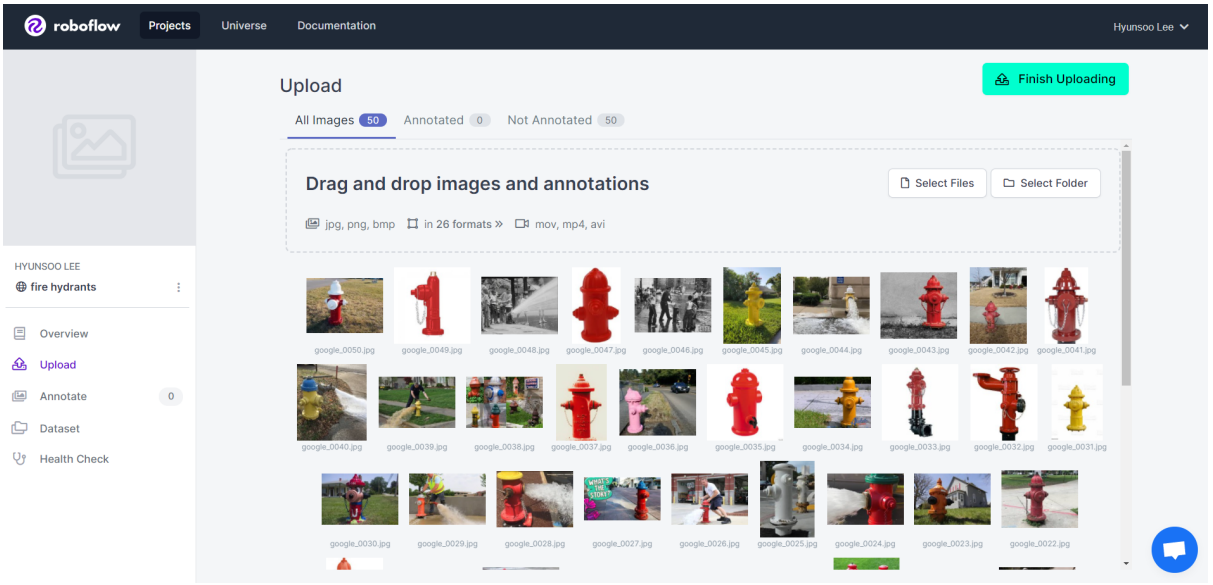
Go to “AutoCrawler/download/ WHATEVER YOU DOWNLOADED FOLDER “



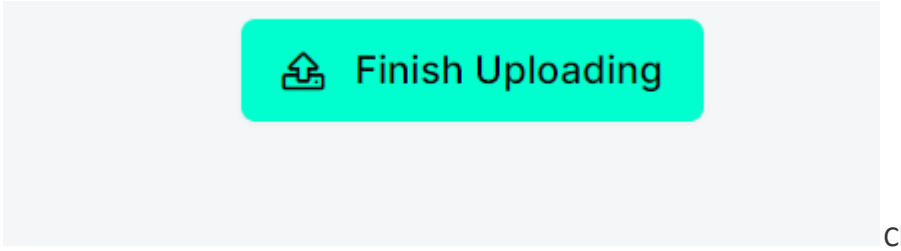




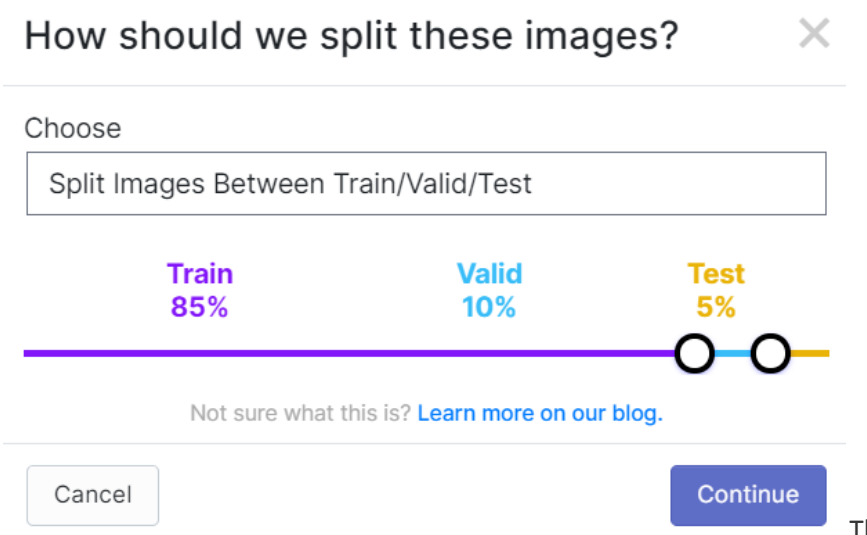

Select downloaded images.



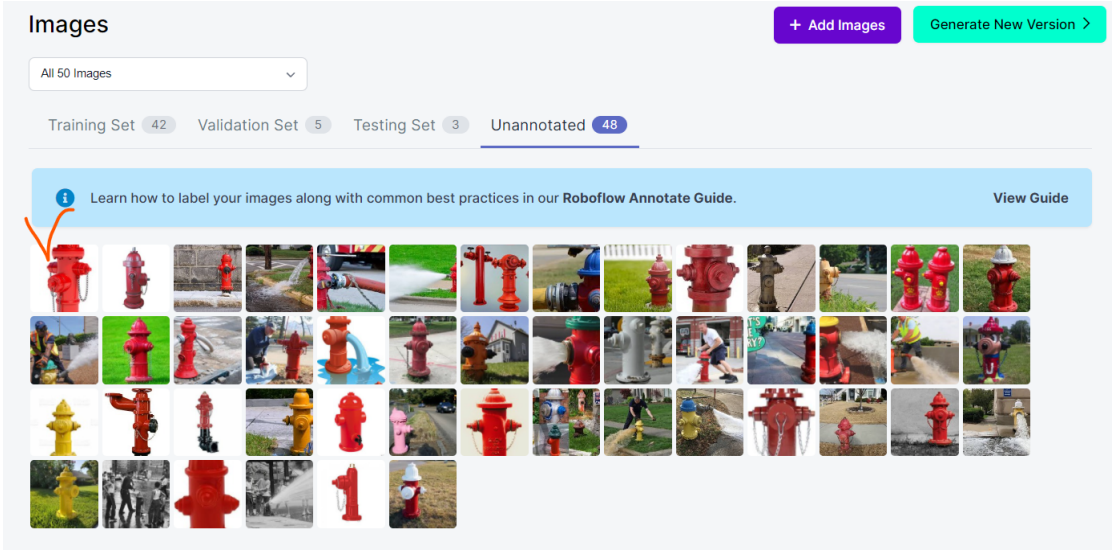
Now, we uploaded our images on Roboflow.



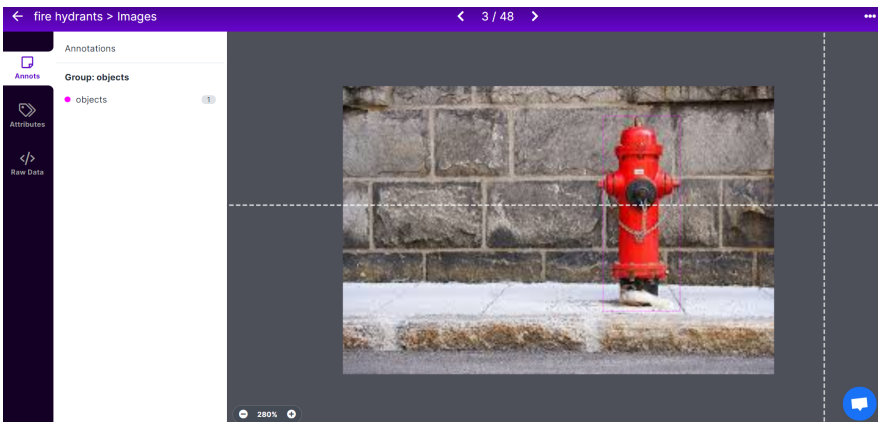
Click the Finish Uploading



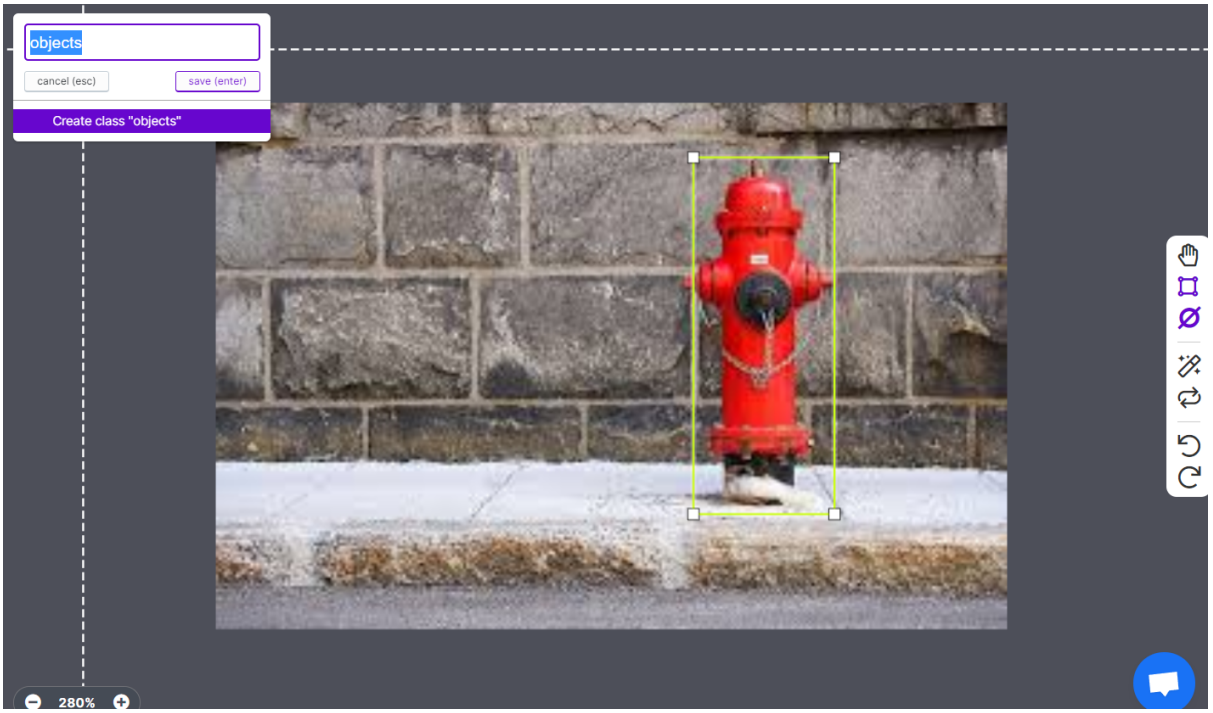
Then, click the continue.



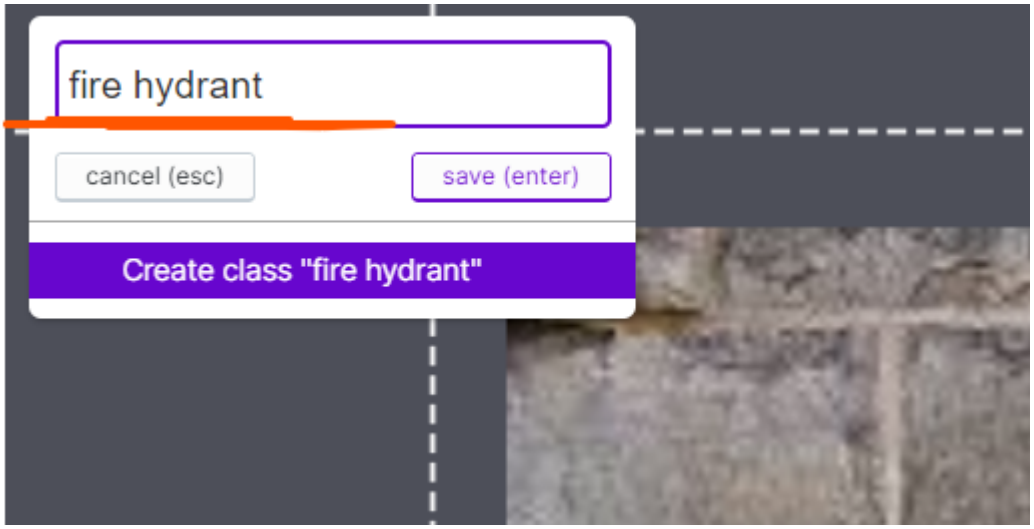
Click the first image



You will see this come up.



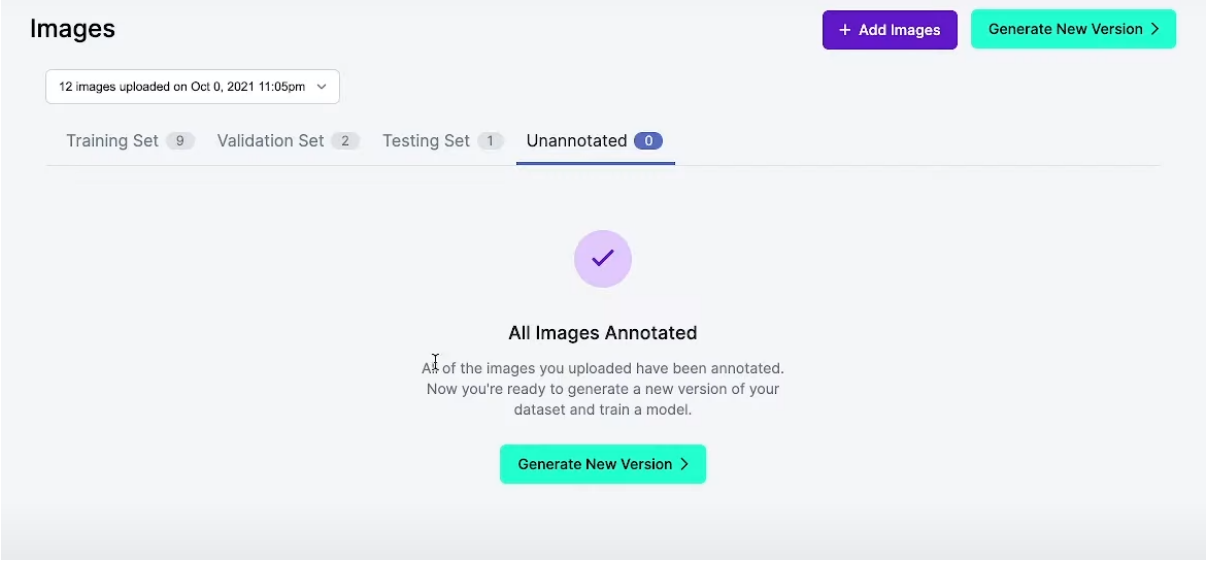
Now, drag only the “fire hydrant” part.



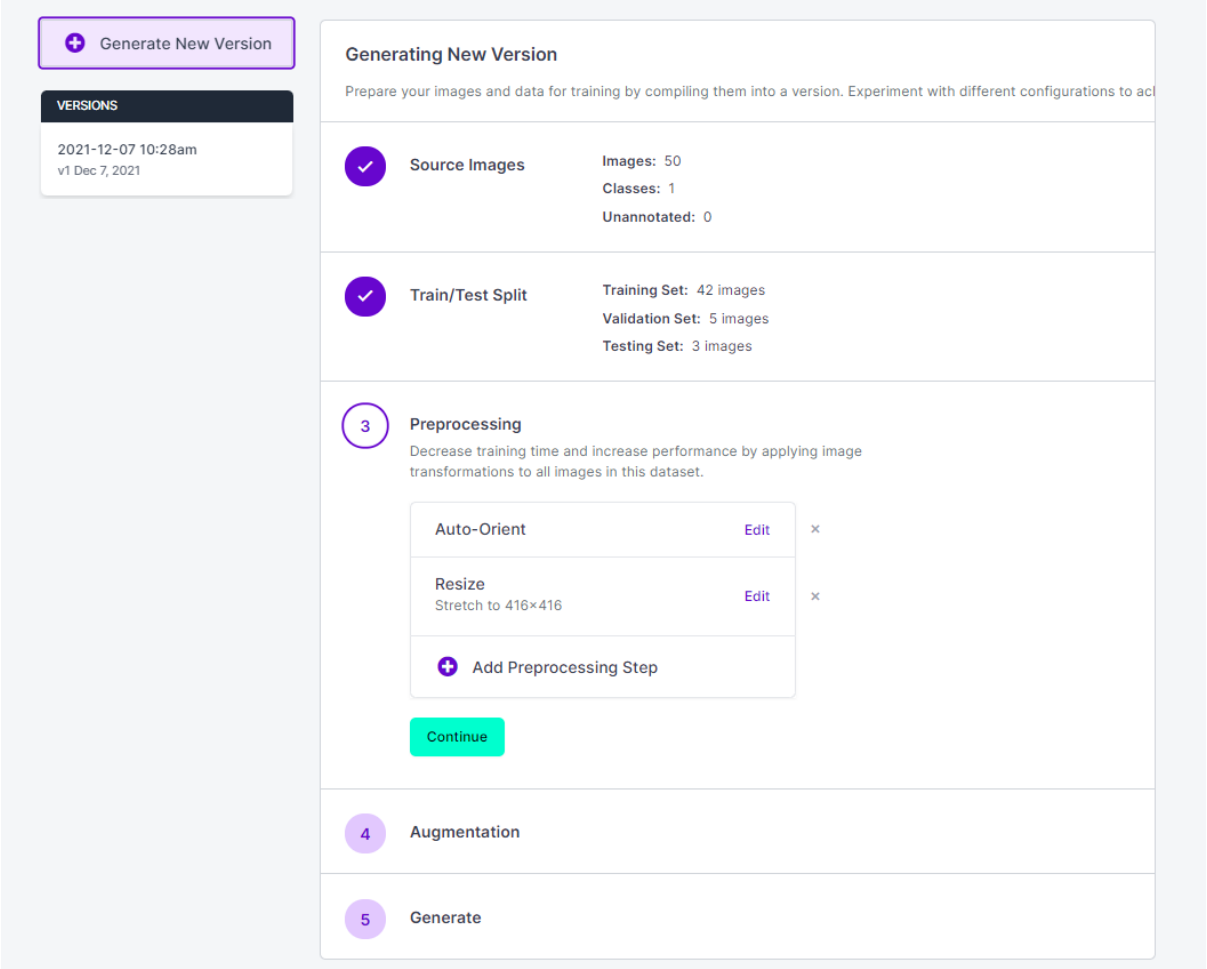
Then, save that bounding box with a name “fire hydrant” or whatever you like to call.

Do the same thing on all other images. The reason why we do this is because we only need to know what fire hydrants look like, but other parts like grass or cars.

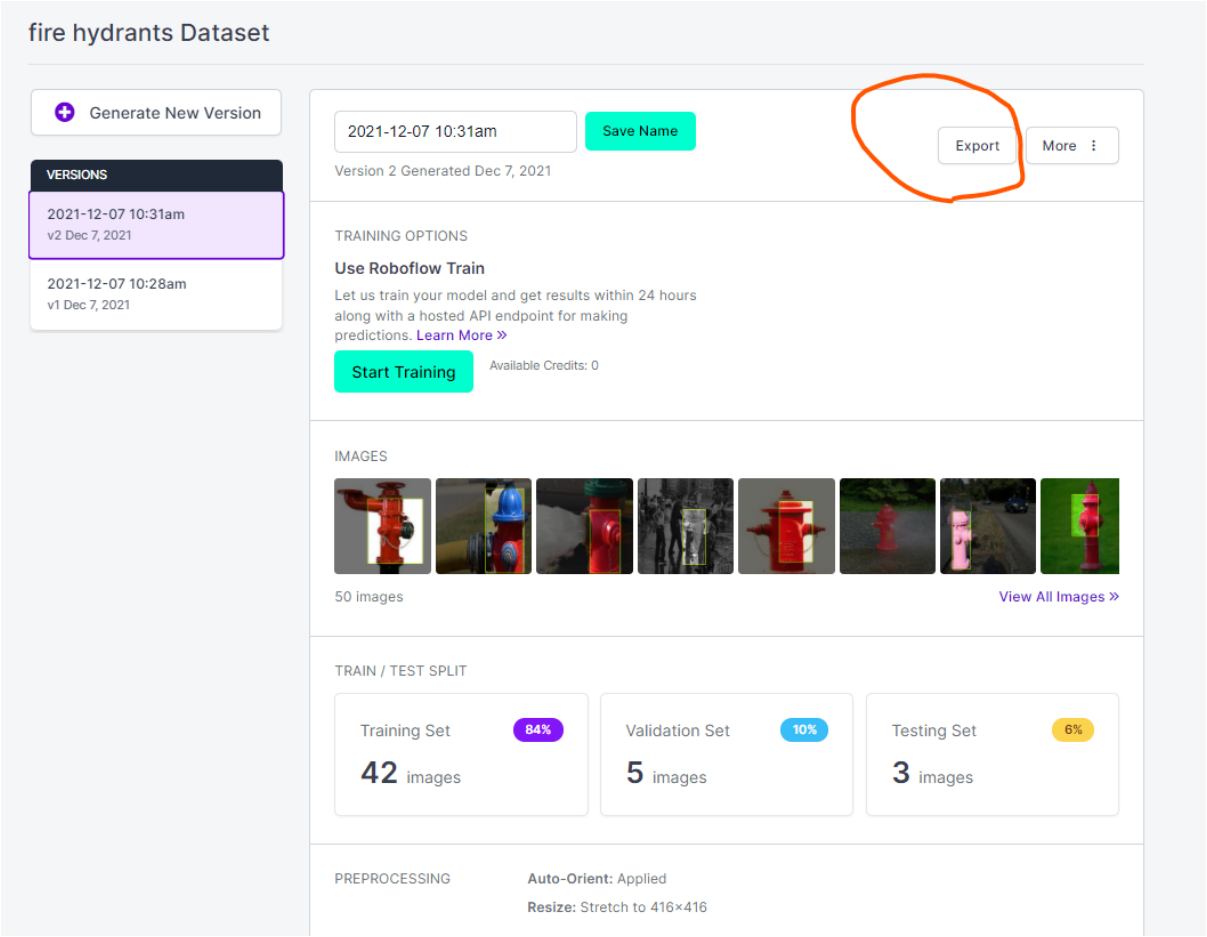
Once you add bounding boxes on all images, there will be nothing on the “Unannotated” tap.



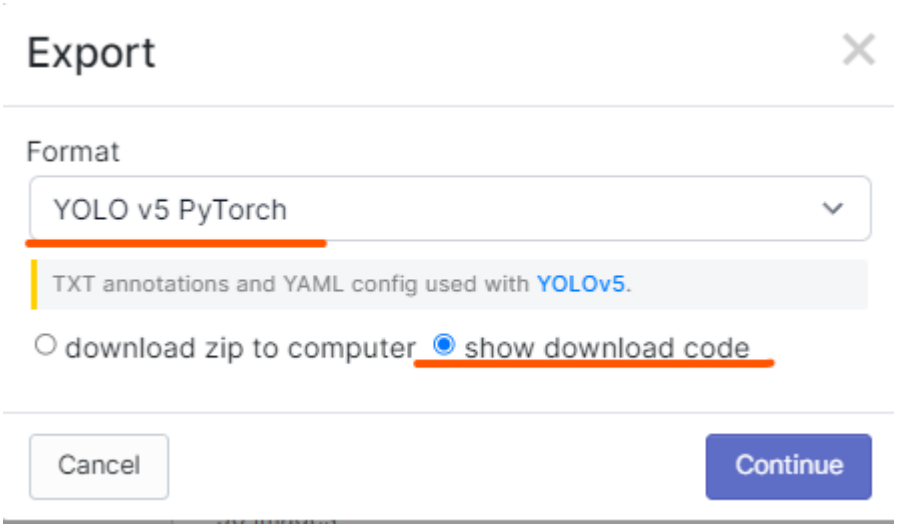
Click the Generate New Version.



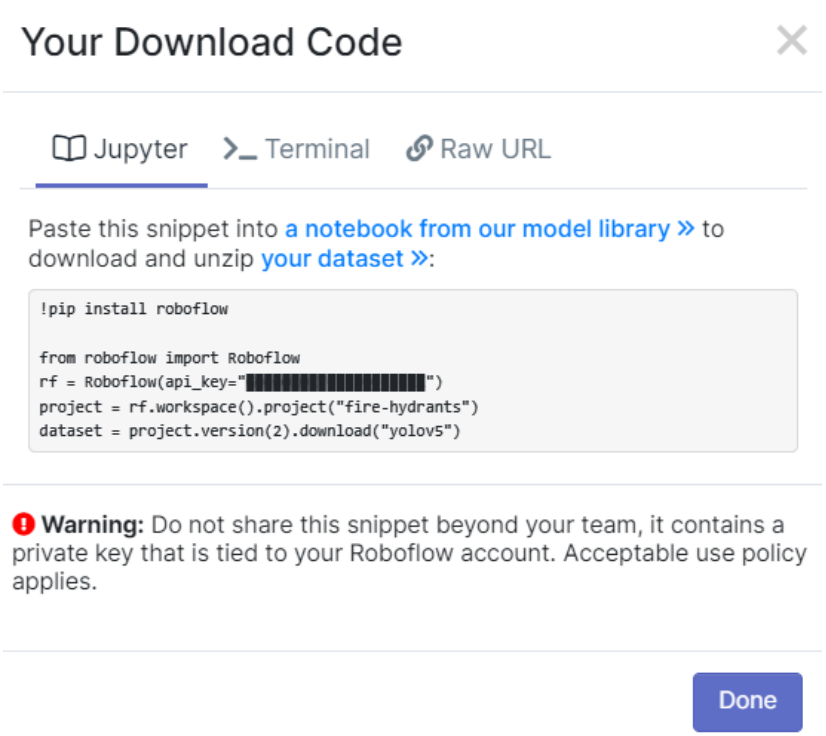
After you click “Generate new version”, you will see this. Just click continue on step 3, 4 and 5.
If you would like to know more about preprocessing and augmentation, here’s a link.
YOLOv5 + Roboflow Custom Training Tutorial



After it's generated, you need to click export.



Choose the format "YOLO v5 PyTorch" and "show download code"



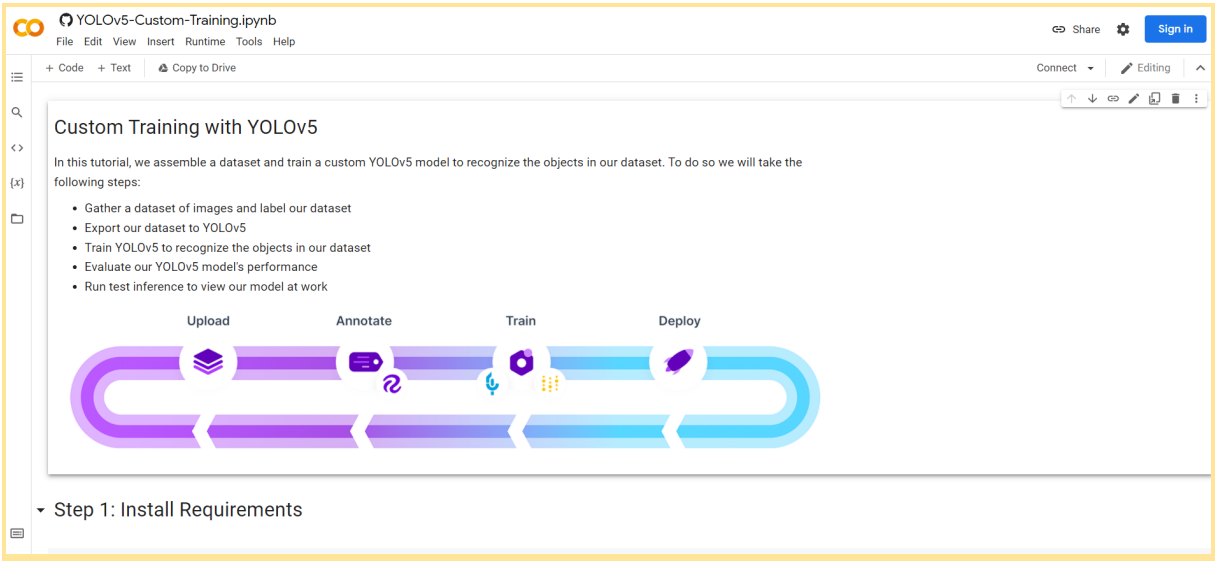
Then, you will see such python source code. Copy and paste it somewhere. We need it very soon.

6.2. RETRAIN OUR MODEL ON GOOGLE COLAB

Google offers a good server for free, so we will use Google Colab for retraining.

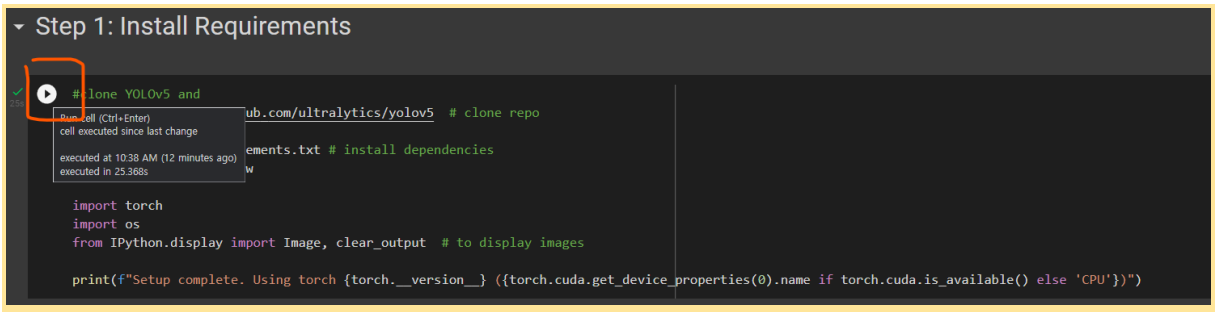
1. Go to the link below.

[YOLOv5-Custom-Training.ipynb - Colaboratory]

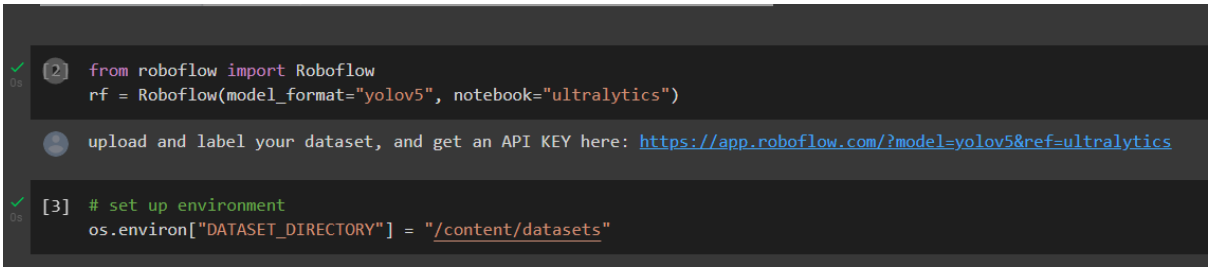


You need to sign in.

2. Run step 1 by clicking the play button.

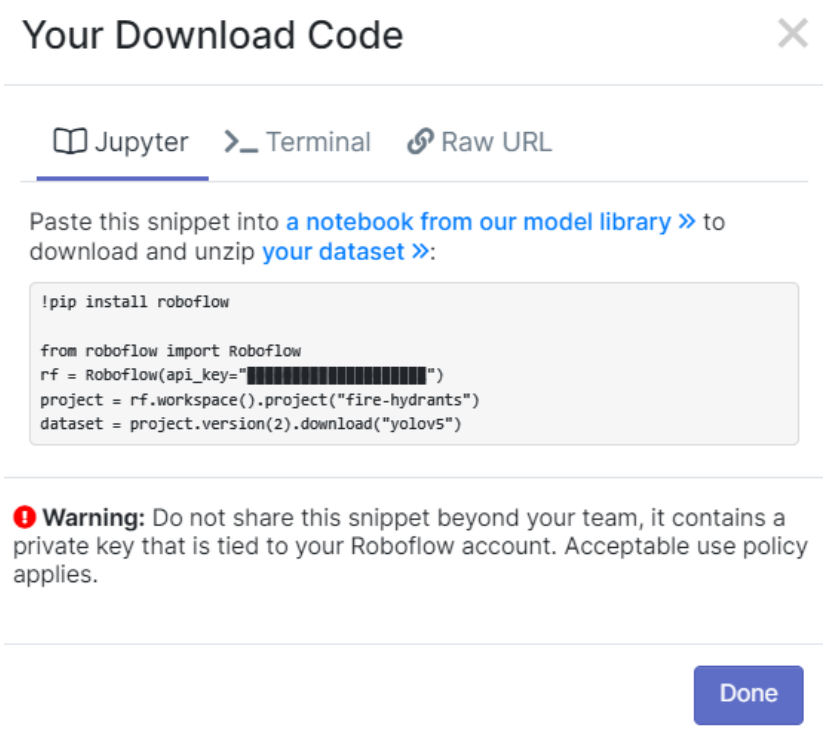


3. Scroll down and run those two codes as well.

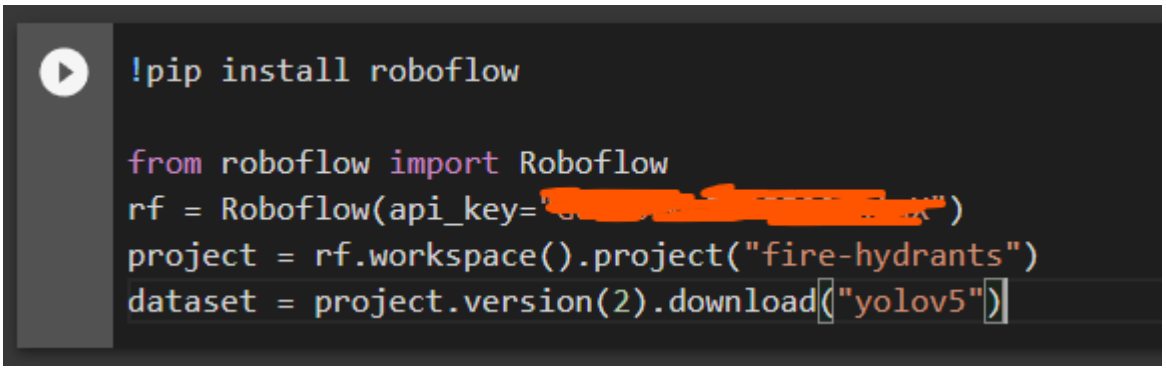


4. The next code will be something like below. This is where you need to paste the python code which you got from Roboflow.

```
[4]  #after following the link above, recieve python code with these fields filled in
     #from roboflow import Roboflow
     #rf = Roboflow(api_key="YOUR API KEY HERE")
     #project = rf.workspace().project("YOUR PROJECT")
     #dataset = project.version("YOUR VERSION").download("yolov5")
```

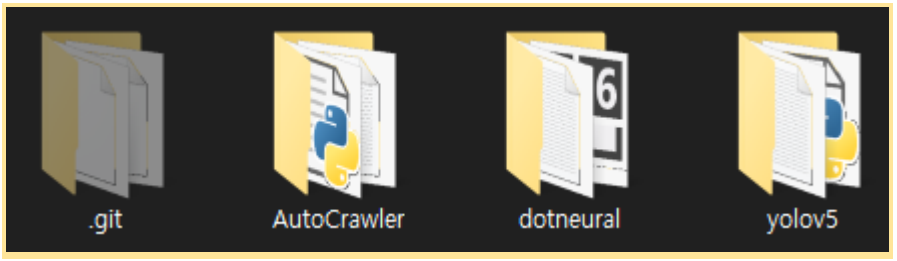


You need to copy and paste this code.

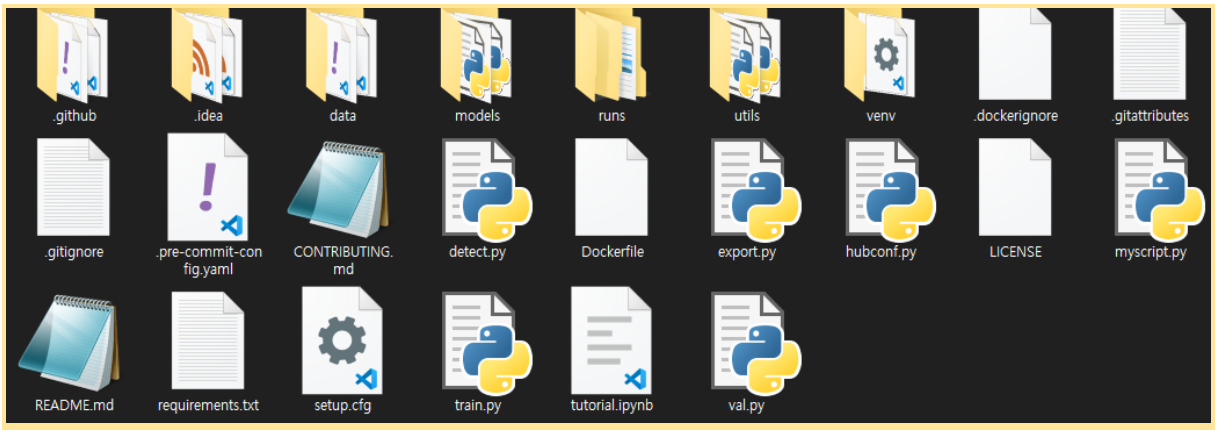


Then run this code.

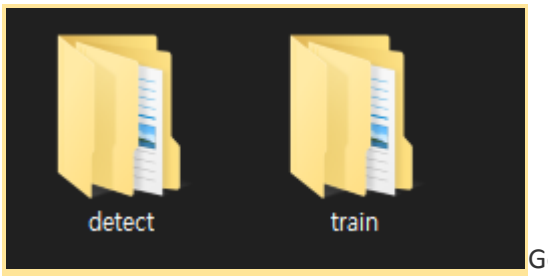
5. Open a folder on your local machine to get our current model data.



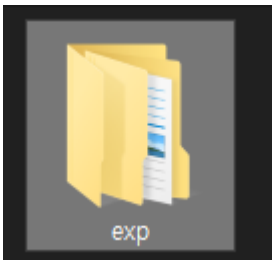
Go to the “yolov5”



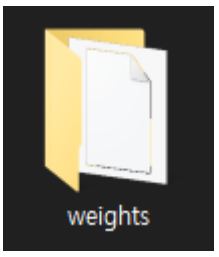
Go to the “runs”



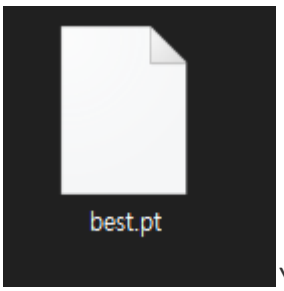
Go to the “train”



→

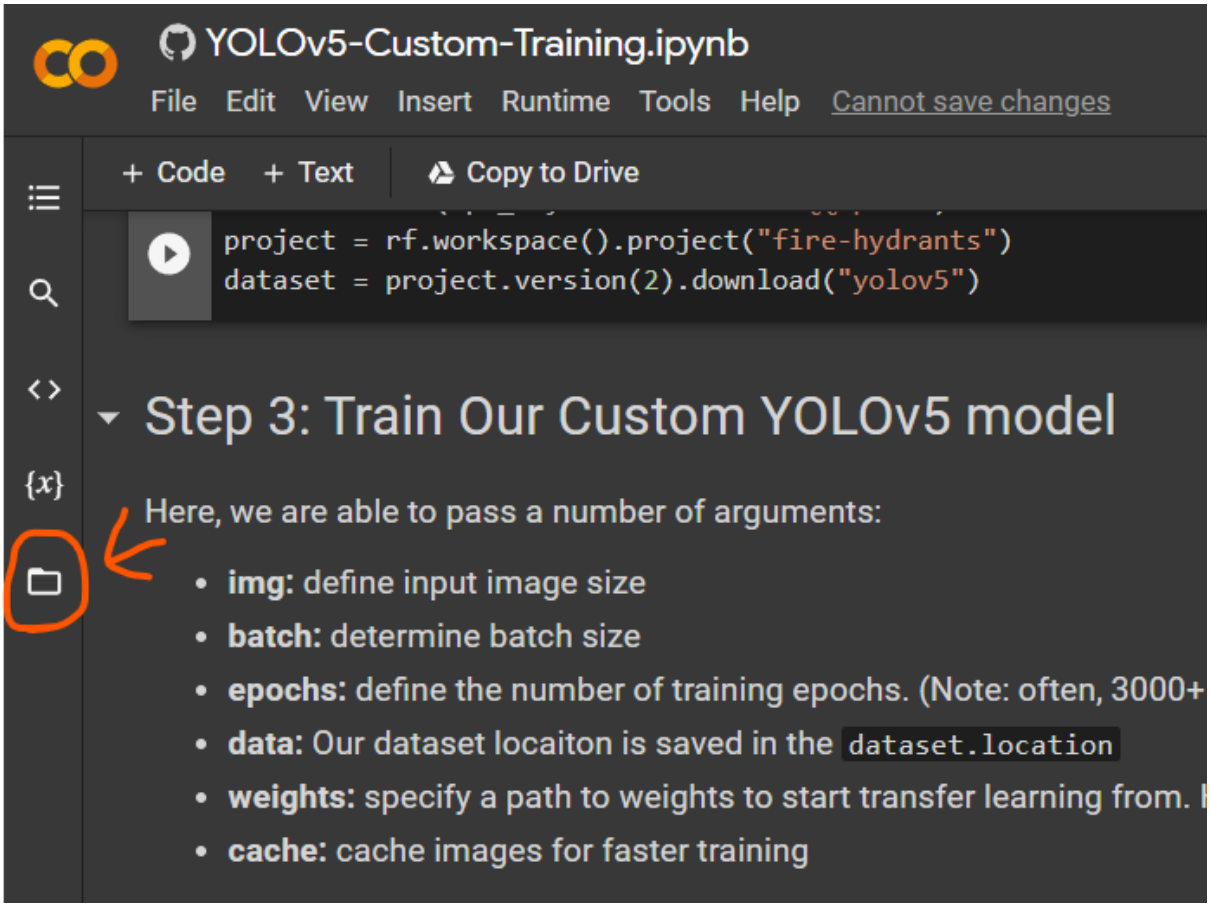


Go to the “exp” -> “weights”

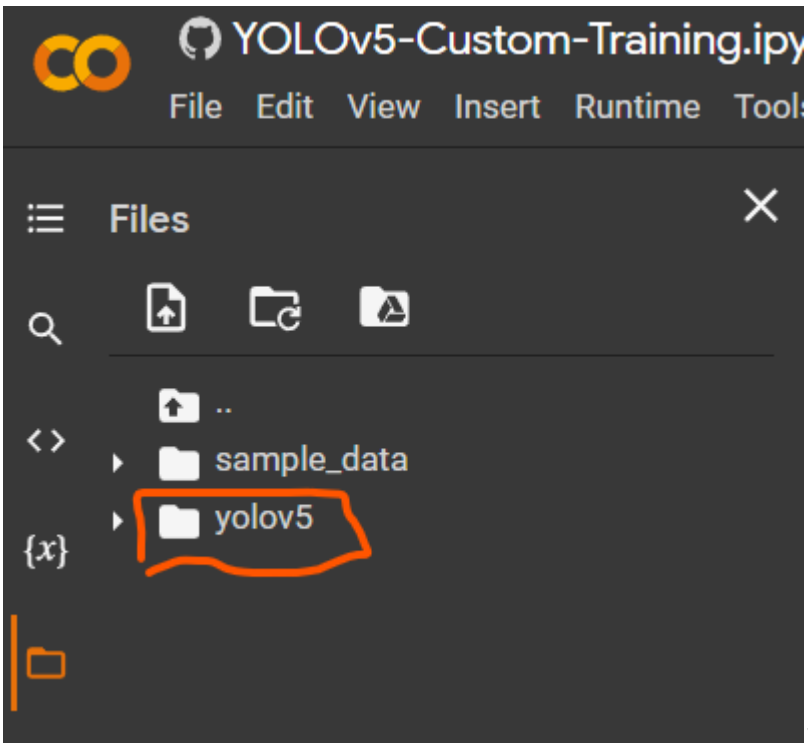


You will see the “best.pt” file. This is everything about our model, simply like a memory. We need to update this memory by retraining.

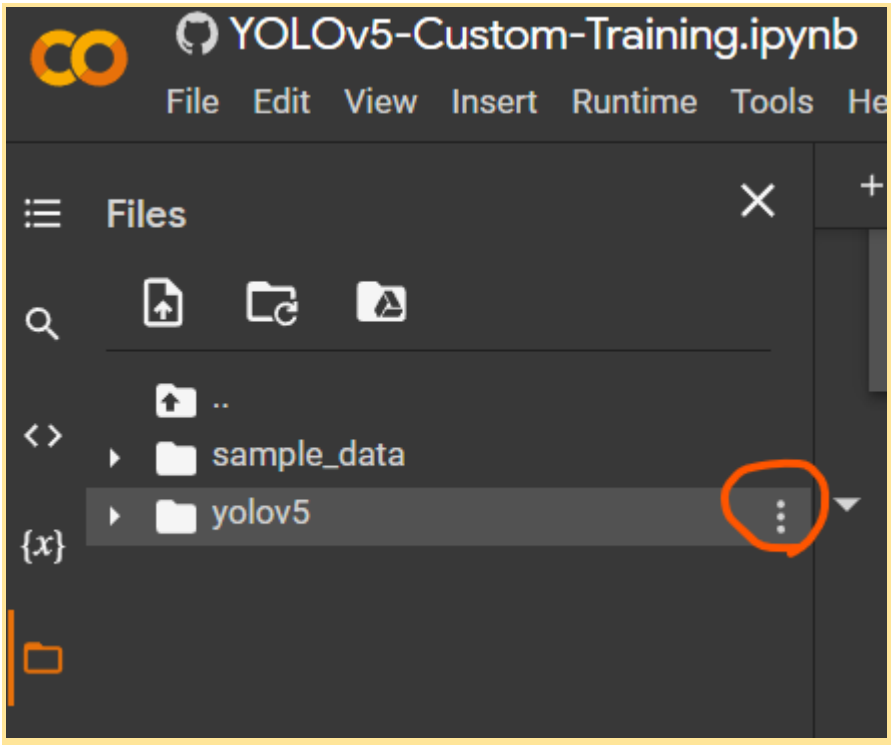
Let’s go back to Google Colab.



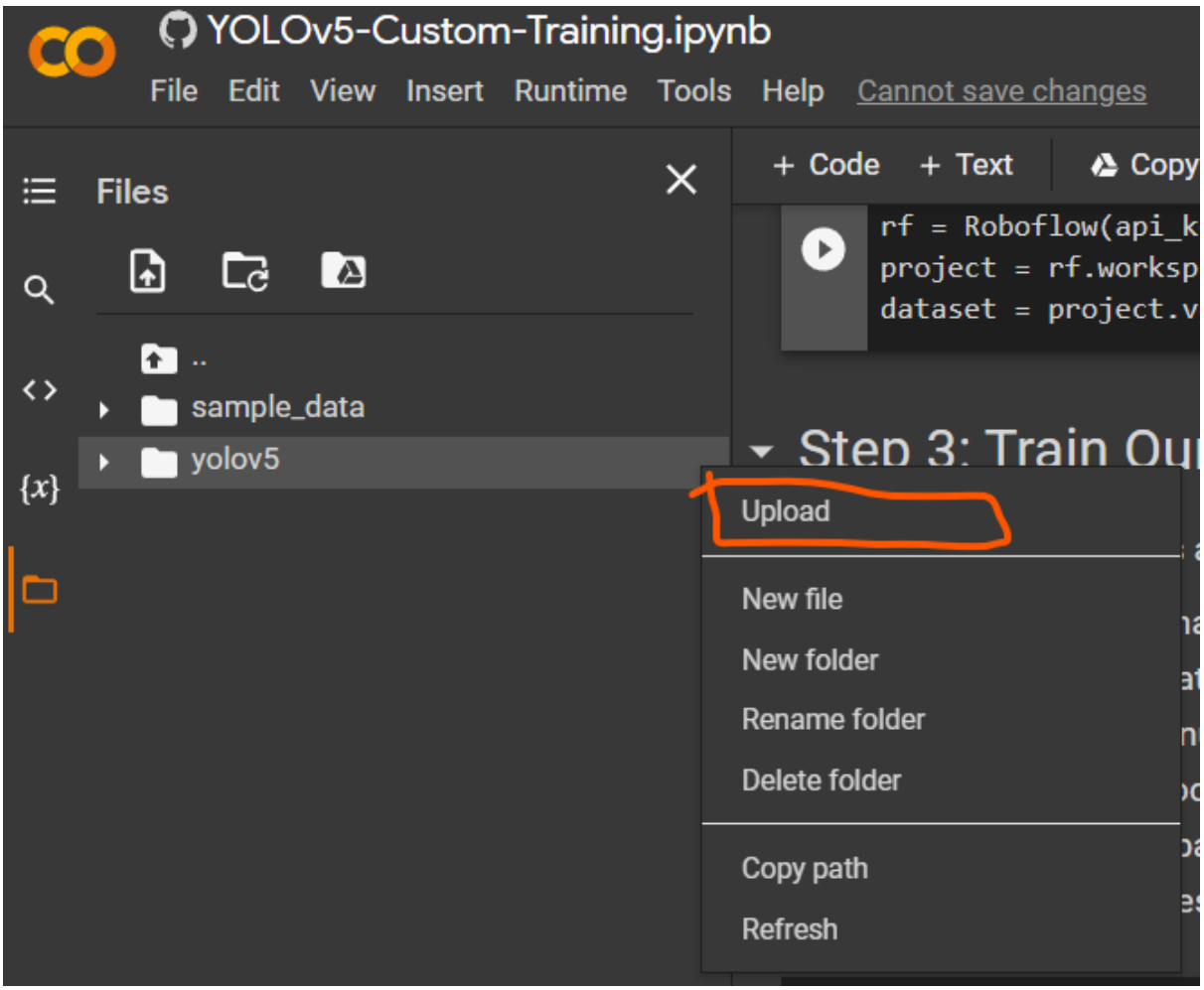
Click the folder icon



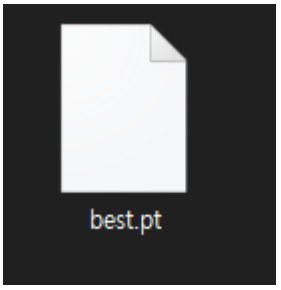
you will see yolov5 folder



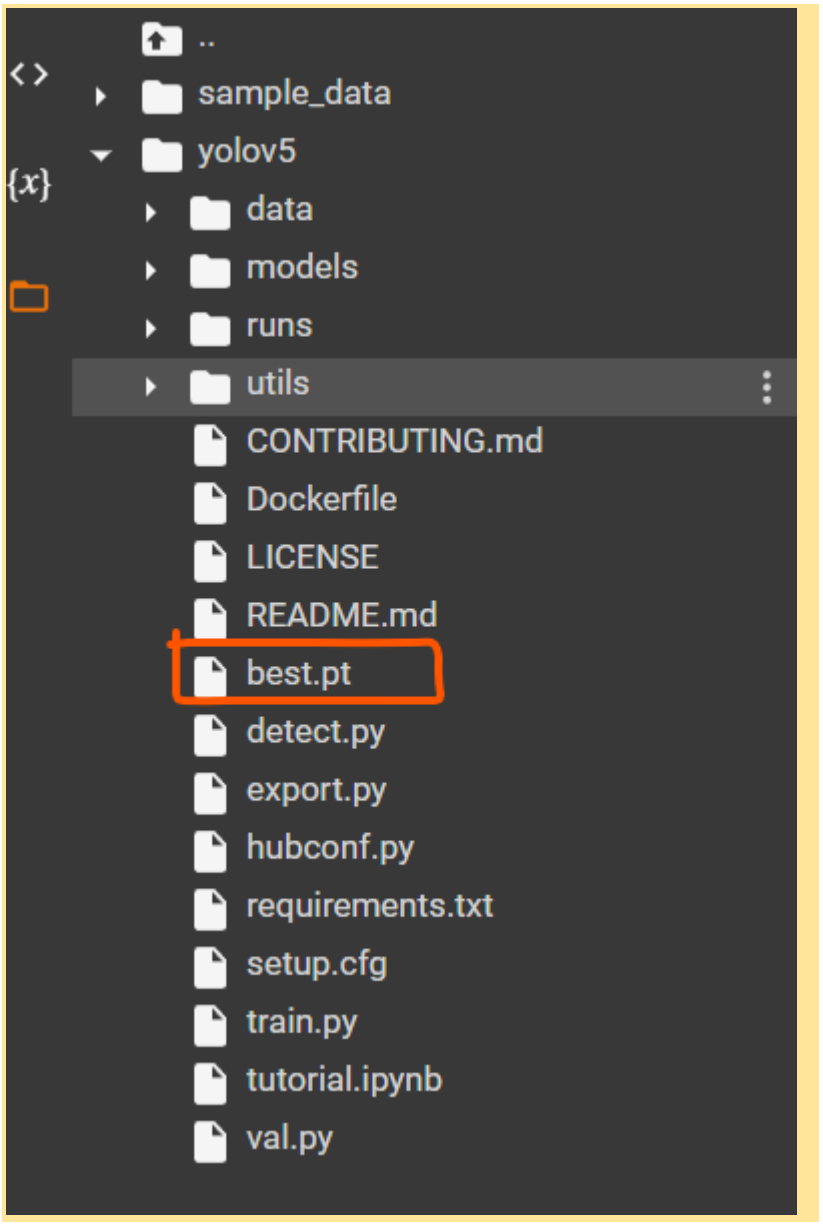
Click the dropdown menu button, orange circle.



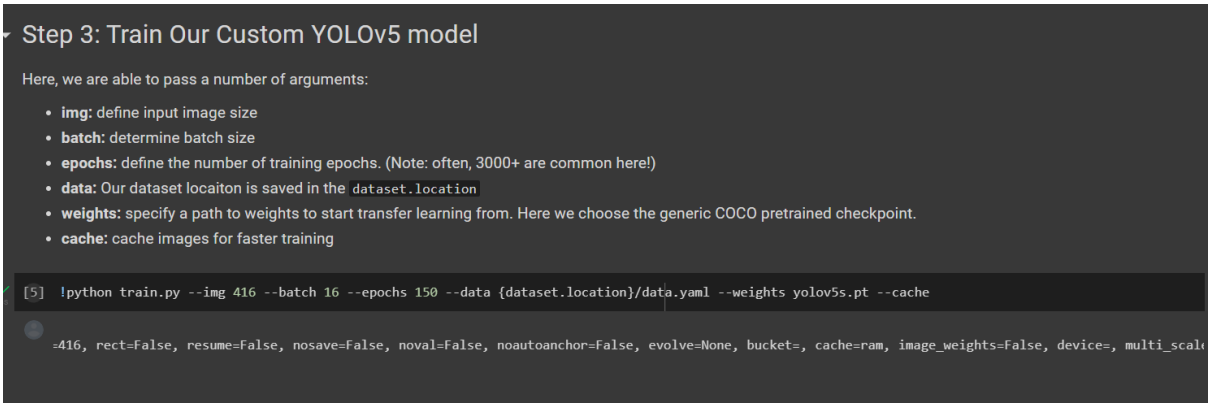
Click the Upload.



Upload this “best.pt” file which we found before.



Once you upload it, our "best.pt" will be inside "yolov5" folder on google colab.



Now, we get back to Step 3, on Colab codes. There is one line code, which is by default

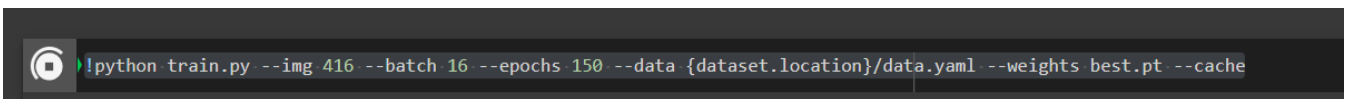
```
!python train.py --img 416 --batch 16 --epochs 150 --data {dataset.location}/data.yaml --weights yolov5s.pt --cache
```

There are a couple of things you might need to know.

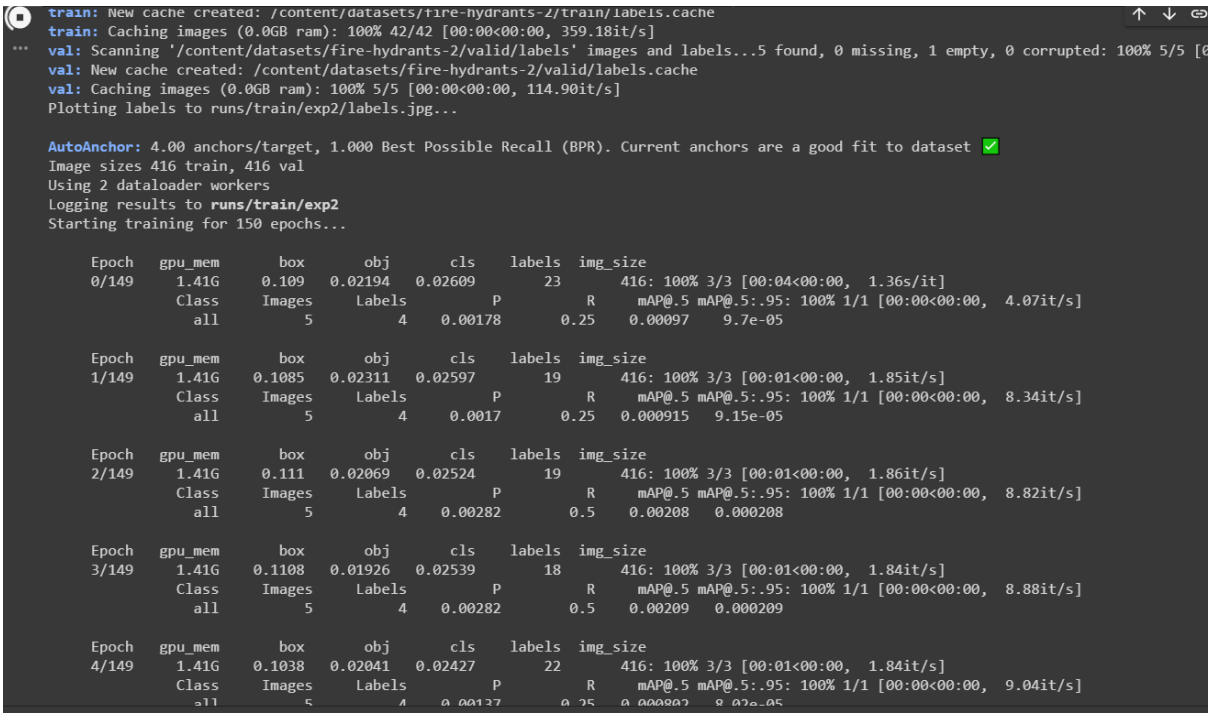
- "epochs" parameter is the number of passes of the entire training dataset. In this case epoch is 150, which means our model will see the "fire hydrant" dataset 150 times during training. You can reduce this number depending on your situation. The higher the epoch is, the more precise our model can detect that object on street images. You should test with epoch 1 or 2 because it will be done very quickly if the number is low.
- "weights" parameter is a starting point where our model starts to retrain. If we did not set the "weights" parameter, our model will forget previous data. So, we set this parameter with our previous data, which is the "best.pt" we looked for before. Briefly, if you set this parameter with a previous data, it will add new data onto the previous one. If you do not, it will forget all previous data and get trained with only new object data.

We need to change the - -weights parameter with our "best.pt" file.

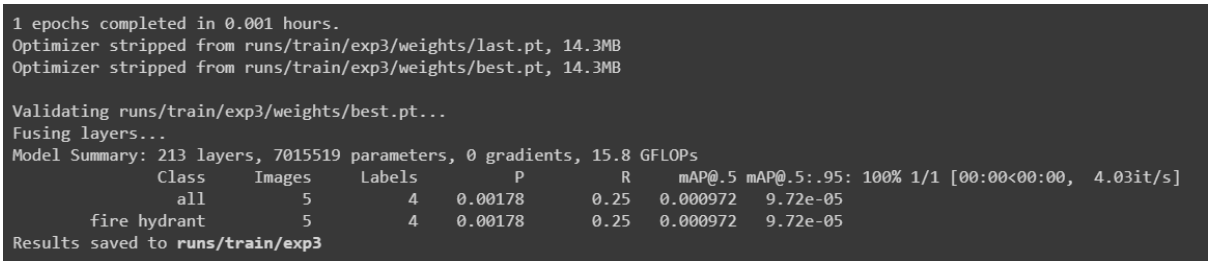
```
!python train.py --img 416 --batch 16 --epochs 150 --data {dataset.location}/data.yaml --weights best.pt --cache
```



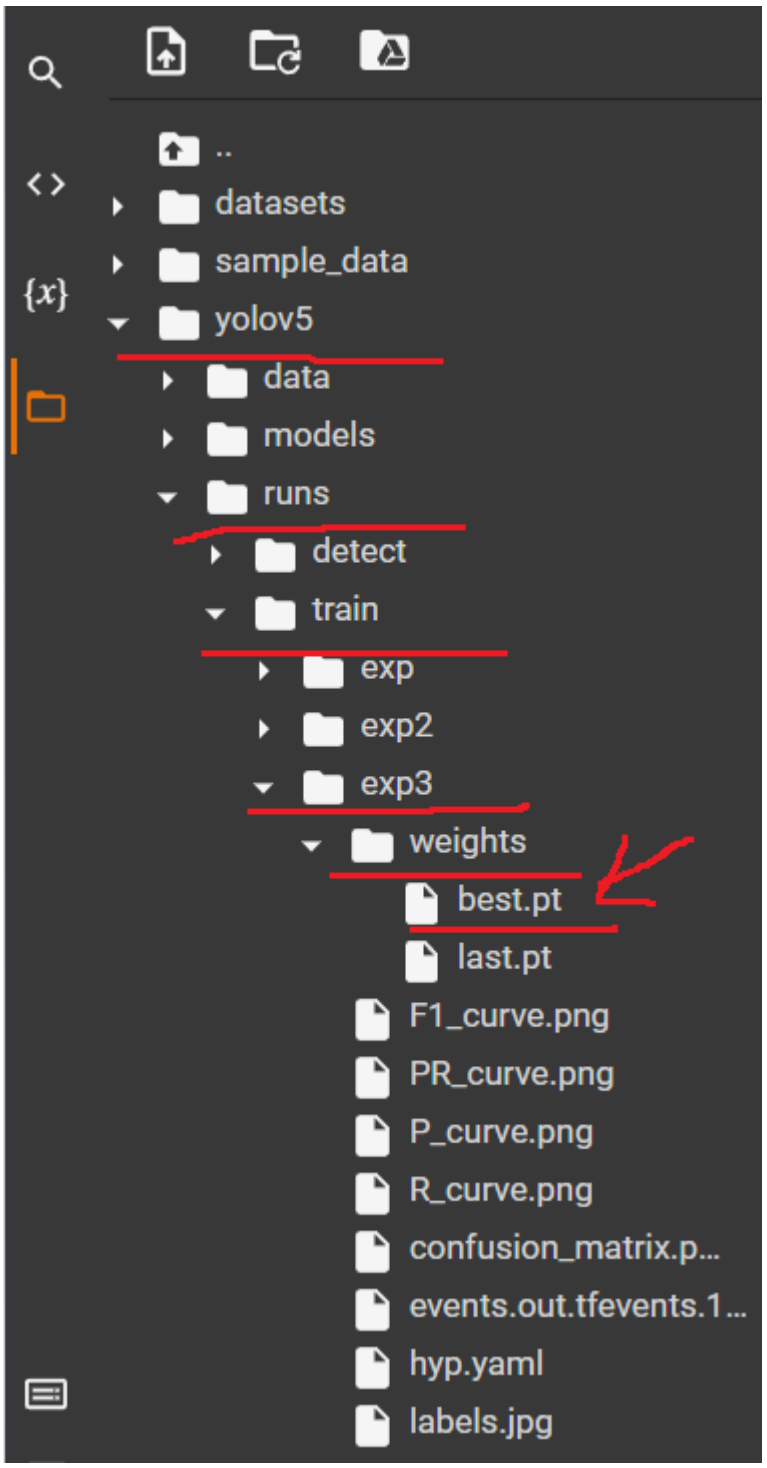
Then run this code.



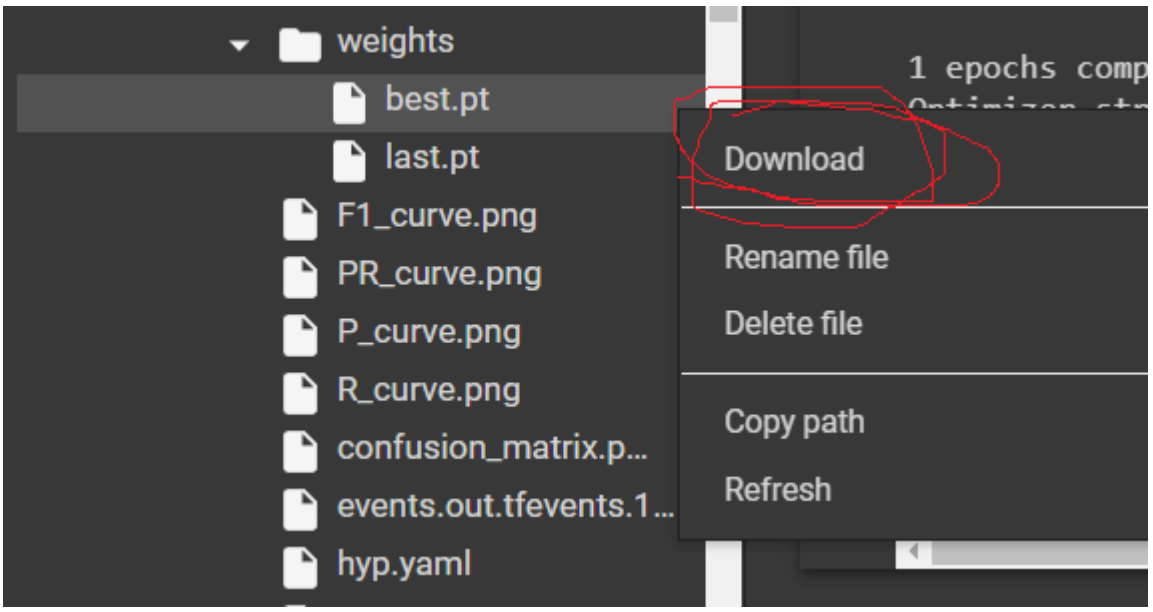
Training started like this.



After training is done, you will see “Results saved to runs/train/exp3 (this will be different on yours)”.

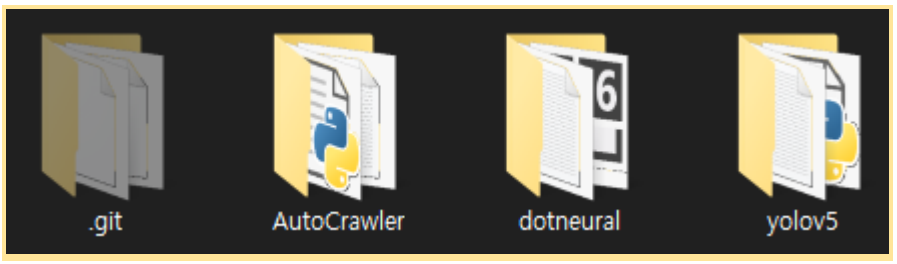


Go to that location, and you will see a “weight” folder and inside there will be an updated new best.pt

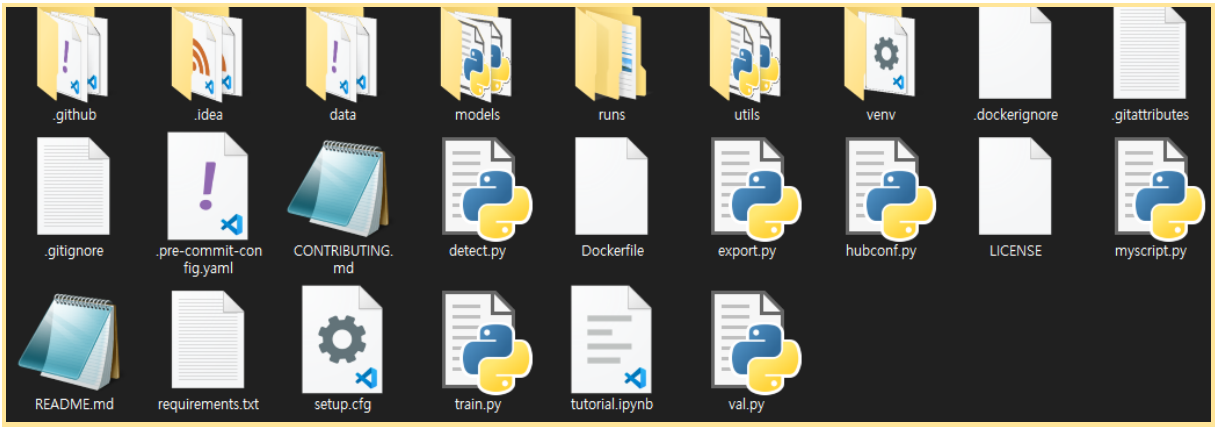


Download the “best.pt”.

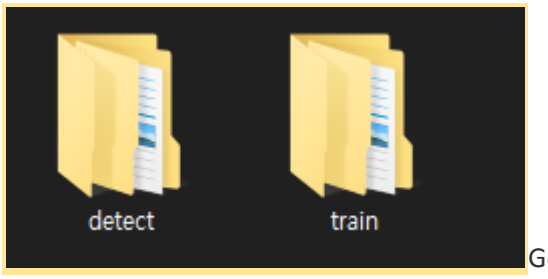
Then, replace a new “best.pt” with your current best.pt on your local machine.



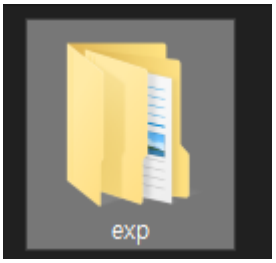
Go to the “yolov5”

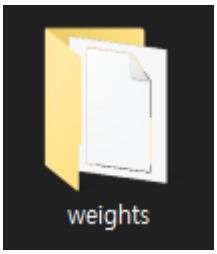


Go to the “runs”

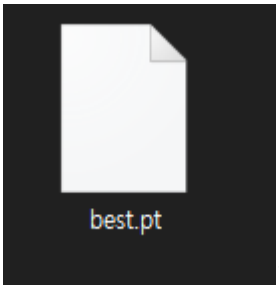


Go to the “train”



→


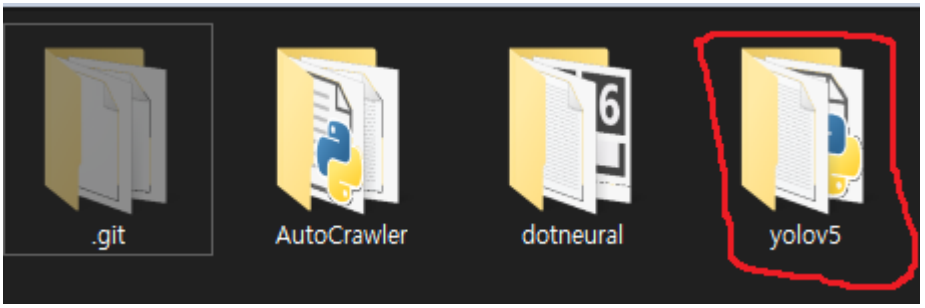
Go to the “exp” -> “weights”

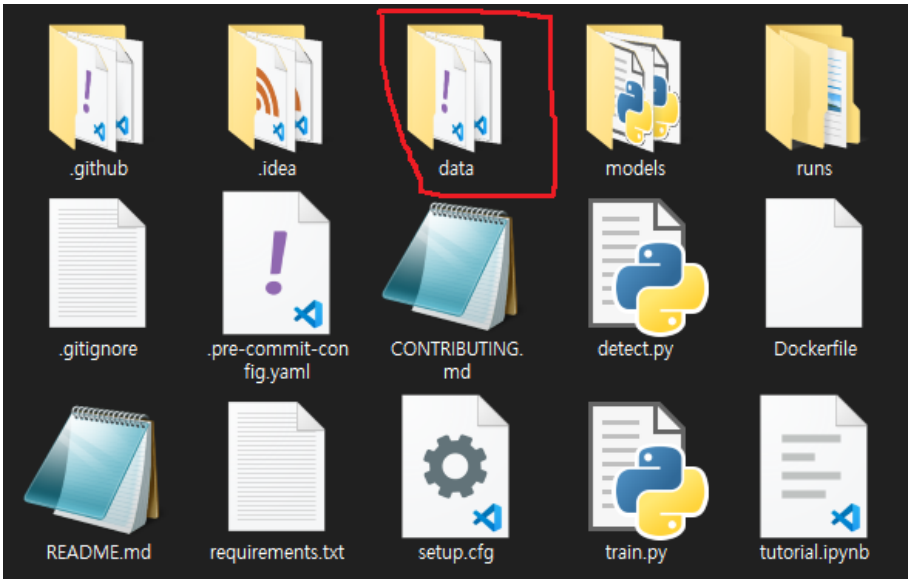


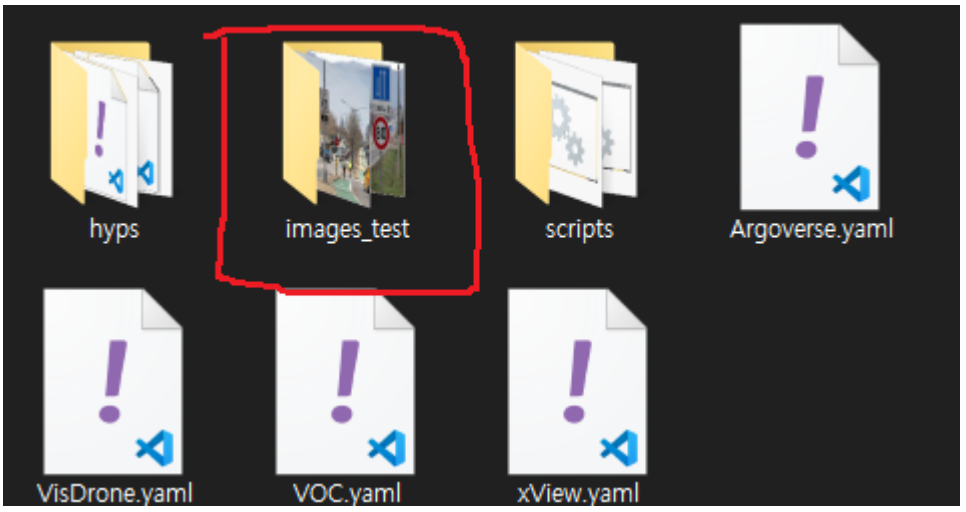
This “best.pt” is now an old one. It must be replaced with the new one, which you download from Google Colab right before step.

Now, “weights” are updated, which means our model has more knowledge than before.

 # Street images should be inside \DOTNN\yolov5\data\images_test

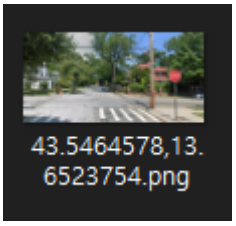






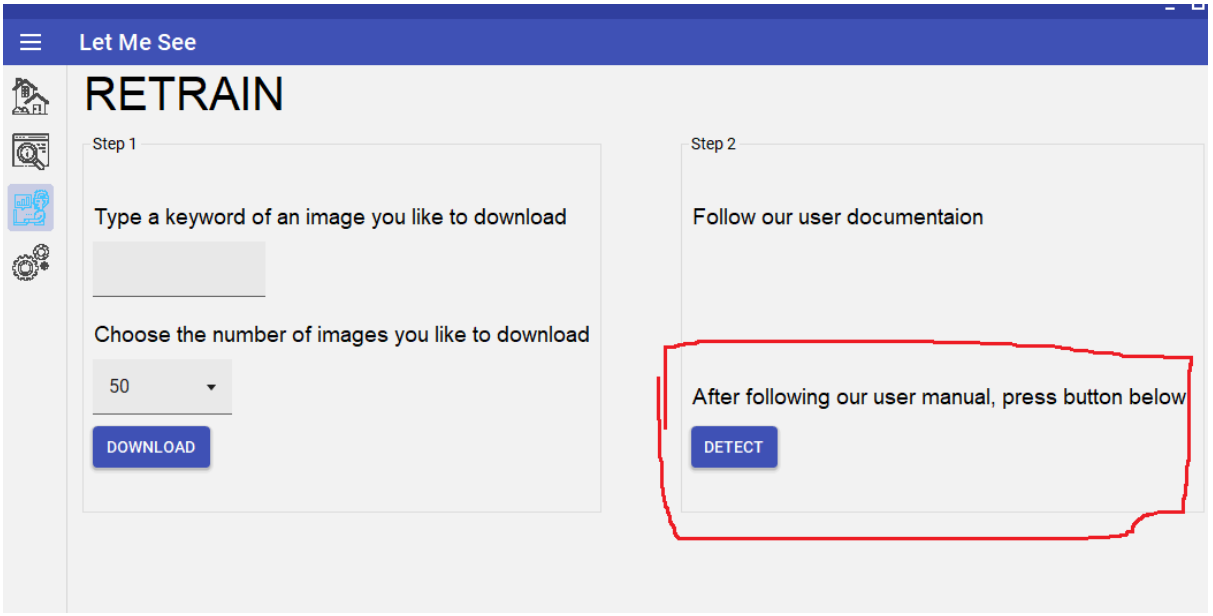
\DOTNN\yolov5\data\images_test

 # A name of a street image should be formatted like [longitude, latitude].png
EX] -24.43256,62.235235.png



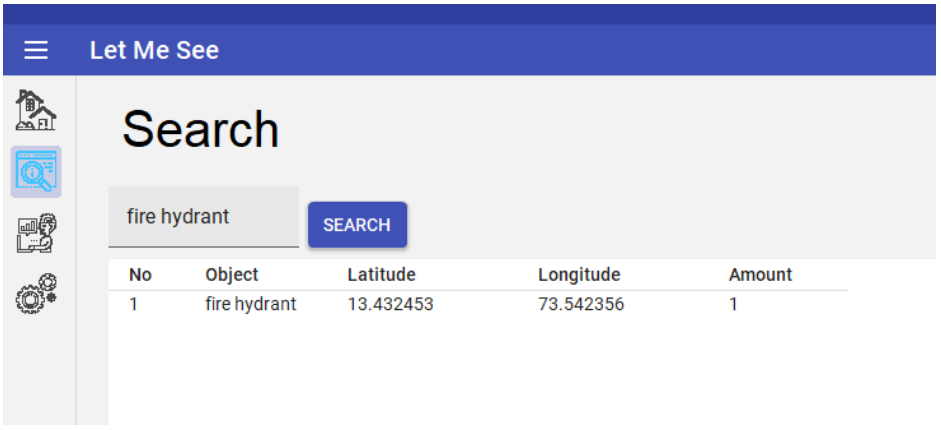
Now, we are ready to detect new objects on street images.

Let's click the "detect" button to find objects you wanted to look for in the beginning.

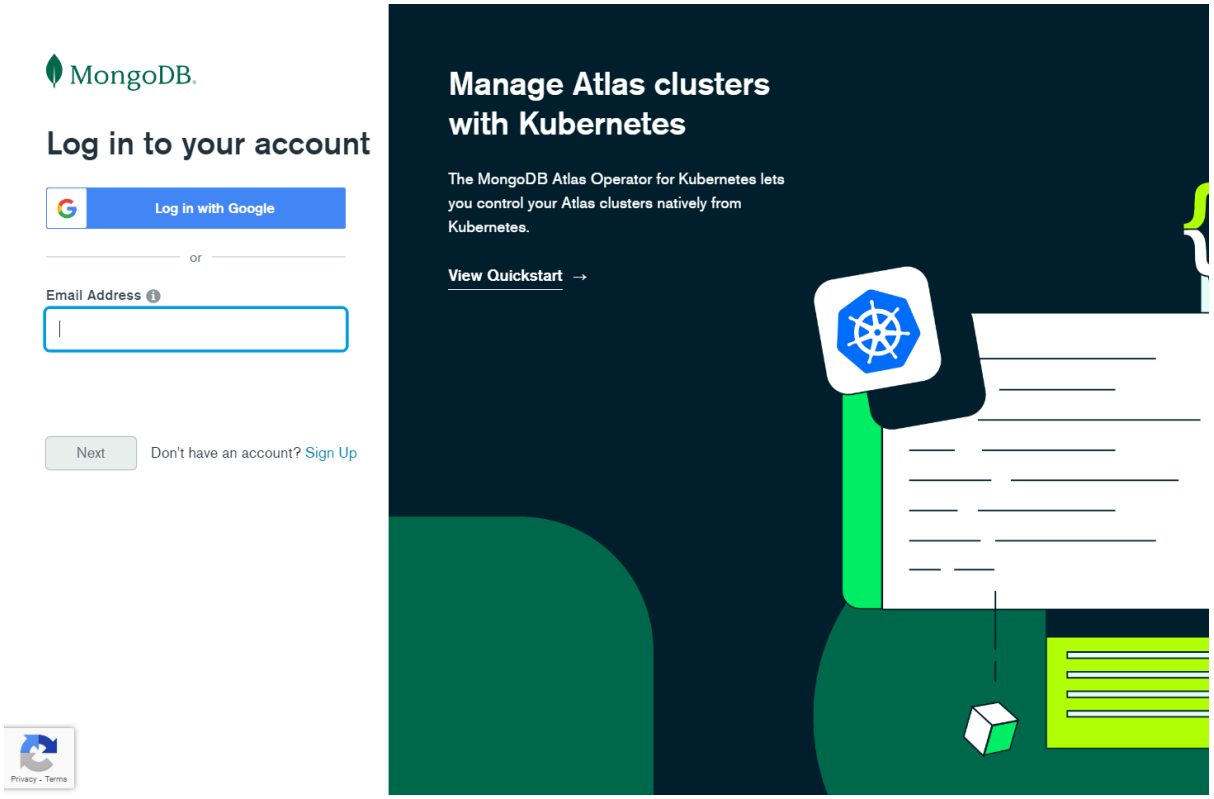


you will see a prompt come up and as soon as our model finishes it will disappear.

Let's search "fire hydrant" again, if our model detected it, this time we will see its information.



7. HOW TO CHANGE DATABASE SERVER



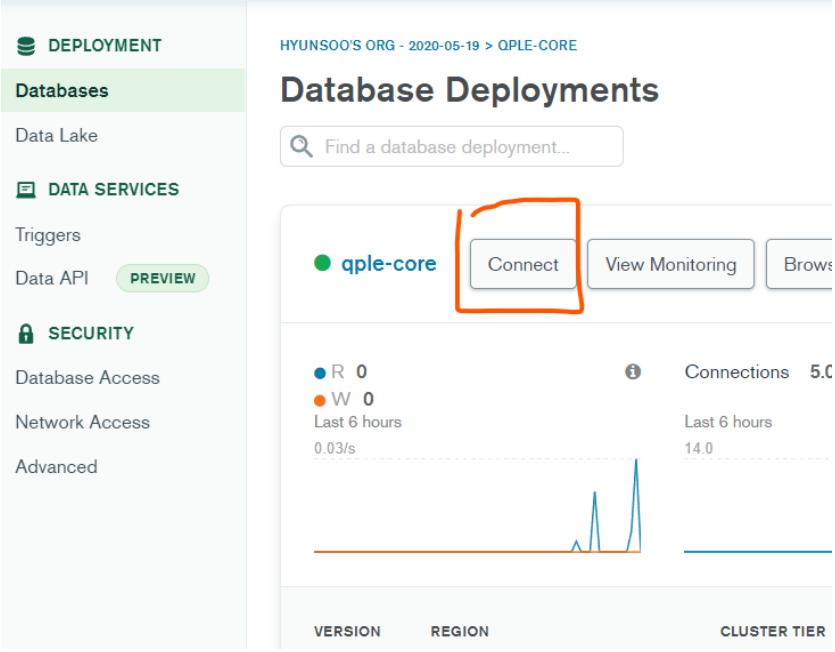
1. **Go to the MongoDB website and log in.**

https://account.mongodb.com/account/login?signedOut=true

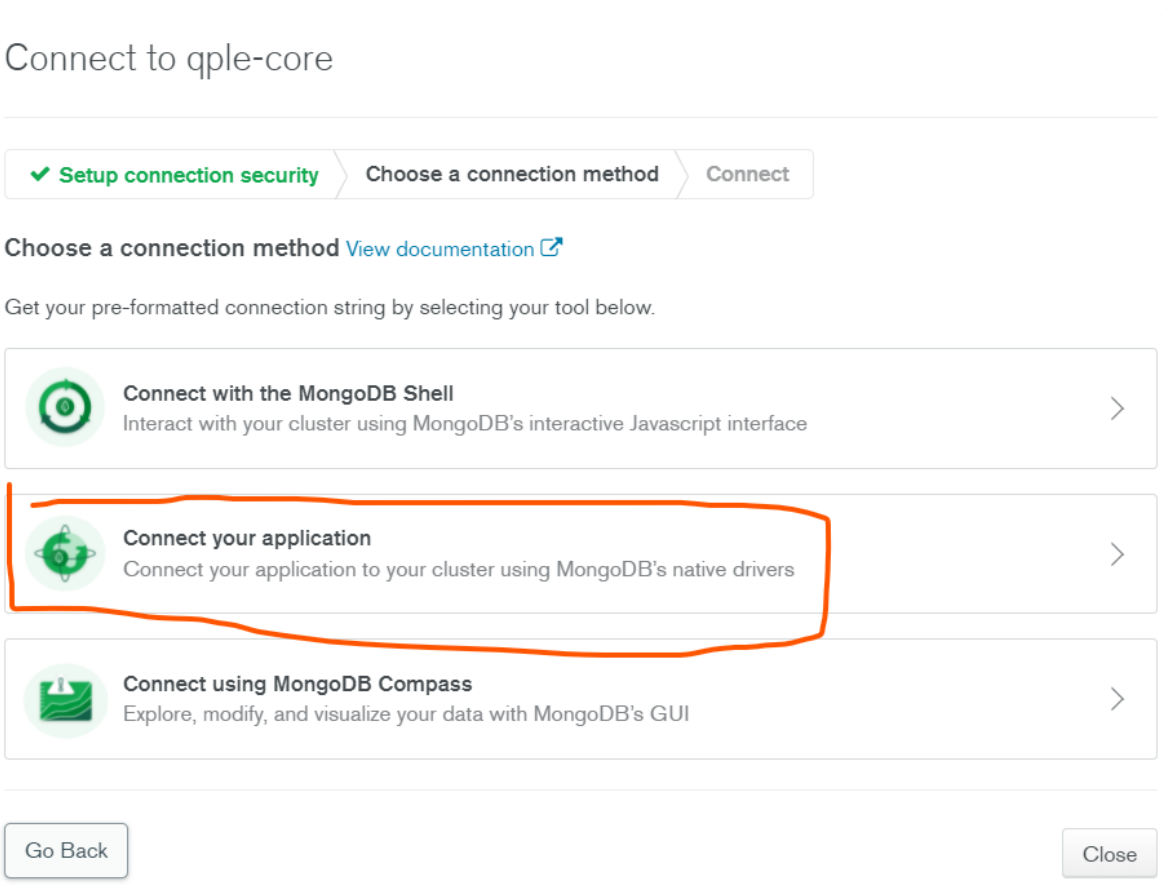
2. **Create a free-tier server.** **Getting Your Free MongoDB Atlas Cluster**

https://www.youtube.com/watch?v=rPqRyYJmx2g

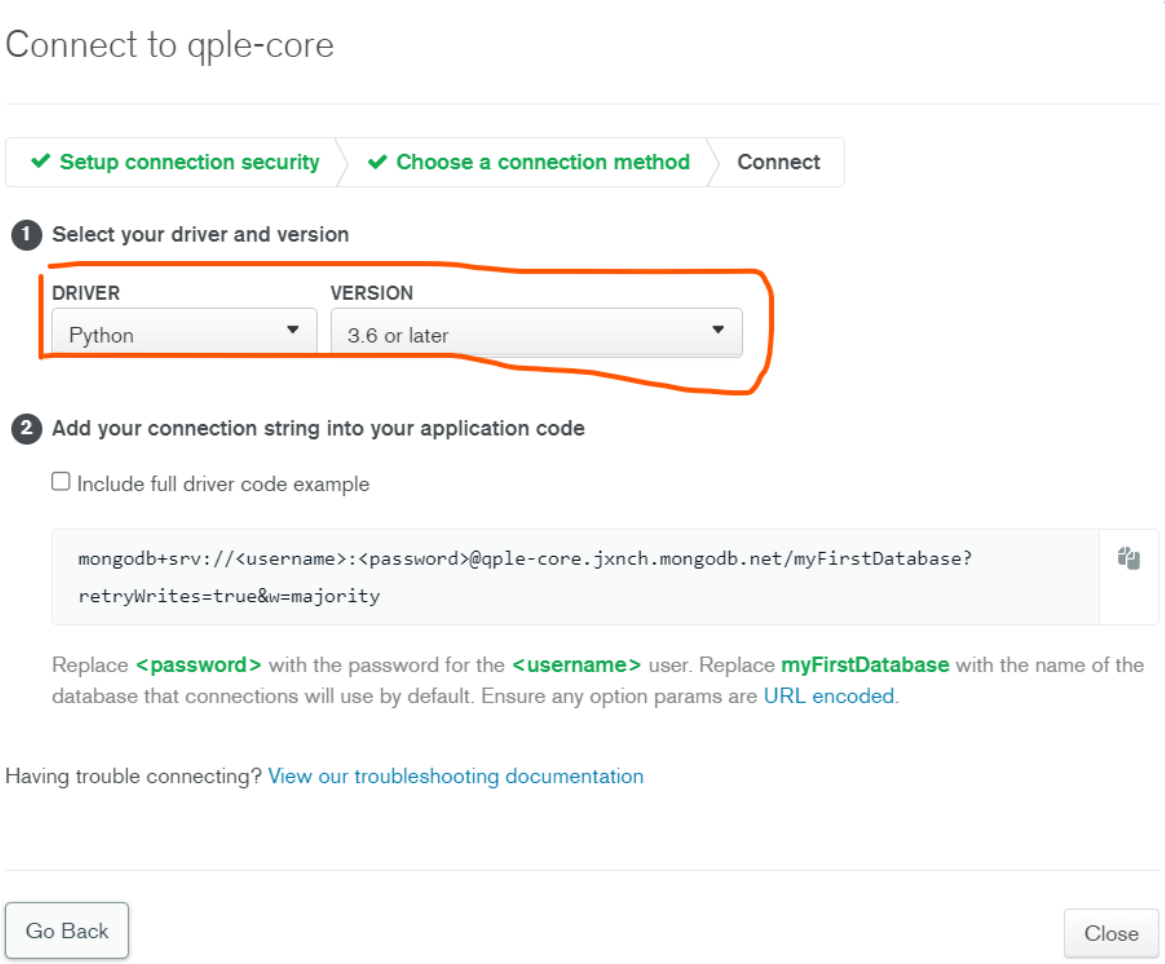
3. **Click “Connect”.**



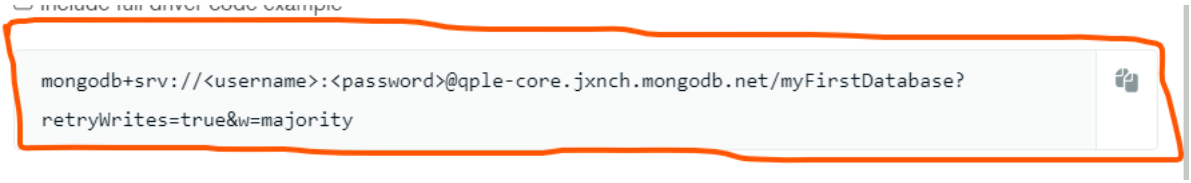
4. **Click “Connect you application”**



5. **Change “Driver” to “Python”, and “Version” to “3.6 or Later”.**



6. **Change <username> and <password> to your username and password**

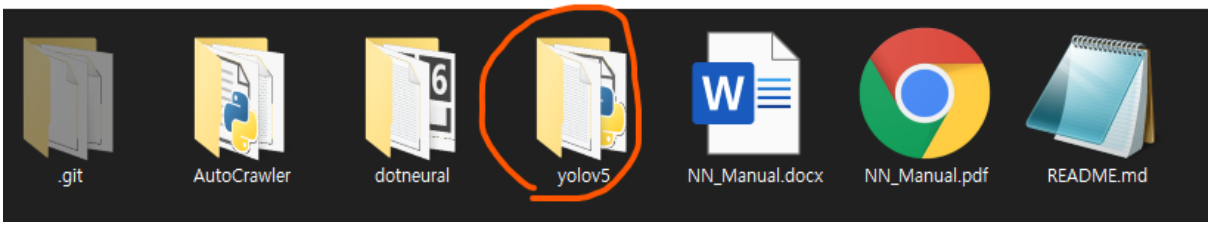


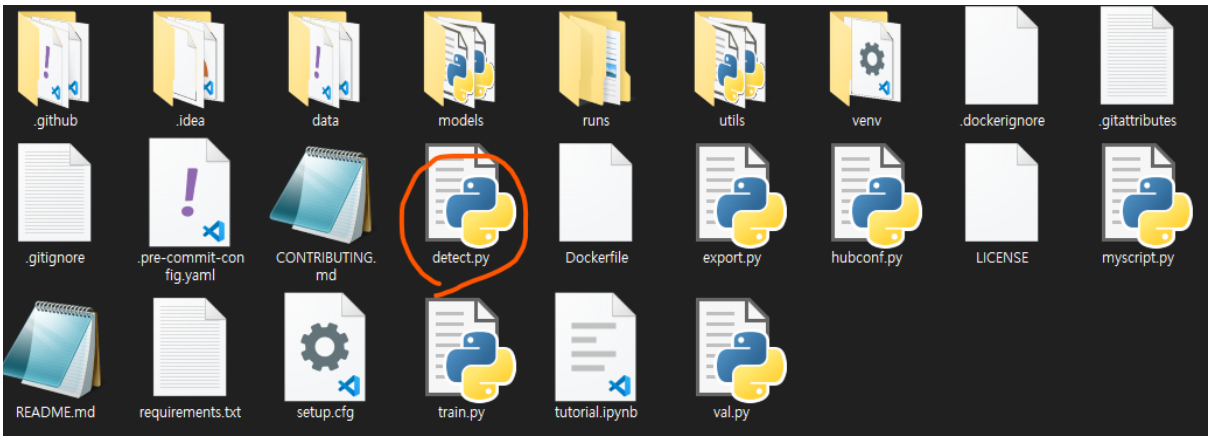
Let's say your username is **jack77** and password is **abcd77**.

your connection string should be below.

```
mongodb+srv://jack77:abcd77@YOUR-SERVER.jxnch.mongodb.net/myFirstDatabase?retryWrites=true&w=majority
```

7. **Go to the yolov5 directory and open detect.py file.**





8. **Go down to line 43.**

```
# MongoDB Connection
from pymongo import MongoClient
from collections import defaultdict
client=
MongoClient('mongodb+srv://developer1:hyunsoo97@qple-co
re.jxnch.mongodb.net/myFirstDatabase?retryWrites=true&w
=majority')
db = client['Lemmefind']
obj = db['Objects']
# END
```

9. **Change line 43 with connection string in step 7.6**

```
# MongoDB Connection
from pymongo import MongoClient
from collections import defaultdict
client=
MongoClient('mongodb+srv://jack77:abcd77@YOUR-SERVER.jxnch.m
ongodb.net/myFirstDatabase?retryWrites=true&w=majority')
db = client['Lemmefind']
obj = db['Objects']
# END
```

8. CONCLUSION

Thank you for following this long document with me.