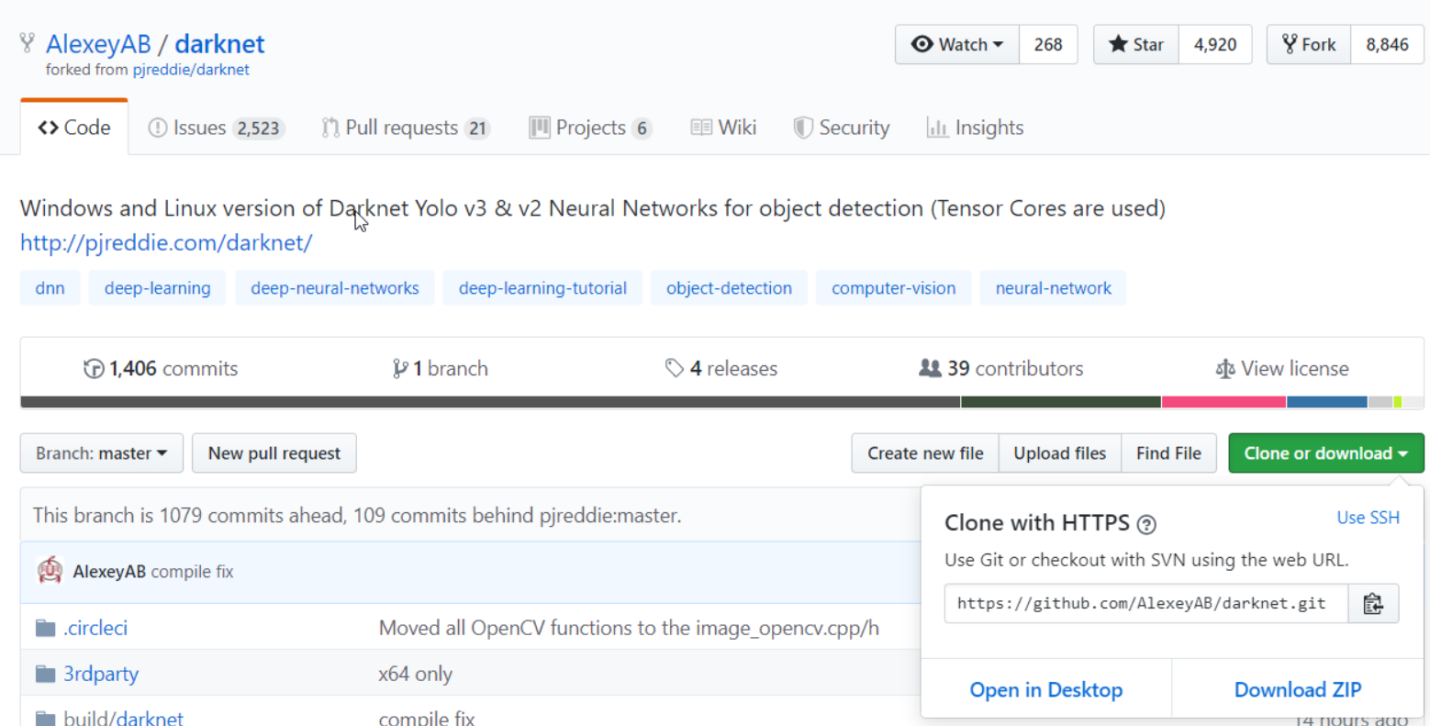
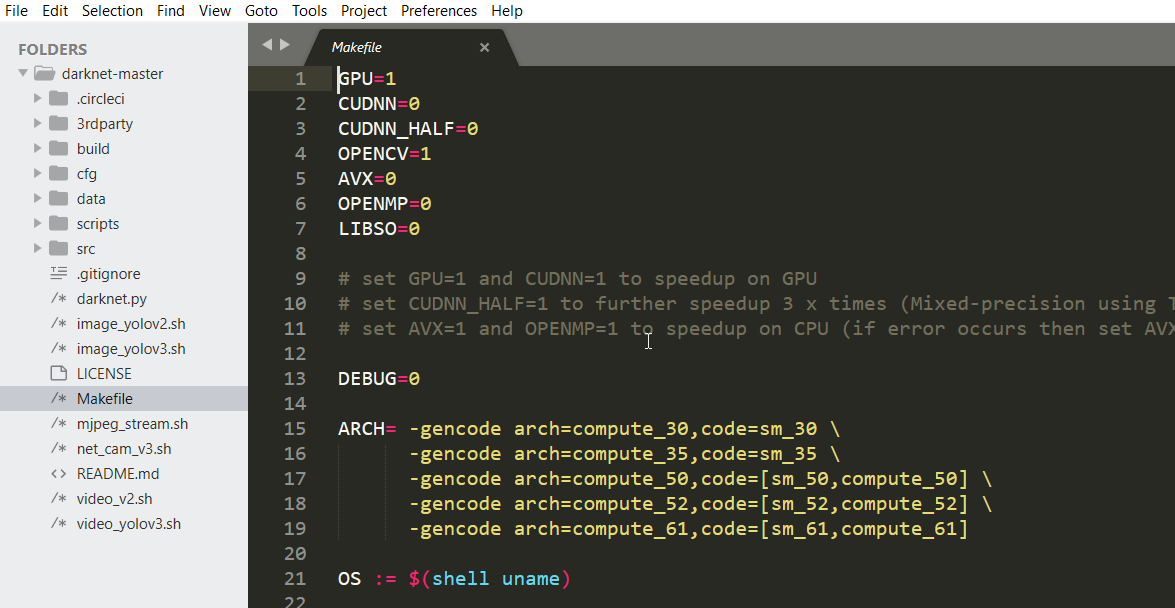
**Yolo Darknet Installation**

**Step 1**.

Git clone  <https://github.com/AlexeyAB/darknet>  or download zip then unzip it



**Step 2**.

Open the Makefile in the unzipped folder **darknet-master**

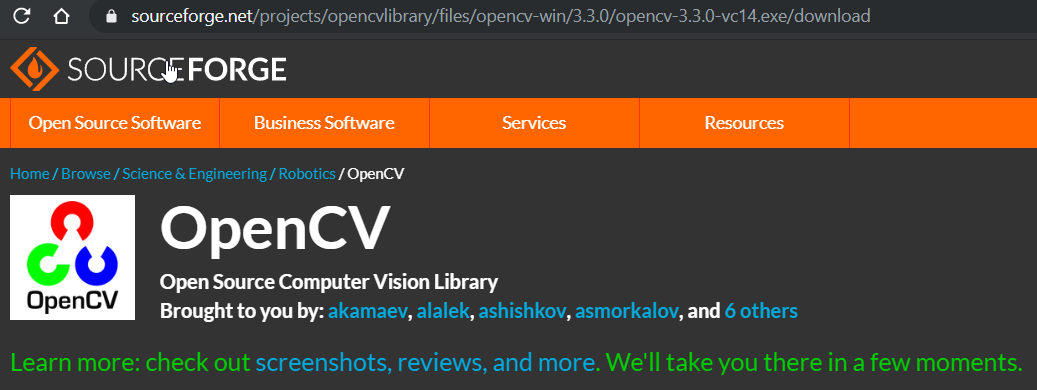
In the Makefile change GPU = 1 and OPENCV = 1

**Step3.**

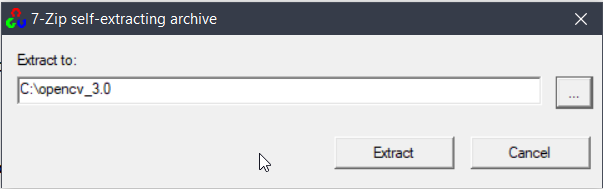
Save the Makefile.

**Step 4**:-

Go to <https://sourceforge.net/projects/opencvlibrary/files/opencv-win/3.3.0/opencv-3.3.0-vc14.exe/download> and Download OpenCV



Create a new folder named **opencv\_3.0** in C drive and install and extract in opencv\_3.0 in C drive



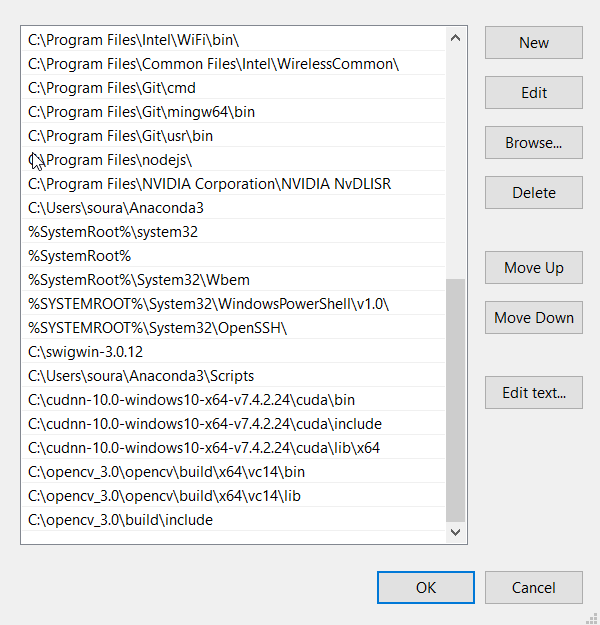
After Successful installation

Add to path in environment variables

C:\opencv\_3.0\opencv\build\include

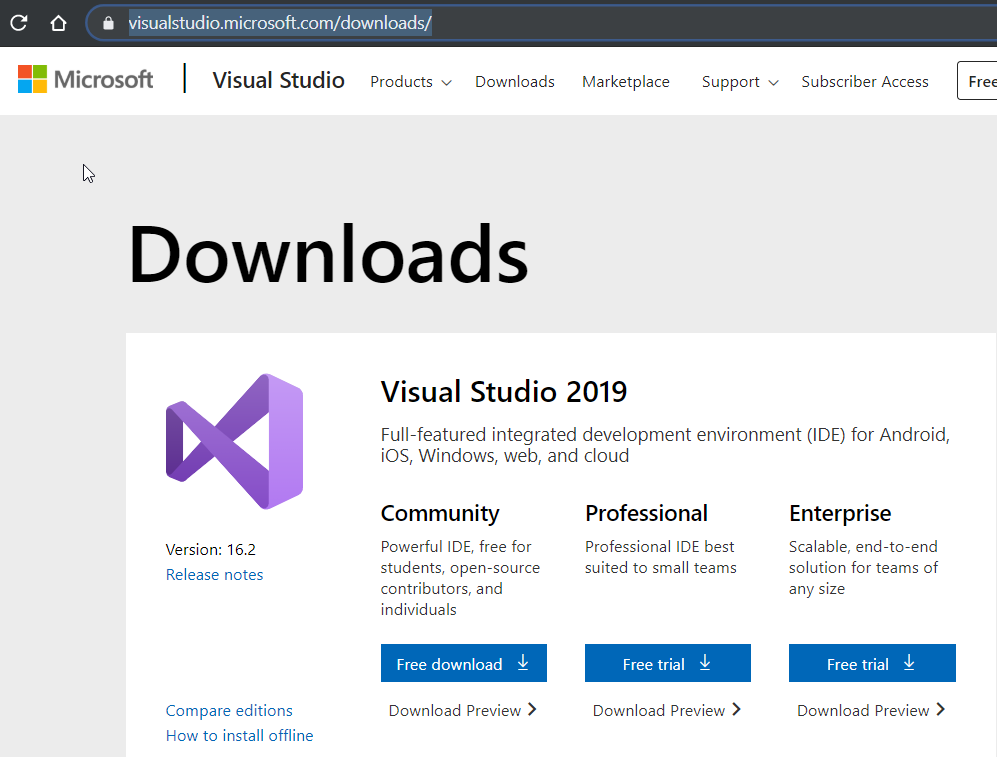
C:\opencv\_3.0\opencv\build\x64\vc14\lib

C:\opencv\_3.0\opencv\build\x64\vc14\bin



**Step 5:-**

Download and Install Microsoft Visual Studio Community Edition from <https://visualstudio.microsoft.com/downloads/> with the C++ bundle packages

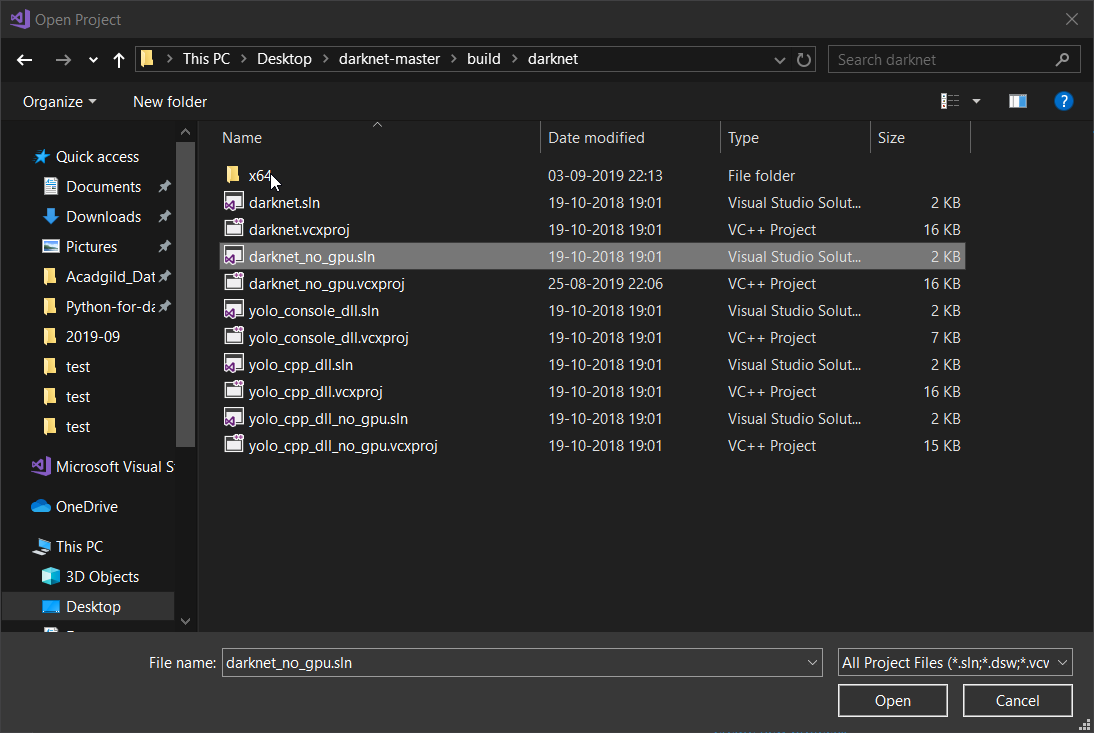


**Step 6 :-**

Open Visual Studio Community Edition 2017

Open New Project

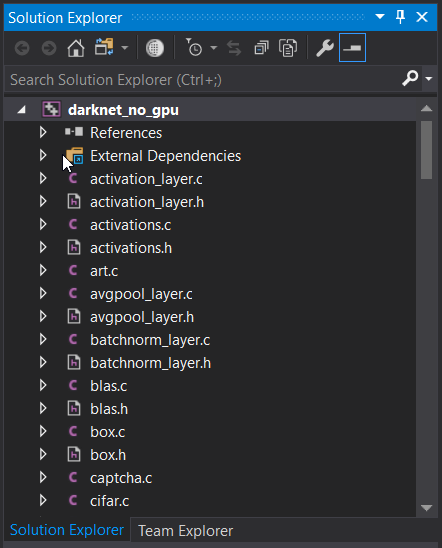
C:\Users\soura\Desktop\darknet-master\build\darknet\_no\_gpu.sln



**Step 7:-**

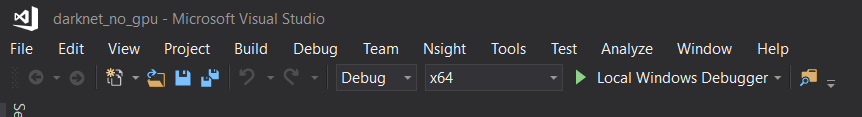
Select OK for retargeted objects if POP UP window appears.

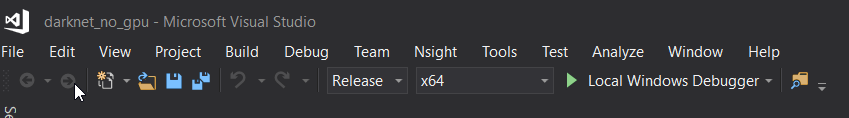
In the Solutions Explorer collapse the menu for darknetdarknet\_no\_gpu.sln



**Step 8:-**

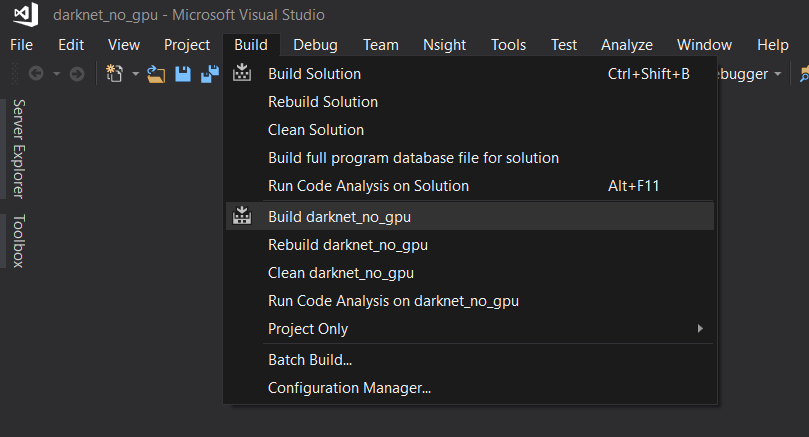
Change the **Debug** option to **Release**





**Step 9 :**

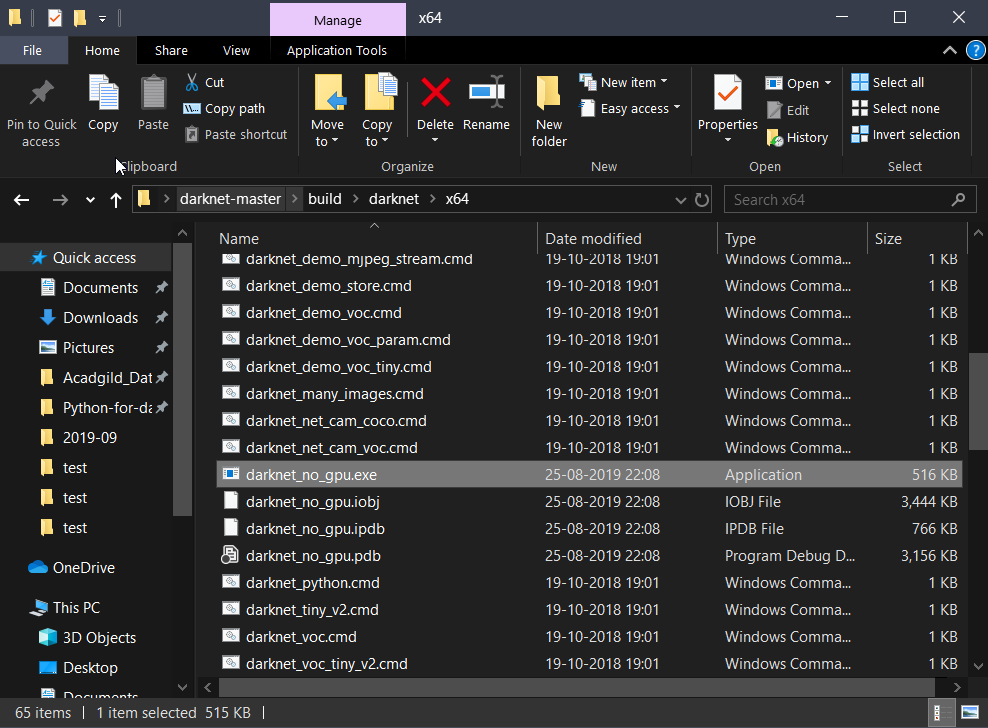
Then go to **Build** Options and select **Build darknet\_no\_gpu**



at end it will show **Build 1 succeeded**.

**Step 11:-**

Now go to C:\Users\soura\Desktop\darknet-master\build\darknet\x64 and you will see darknet\_no\_gpu.exe



**Step 12:-**

Download the yolo weights from <https://pjreddie.com/media/files/yolov3.weights> and paste in x64 folder

**Step 13:-**

Open CMD here or in the x64 folder

**Step 14:-**

For Images

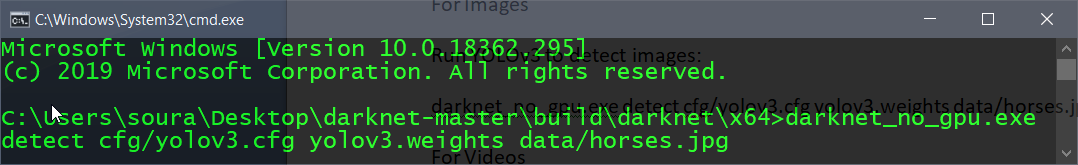
Run YOLOv3 to **detect images**:

darknet\_no\_gpu.exe detect cfg/yolov3.cfg yolov3.weights data/horses.jpg

For Videos

Run YOLOv3 to **process videos**:

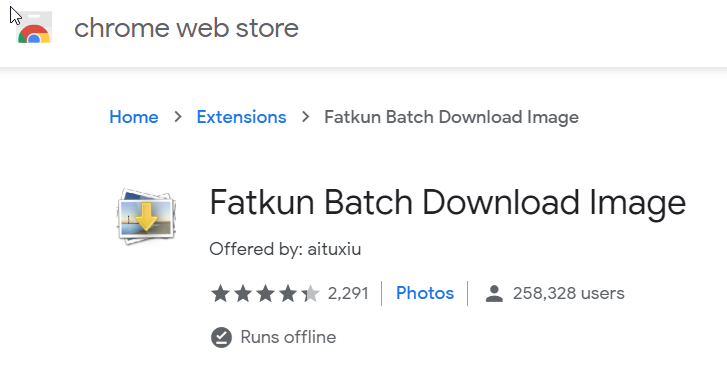
darknet\_no\_gpu.exe detector demo data/coco.data cfg/yolov3.cfg yolov3.weights testvideo.mp4 -out\_filename result.avi



## Collect images and label

**Step 1:**

Collect images using a **scrapper**



**Or use**

<https://github.com/jiwitesh/ImageScrapper/blob/master/ImageScrapper/imagescrapperservice/test2.py>

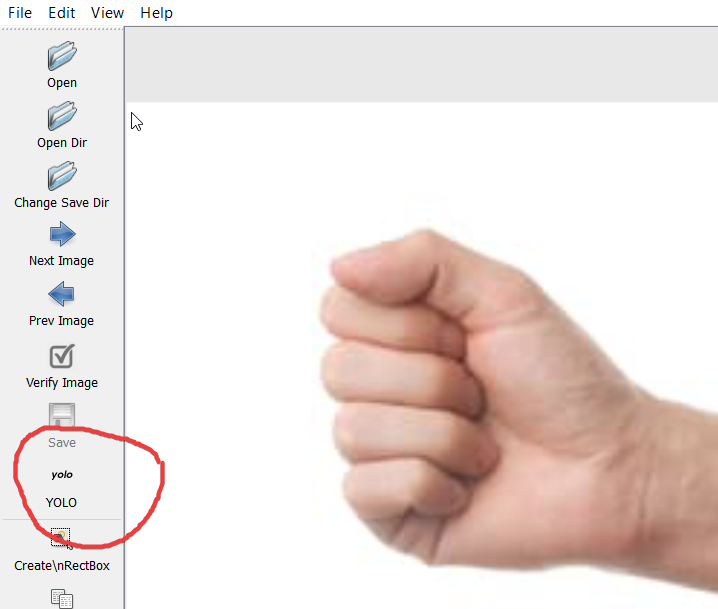
**Step 2:**

Collect all the images in a single folder

**Step 3:**

Do the labelling using LabelImg or your preferred tool.

Open LabelImg and change the format from PASCAL VOC to YOLO.



Open the directory where all images are present and start labelling

All the annotations will be saved in the .txt format with the label id as generated in classes.txt

**Step 3:**

Create a new folder named labels

**Step 4:**

Inside that create another new folder called images and keep all scrapped images there and

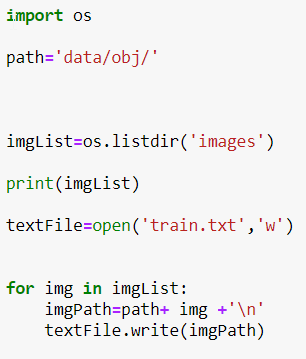
all corresponding .txt in a separate folder called txt

**Step 5:**

use the below script and save it in a new file called train.py

run the train.py file and

train.txt will be generated....................***Keep this file for future use***



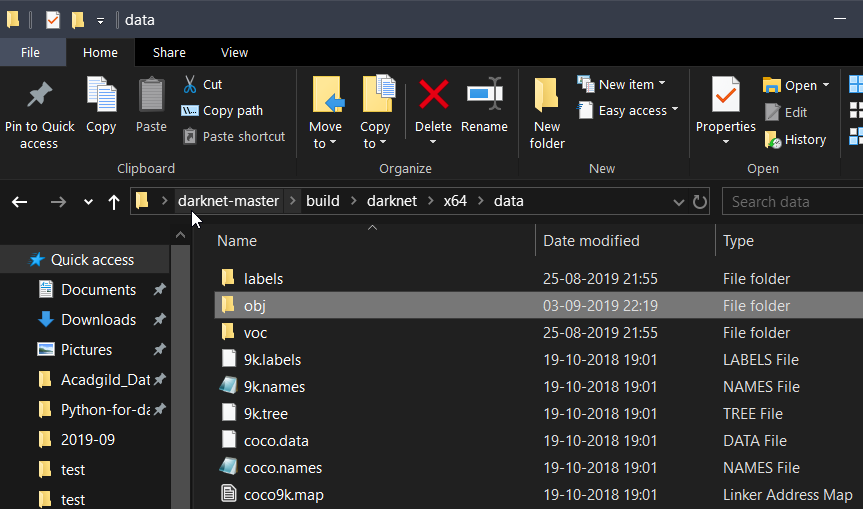
**Custom Training of Yolo(Tiny)**

**Step 1:**

Inside the X64 folder...There will be data folder.

Go to the data folder

Create a new folder named obj



**Step 2:**

train.txt was created in **labelling step**....so copy the file and paste inside **data** folder

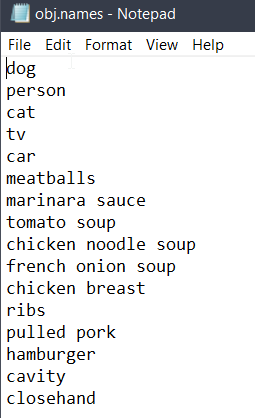
**Step 3:**

Move all images from **images** folder and paste in **obj** folder

Move all .txt files from **txt** folder and paste in **obj** folder

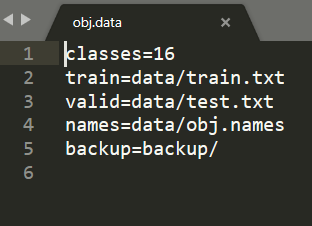
**Step 4:**

Create a new file called obj.names and enter all the class names acc to urs



**Step 5:**

Create a new file called object.data and enter



**Step 6:**

Go the cfg directory in x64 folder

and copy the file yolov3-tiny\_obj.cfg to x64 folder.

**Step 7:**

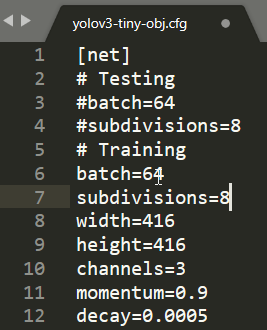
Rename the file in x64 folder from yolov3-tiny\_obj.cfg to yolov3-tiny-obj.cfg

**Step 8 :**

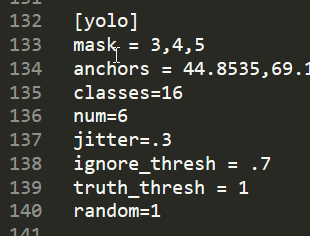
We will modify this cfg file now...Open in text editor and change the lines

1. **batch = 64**

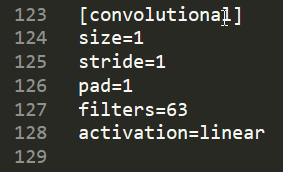
2. **subdivisions = 8**



3. change line **classes=80** to your number of objects in each of 3 [yolo]-layers



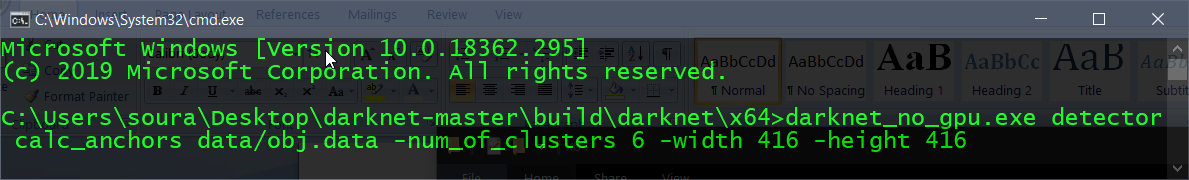
4. change **[filters=255] to filters=(classes + 5)x3** in the 3 [convolutional] before each [yolo] layer



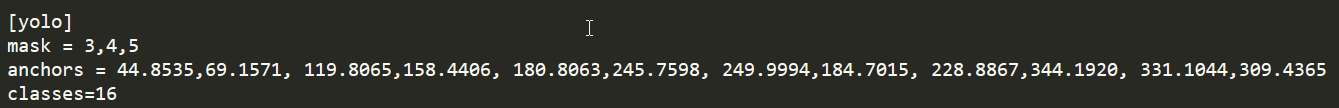
**Step 9:**

Run this command to get the anchors value , to be inserted in the cfg file

darknet\_no\_gpu.exe detector calc\_anchors data/obj.data -num\_of\_clusters 6 -width 416 -height 416



Copy the result of anchors value from console and paste in anchors variable in yolov3-tiny-obj.cfg.



*\*\*\*\*Note: if error Out of memory occurs then in .cfg-file you should increase subdivisions=16, 32 or 64\*\*\*\**

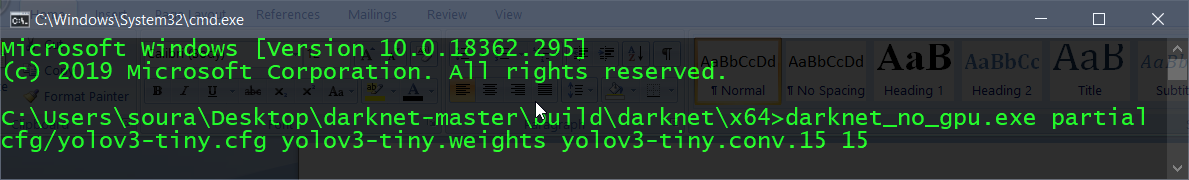
**Step 10:**

Download default weights file for yolov3-tiny: <https://pjreddie.com/media/files/yolov3-tiny.weights>

**Step 11:**

Get pre-trained weights yolov3-tiny.conv.15 using command:

darknet\_no\_gpu.exe partial cfg/yolov3-tiny.cfg yolov3-tiny.weights yolov3-tiny.conv.15 15



**Step 12:**

Start training:

darknet\_no\_gpu.exe detector train data/obj.data yolov3-tiny-obj.cfg yolov3-tiny.conv.15 -dont\_show

