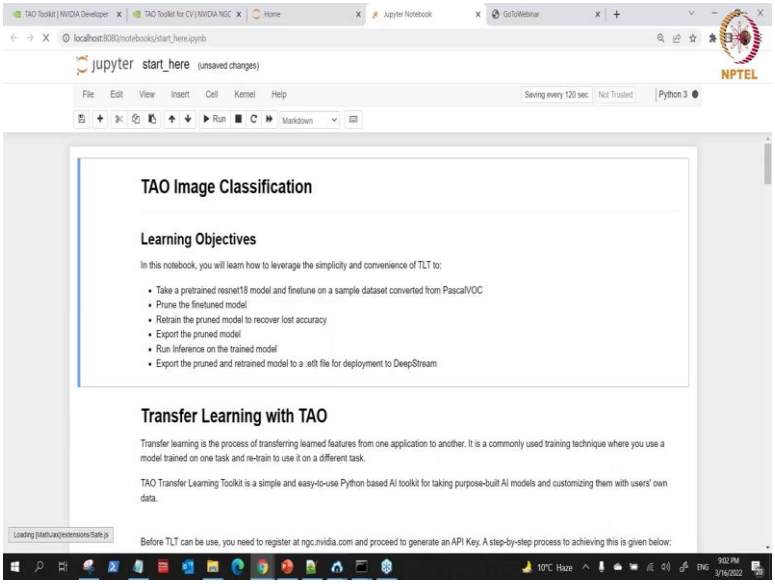
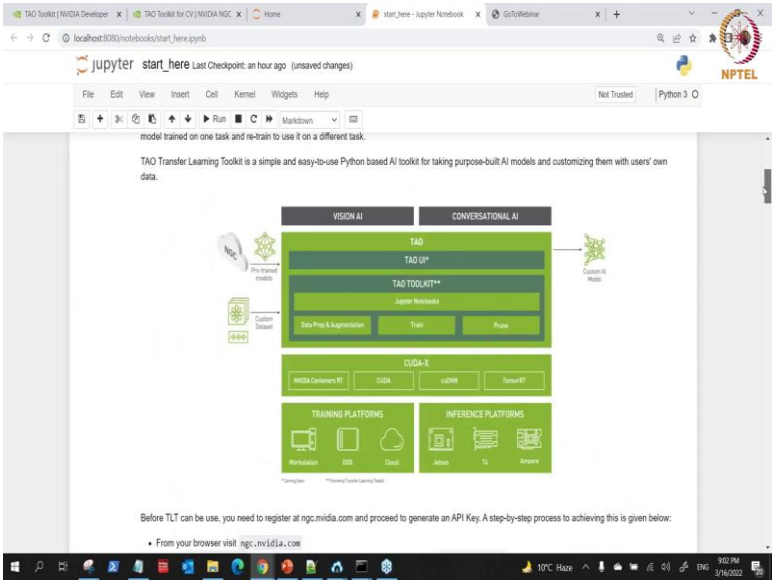
Applied Accelerated Artificial Intelligence
Dr. Tosin Adesuyi
Department of Computer Science and Engineering
Indian Institute of Technology, Palakkad

End to End Accelerated Date Learning
Lecture - 33
Optimizing Deep Learning Training: Transfer Learning Part - 2

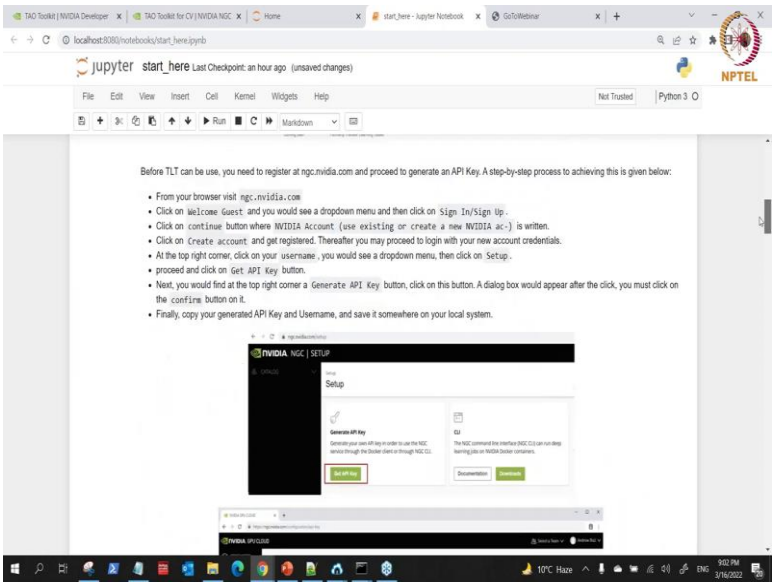
(Refer Slide Time: 00:14)



(Refer Slide Time: 00:19)

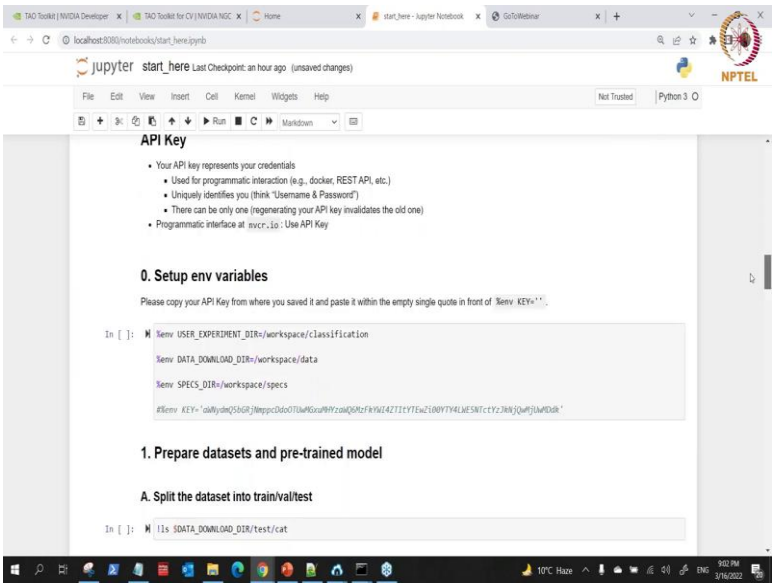


(Refer Slide Time: 00:21)



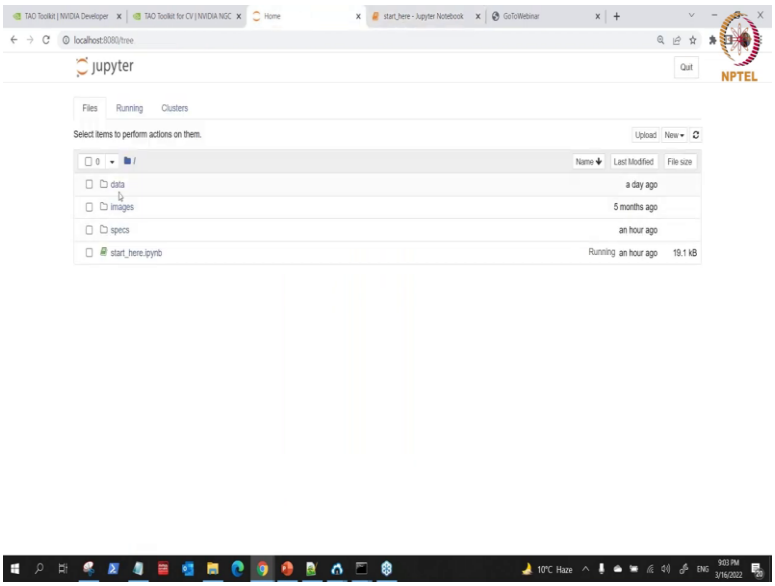
So, here we go. Our notebook is ready. So, next thing to do is I explain all what you can see here. So, let us just go quickly, so that we do not waste time. We go to run the notebook.

(Refer Slide Time: 00:23)



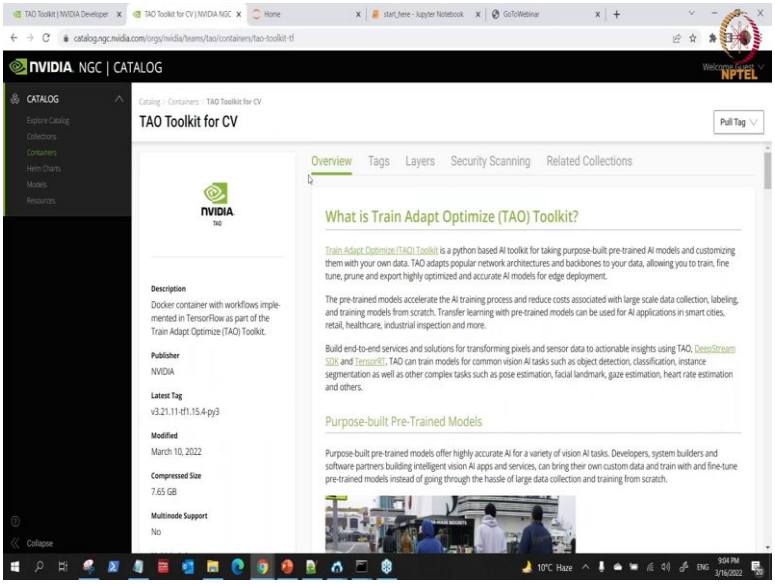
So, the first thing we have to do is to set our environment variable. We set the environment variable to USER_EXPERIMENT_DIR. So, it to be on the workspace classification. This one is by default. Then, the next one is what where we have our data.

(Refer Slide Time: 00:45)



So, we are setting this path, this is what we are setting this path to data. So, this is to data. Then, we set for our specification workspace to the specification file. This is for this specification file the path. So, why we said this is that we do not want along the line we need those path and we do not want to be writing lengthy path that way. So, it will just be represented by all these variables there.

(Refer Slide Time: 01:32)

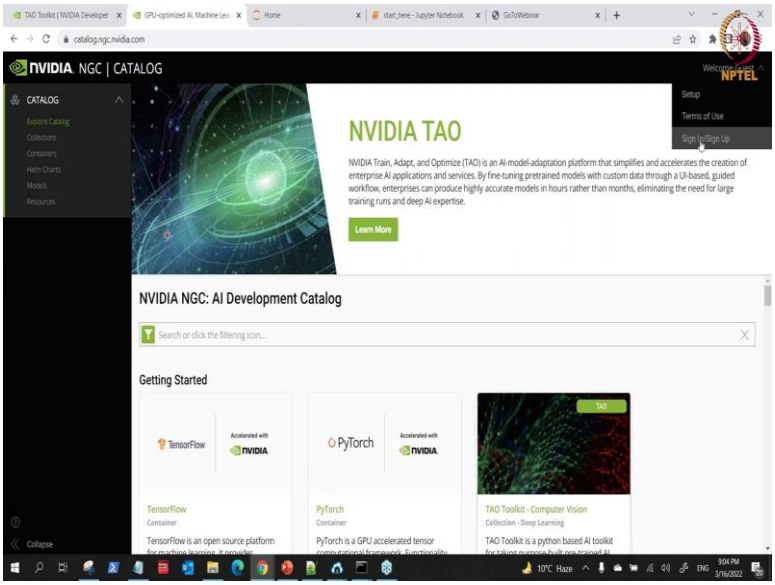


So, the next one is we need environment key. Now, the environment key, you need to have an environment your key for the TAO key. So, what you need to do API key, is

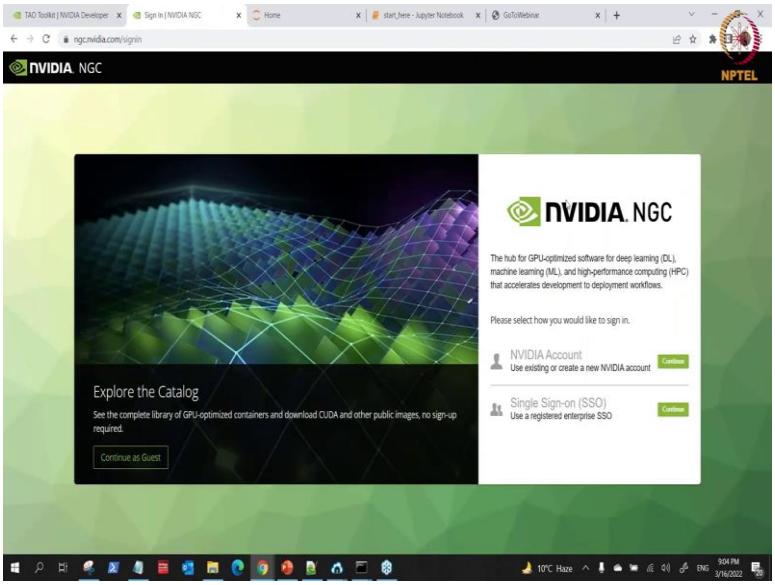
called API key rather; you need to register on the NVIDIA NGC forum. These are the steps you need to follow, but I will show you the step. Now, let us quickly run that.

So, when you get here you know you pull, when you pull your container you do not need to, you do not need any registration, but now you will want to get our API key. So, what we need to do is what you come here; ok.

(Refer Slide Time: 01:50)



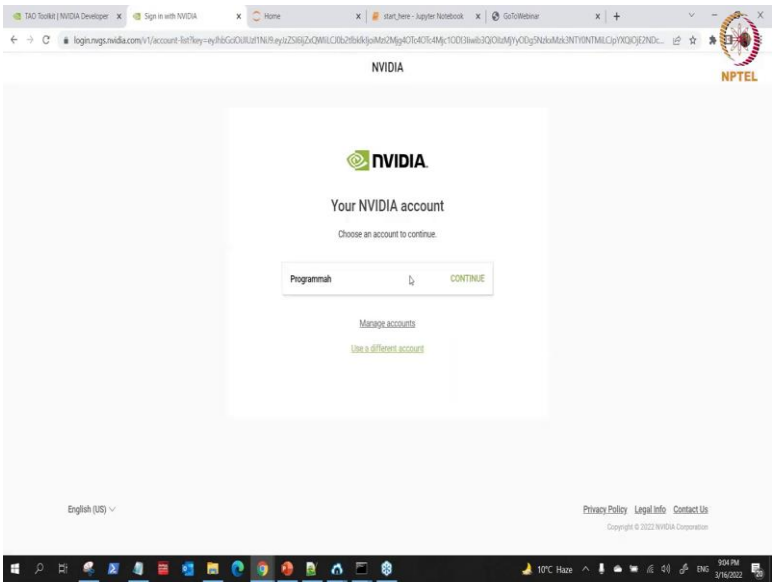
(Refer Slide Time: 01:57)



Let me go back to and you see here this is the first page.

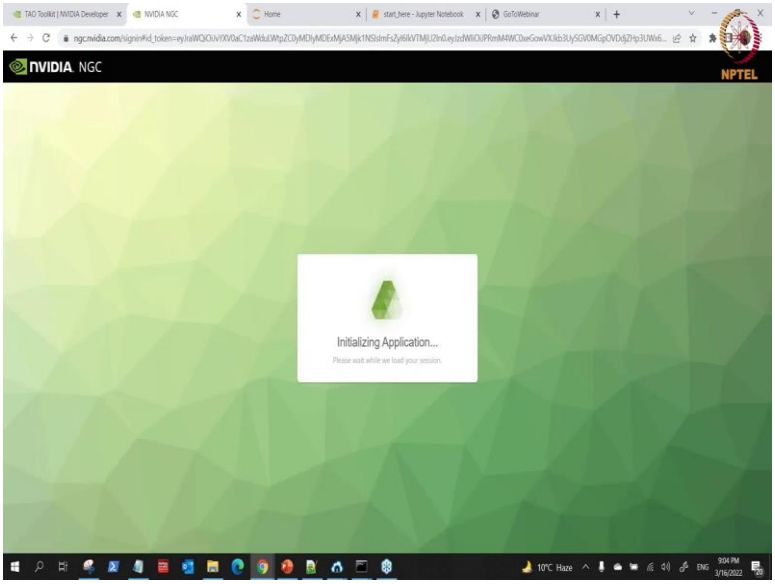
You see where you have guest here. So, what you need to do is what you click on sign in or sign up, sign in or sign up. So, when you have your sign in or sign up, when you get here, you click you make use of the NVIDIA account, use existing account continue, that is the one you click, first one.

(Refer Slide Time: 02:09)



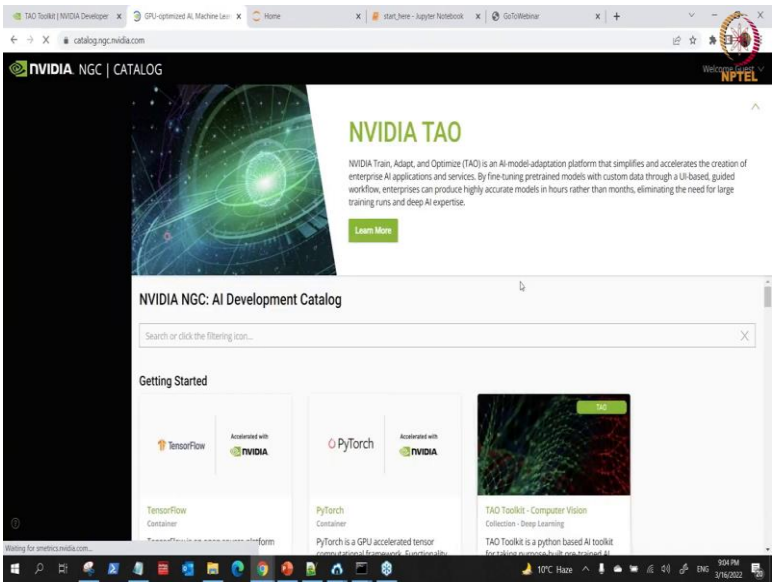
So, because I have an account, I can continue this way. For those that have not registered at all, you can use a click on different account. You will see where you sign up. So, after you sign up you can come back and then log in. So, once you log in like the way I am going to, it is going to allow me to log in here.

(Refer Slide Time: 02:29)



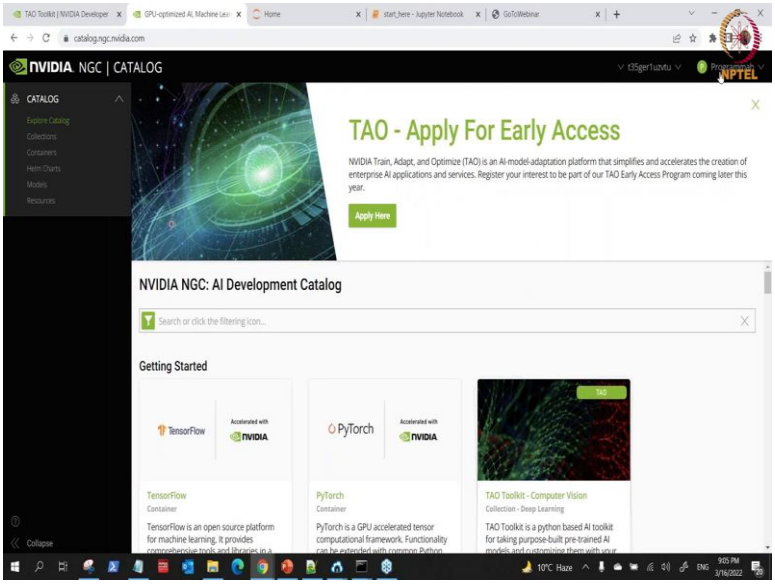
So, I am logging in now.

(Refer Slide Time: 02:33)



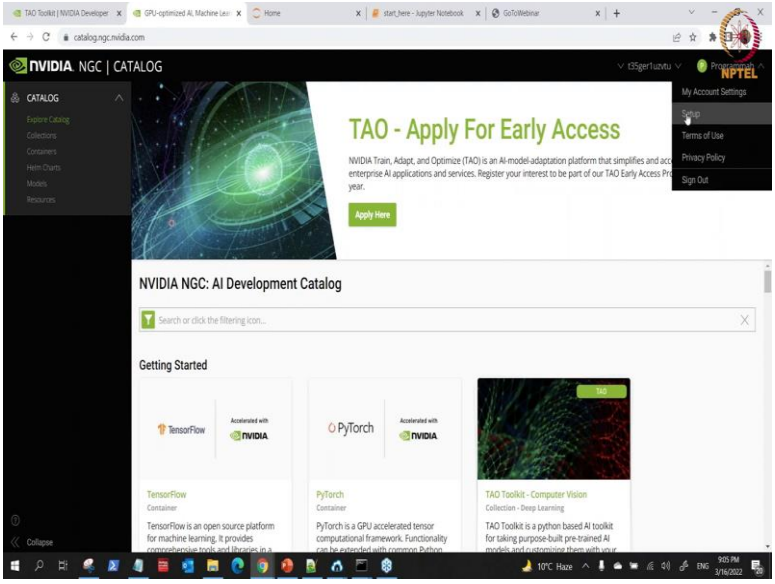
So, yes what I am looking for login then you see this place will change it will no longer be guest.

(Refer Slide Time: 02:34)

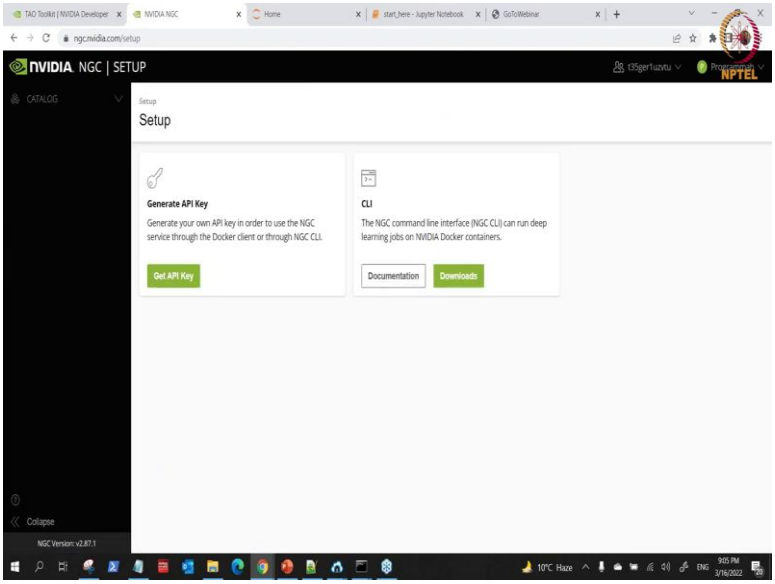


You see my, it has changed my name here programmer.

(Refer Slide Time: 02:43)

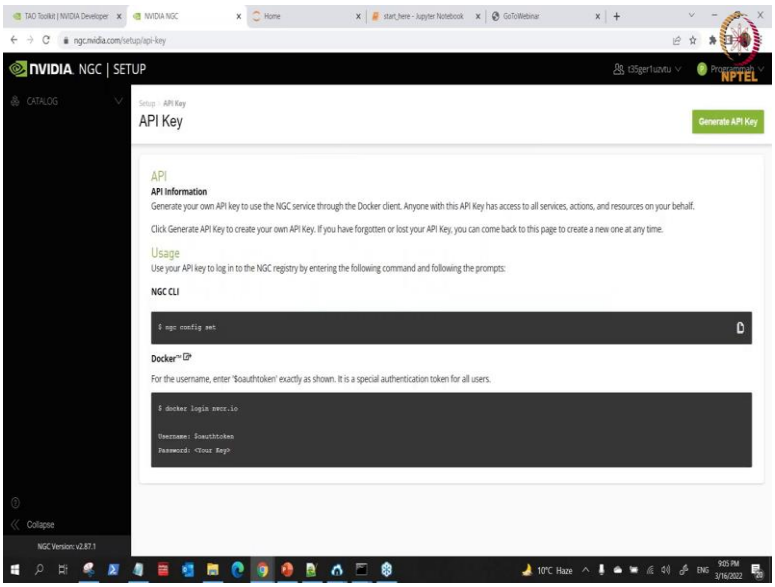


(Refer Slide Time: 02:48)



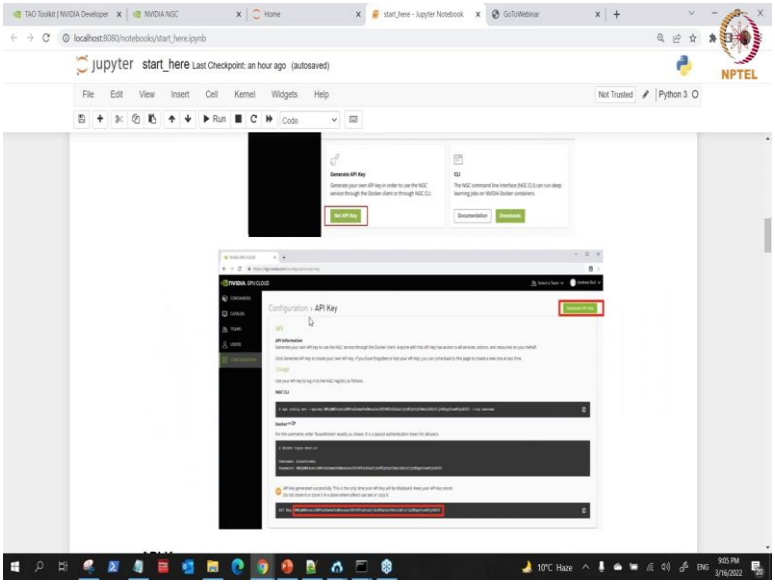
So, what you need to do next is to drop it down and look for click on word setup. So, you click on setup and it will load. So, you need an API key. So, you can click on the get API key.

(Refer Slide Time: 02:54)



So, when you click on get API key, the next phase to do is to work to generate, to click on generate. But because I have key, I do not need to click on generate again. So, we can come back here and say see.

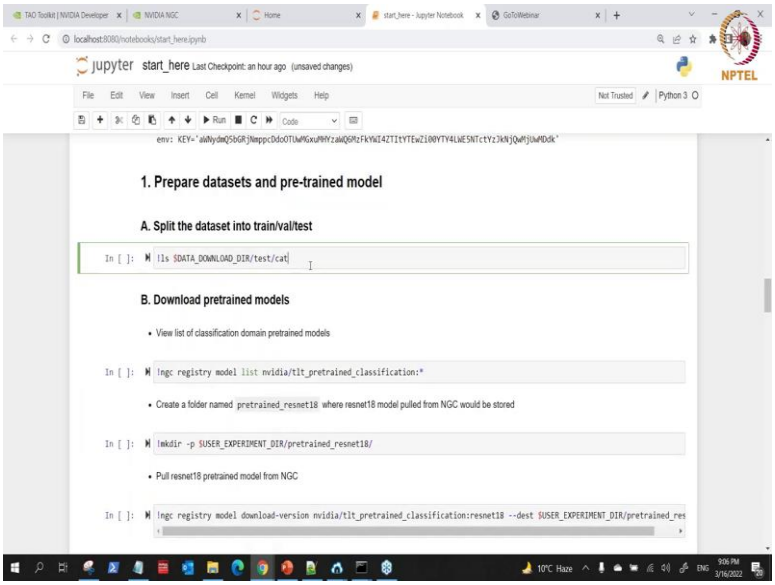
(Refer Slide Time: 03:08)



So, when you click on the API key, get API key, it will bring this interface. So, on this interface you click on generate, when it generates here, you will see the API key here. So, you copy it and save it somewhere. So, anytime you want to work with you using your NVIDIA TAO, you can now work paste the API key. Make use of it here. So, this is our API key which is being used.

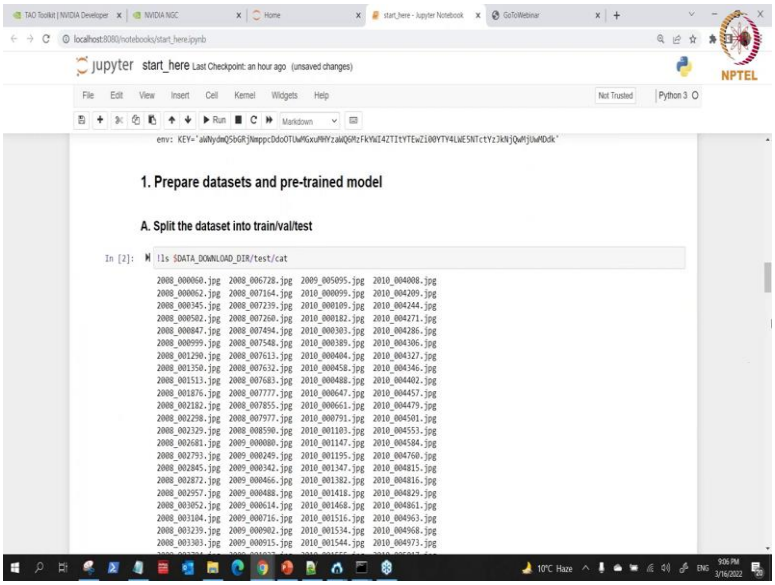
So, the first thing to do now is to run this cell. So, I am going to run this cell. So, everything is initialized this way, running this cell.

(Refer Slide Time: 03:41)



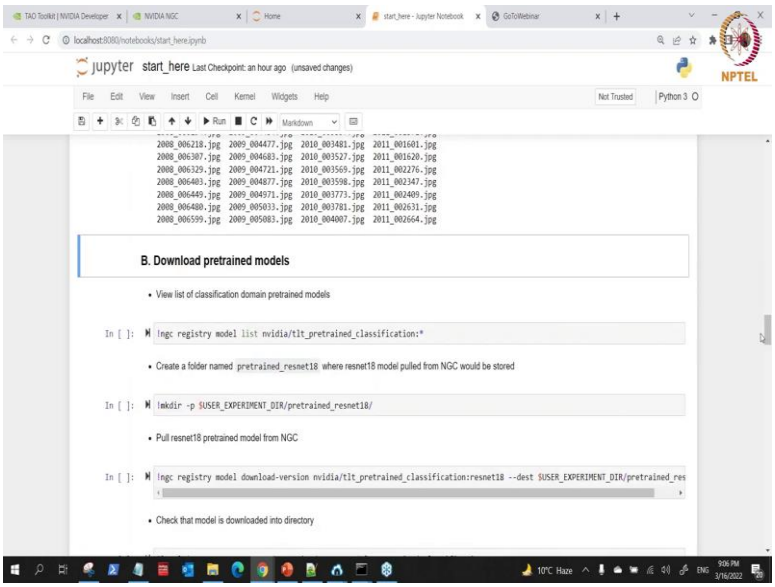
So, the next one is to let me see I want to know if my data are visible. So, I want to look at the test data to look at what just one class is cat.

(Refer Slide Time: 03:52)



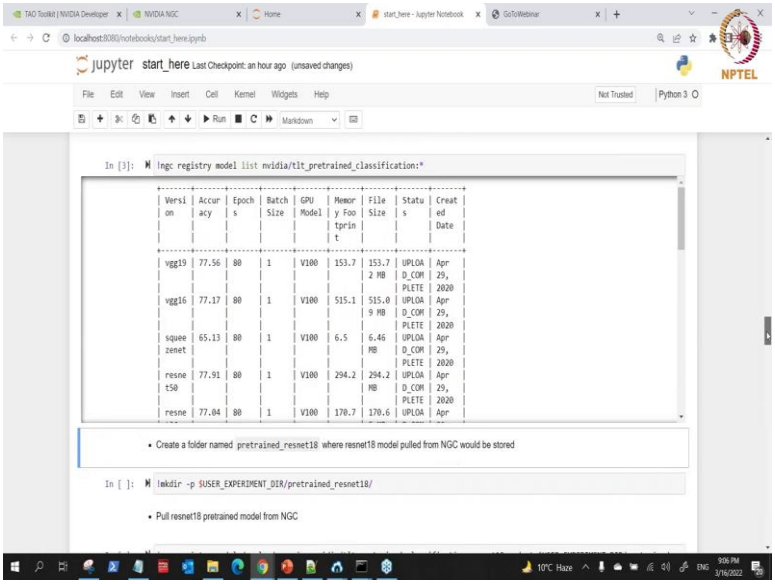
So, it can fetch that is, ok it is reachable, our data is reachable.

(Refer Slide Time: 03:57)



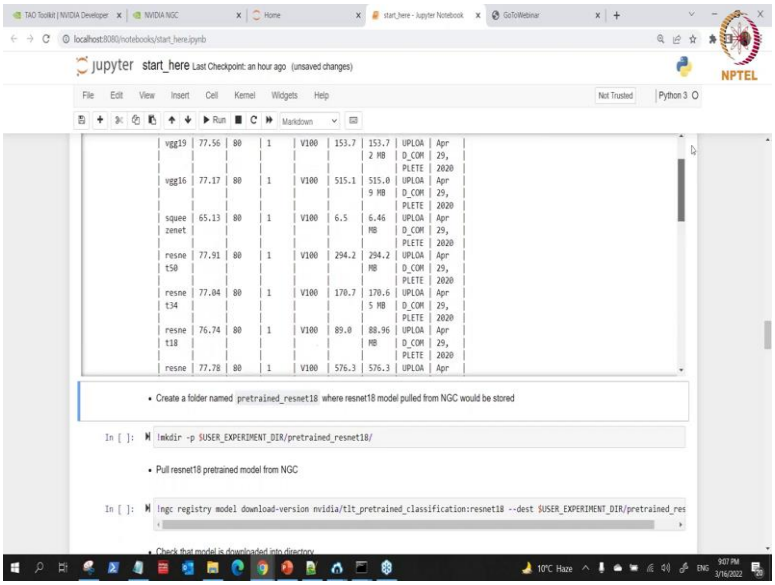
Then, the next thing to do is to see what are the available, what are the available pre-trained model for classification that exists here.

(Refer Slide Time: 04:25)

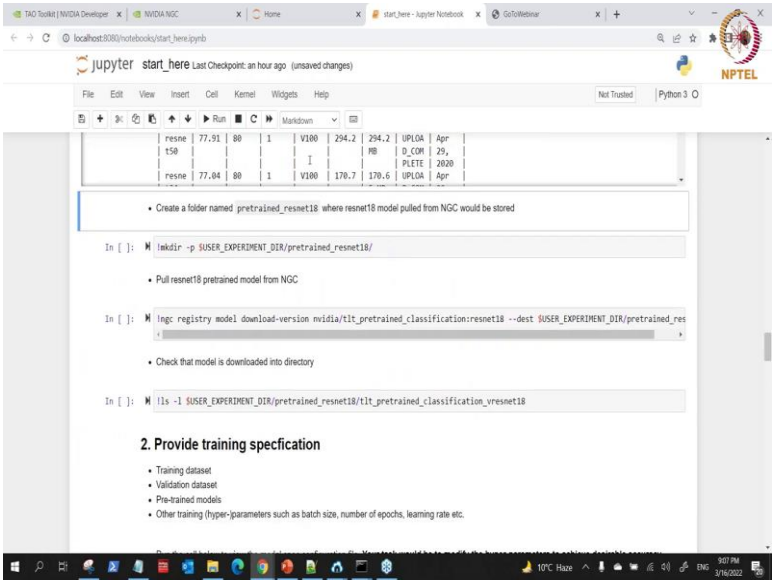


So, with this ngc registry model list, it will list them nvidia/tlt_pretrained_ classification**:** ***, that is all. If you show all it can show all to you. So, we will be able to see the list of all the, yes it has shown all of that. So, you can see versions vgg19, vgg16, their accuracy, the epoch you can see. So, there are many of them there resnet 50, 34, 18, 101 and all of them. So, we can make use of you can make use of any of them that are there. So, we can make use of them.

(Refer Slide Time: 04:44)



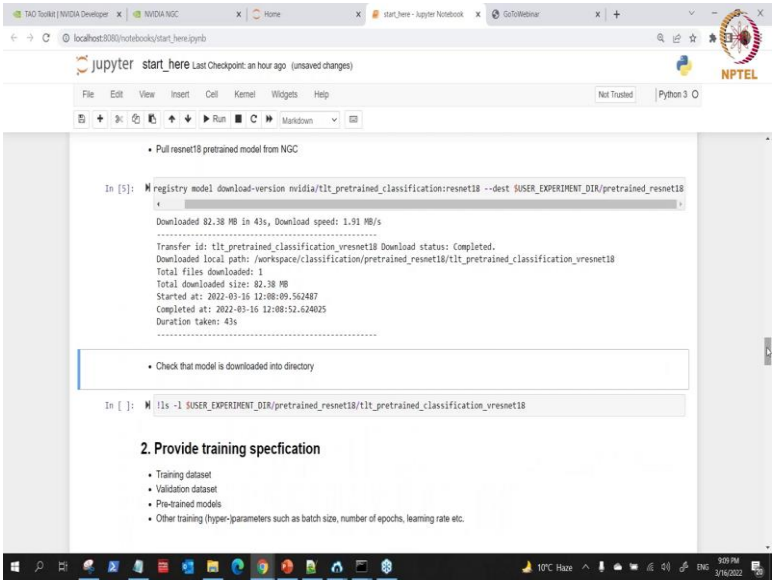
(Refer Slide Time: 04:52)



So, the next phase not to waste time. So, is to what; since we know the pretrained model the that exists we can now make a folder directory where we will save our downloaded pretrained model.

So, let us create a directly call pretrained resnet18 because that is what we want to use. So, I am running that. So, that is done. You can see when its gives a number it shows it has run. And then, so now, let us pull that what pretrained model into this directory, yeah. So, how do you put that? You have, do you when you write ngc registry the model, then download version. So, we have nvidia/tlt_pretrained_ classification. So, what kind of model? Resnet18.

(Refer Slide Time: 05:46)

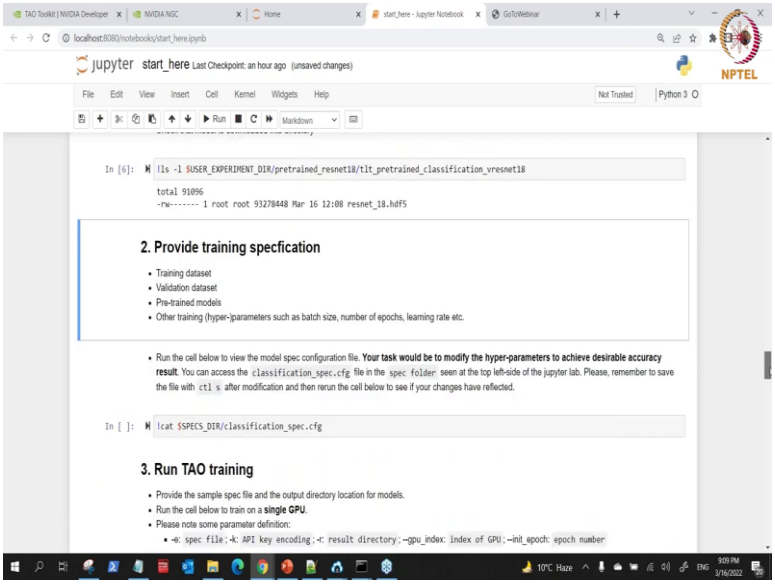


The destination you should save it here because you know we created this folder this one we created it here. So, it is going to you should save it in this destination. So, I can run that now. So, it is going to run and we can see our model will be downloaded. So, the download has begun.

So, why that is loading? So, once that is completed, the next thing to do is to just verify and see whether it is the directory is the right directory. So, just confirm if it is a right directory because of the ssh it should have been much more faster than this I guess, should be faster than this. So, that have not done, ok, yeah. So, it is done. So, the model has been work has been unloaded.

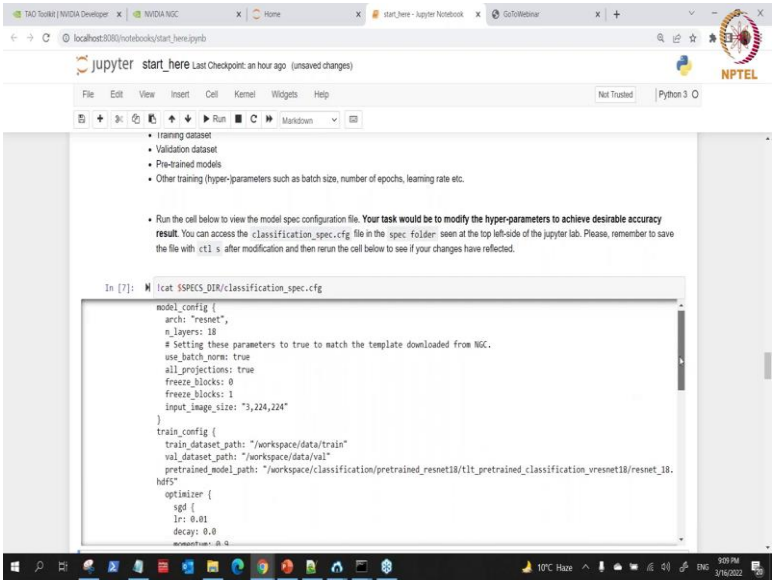
So, we can view the model here. So, we so, the model resnet _18.hdf5. So, that has been done.

(Refer Slide Time: 06:41)

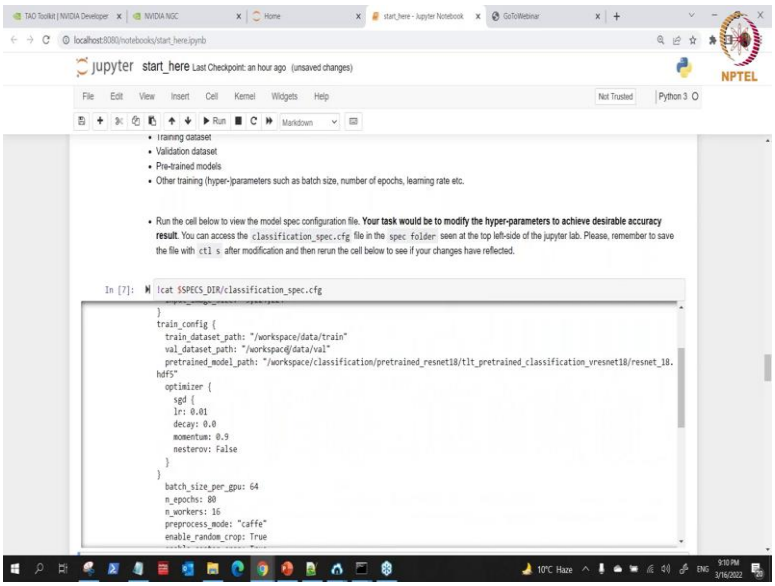


So, the next thing to do now we want to train. Since, we have our model, we have our data set, we want to train now. So, before we train like I have explained earlier on that, you need to configure your configuration that is specification file and this is the path to the configuration file, the specification file, this file, yeah. So, we want to see the one for training is the one we call the classification and that is called spec.cfg.

(Refer Slide Time: 07:14)

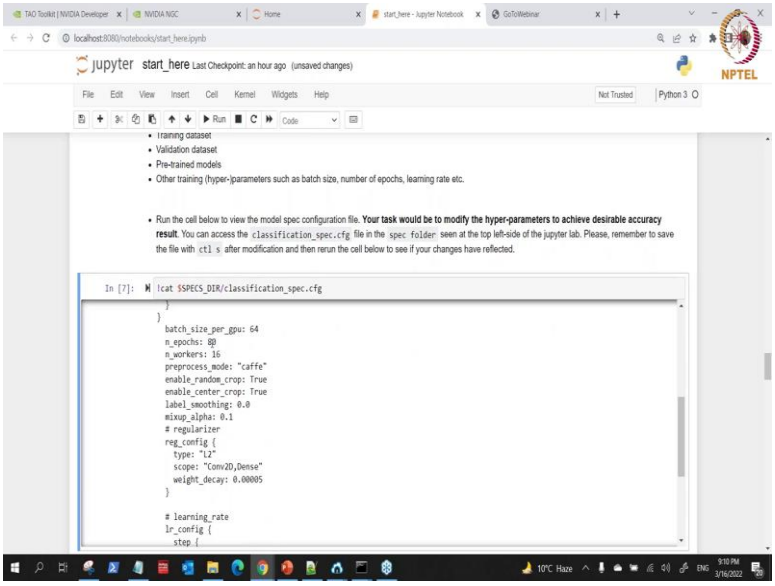


(Refer Slide Time: 07:36)



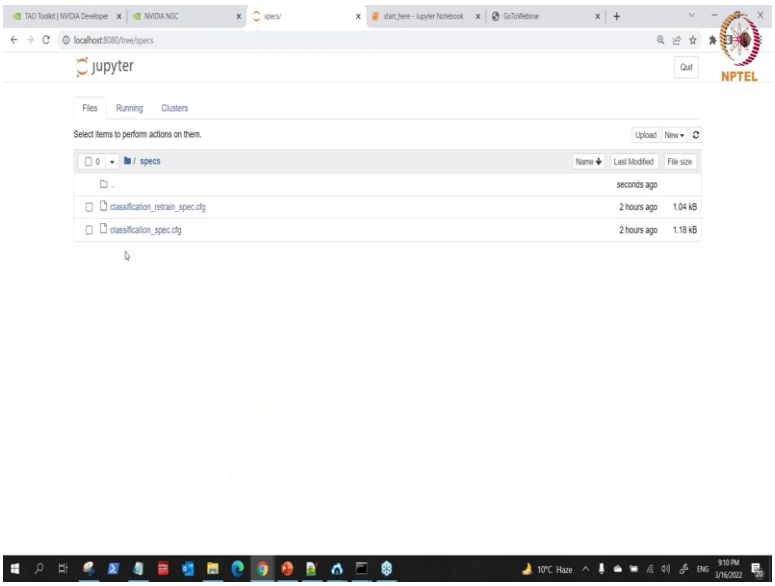
So, we can open that and see what is inside. So, if you look at it this is what I showed you before. The architecture is what resnet, the layers 18, our input is 3; 3 by 224 by 224. The path to our training, the path to our validation set, the path to our model to be trained all of that you can see here.

(Refer Slide Time: 07:51)



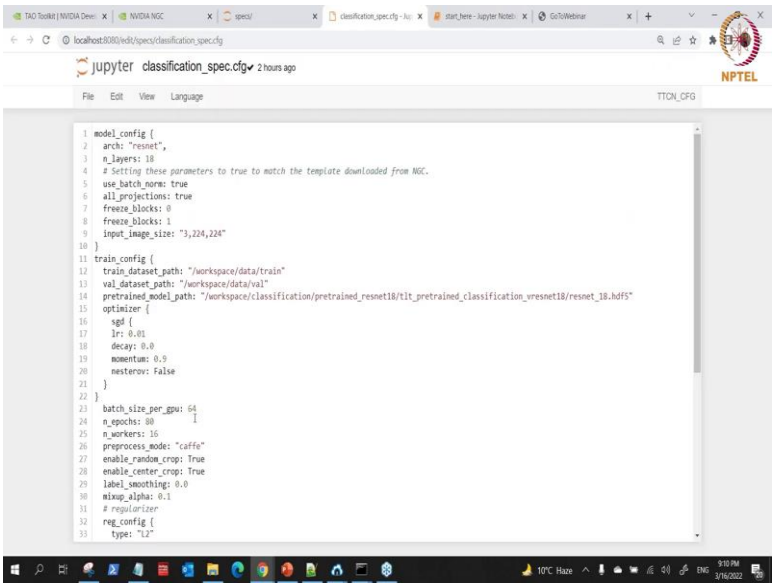
And then the number of epoch, now this is 80. We cannot use 80 now because this is for demo. So, otherwise we want to sleep here. So, we need to change this for this demo, yeah.

(Refer Slide Time: 07:56)



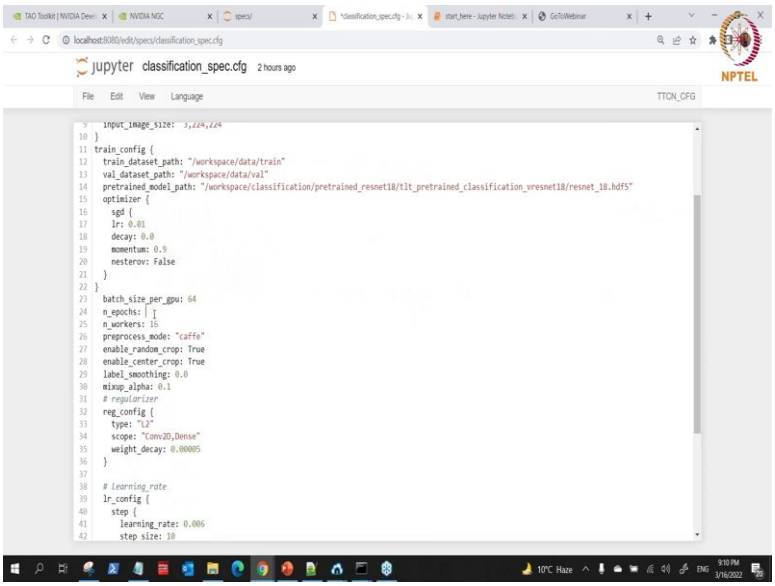
So, for this demo. So, we can come here, I can come here. So, if you are training on your workstation you do not need to change because that is your own time, but now we just have limited time.

(Refer Slide Time: 08:07)



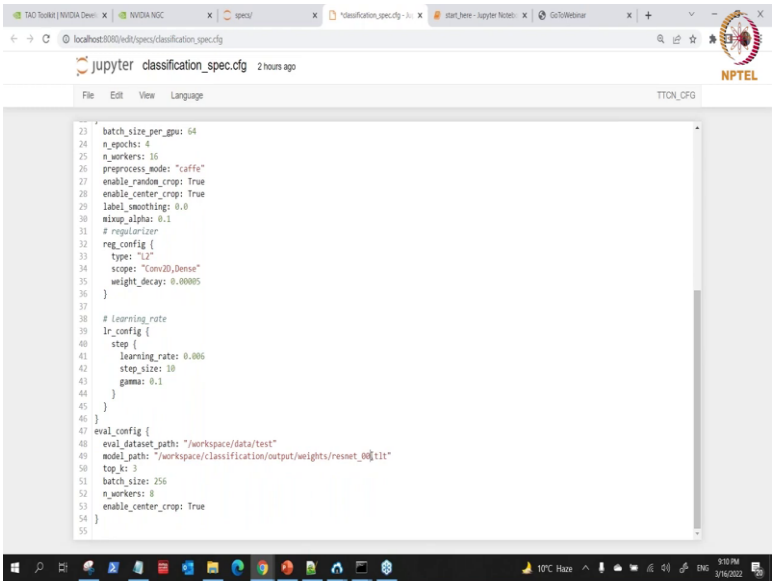
So, what I will do is what I open this, this is the one for training, then I will look for where we have the 80 just to show you, just to show. So, we are less concerned about accuracy now because just use 4 epochs. So, that it can train faster.

(Refer Slide Time: 08:13)



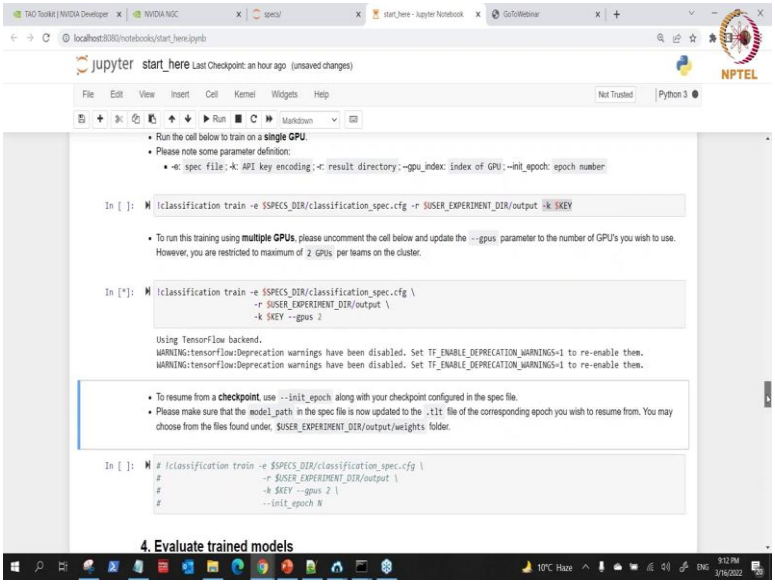
And when you say, 4 we have to consider the evaluation also because this is for 80. So, you have to change, I have to change this also.

(Refer Slide Time: 08:28)



Well, you do not need to bother about this. This is your training for me ideally you should train with 80 epoch or more because for training because for demo that is why I reduce these to just 4 epoch. So, I have saved that, then I close that, I get back here.

(Refer Slide Time: 08:57)

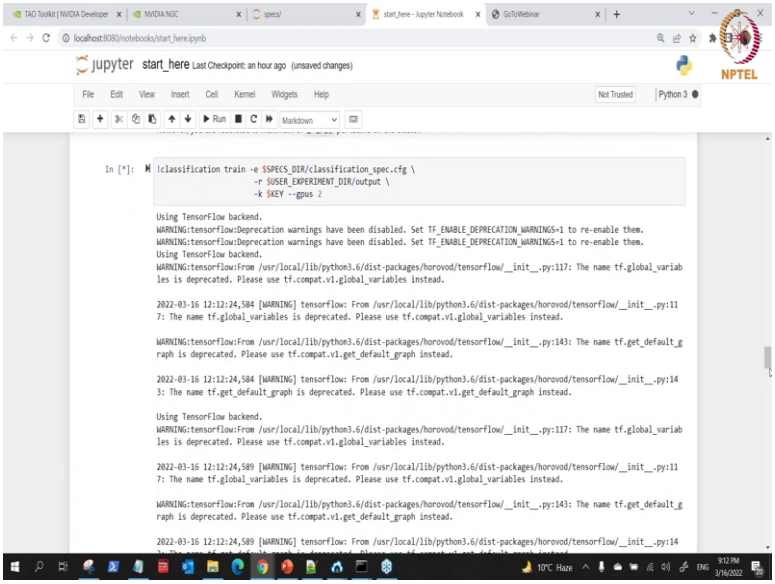


So, let me relaunch this one again and see, ok it has been affected. So, we can go now. No train for just for epoch. The result might not be good, but we are not bothered by that one for now because we just want to show you demo.

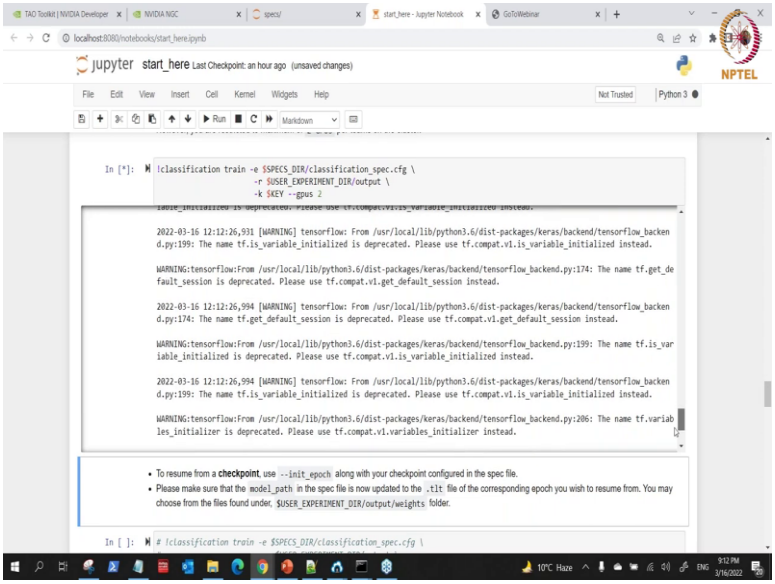
So, here you can train if you want to train you have classification, then you this keyword train. So, these are flag here, here what is specify is that it is referring to the specification file which is just configured now. And then so, when it finished training where does it need to save it, this one refer to that it will save it in a folder called output. And then it needs an assets. So, this is the key assets. So, these are the flags to use to do that.

So, this in this is meant for just single GPU. So, because I have two GPUs, let me use two GPUs. This is meant for two GPUs here. So, I can just specify this as. So, if you want to specify number of GPU, this is how you do that --gpus and 2. If you have 4 GPUs on your workstation, it will be work 4. If you have 8, it will be 8. So, I have just to; so, I can use this now and run this.

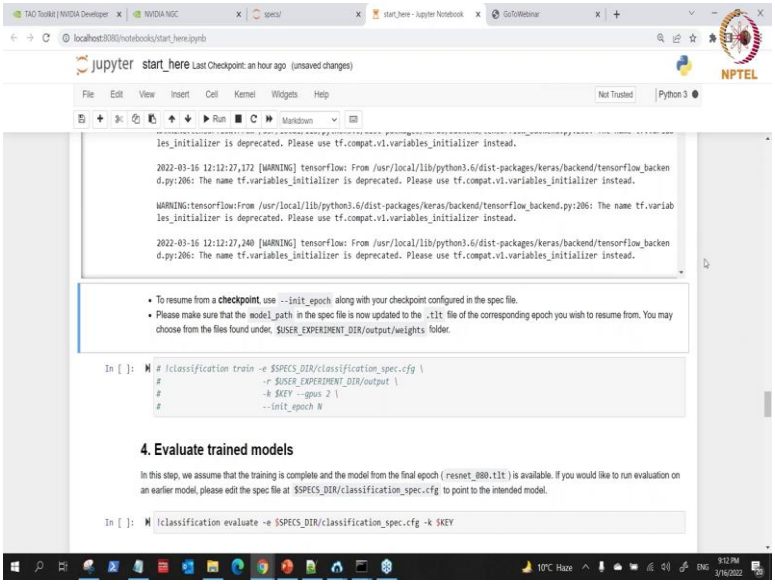
(Refer Slide Time: 10:02)



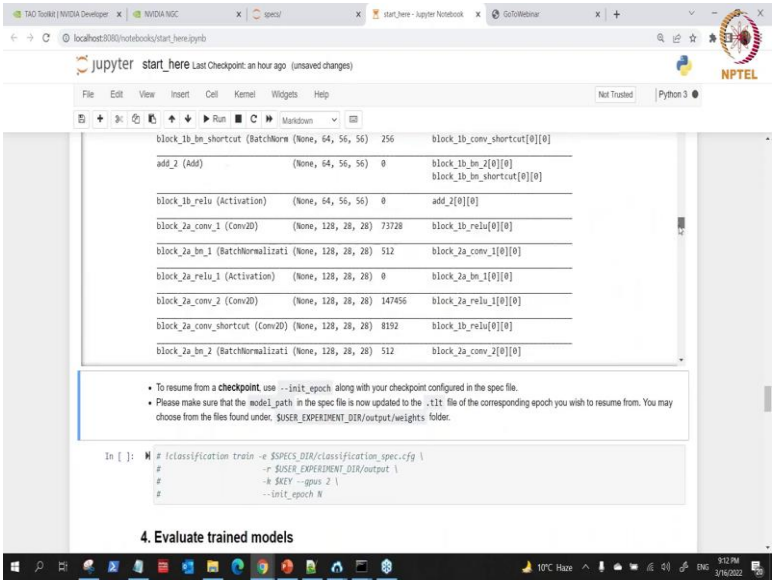
(Refer Slide Time: 10:08)



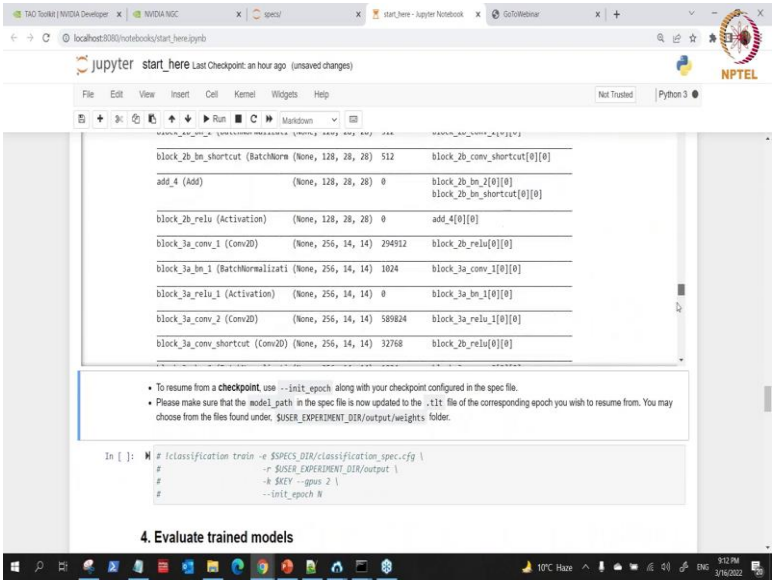
(Refer Slide Time: 10:20)



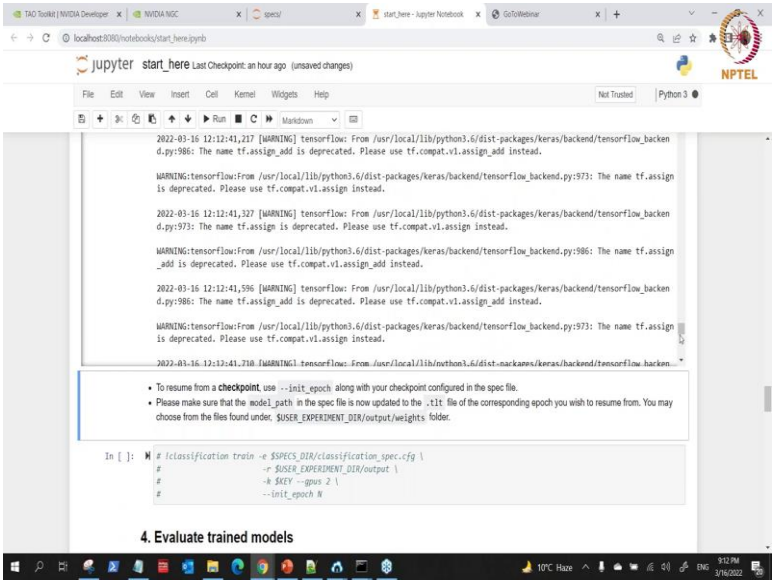
(Refer Slide Time: 10:24)



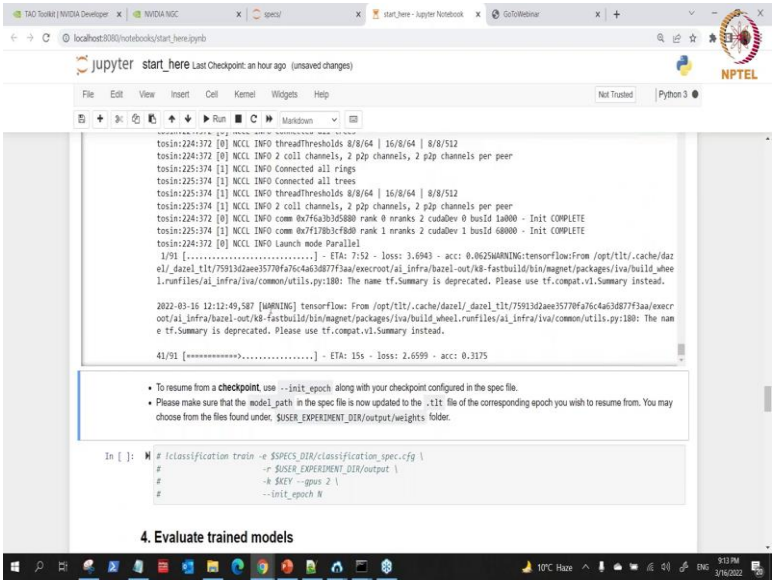
(Refer Slide Time: 10:27)



(Refer Slide Time: 10:30)



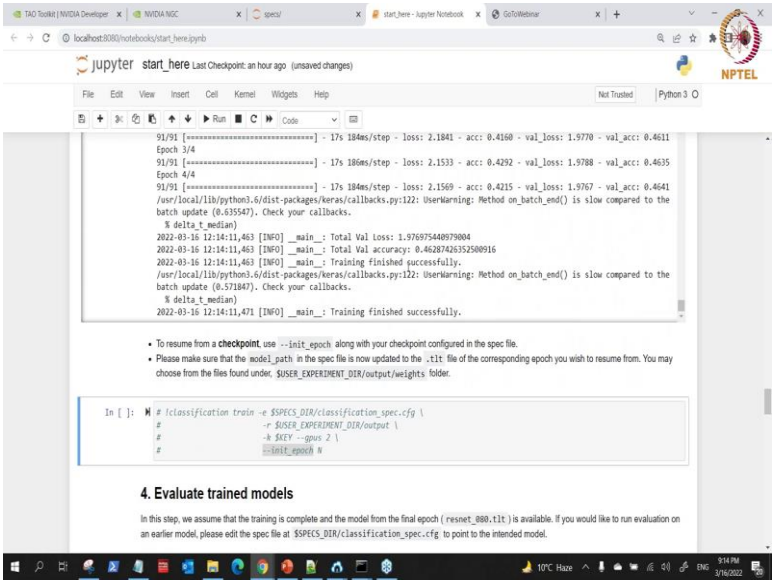
(Refer Slide Time: 10:34)



So, it is running now, its running and see it is running. So, it is running. So, ok can see. So, this are the is loading the what the layers of the model here, exactly, loads them and then it will continue to the epoch. So, you can see here to start running the initial epoch. So, its run them epoch by epoch. So, we are on the second epoch now, so it do not take us more than 2 minutes to do to complete everything then.

So, the next what you can see here why that one is training. The next phase here is I have commented this one out, this one is meant for maybe your trained and you are unable to complete your training, you just save it somewhere. So, by the time you want to start your training again, instead of you starting from the beginning, you can continue from where it stops.

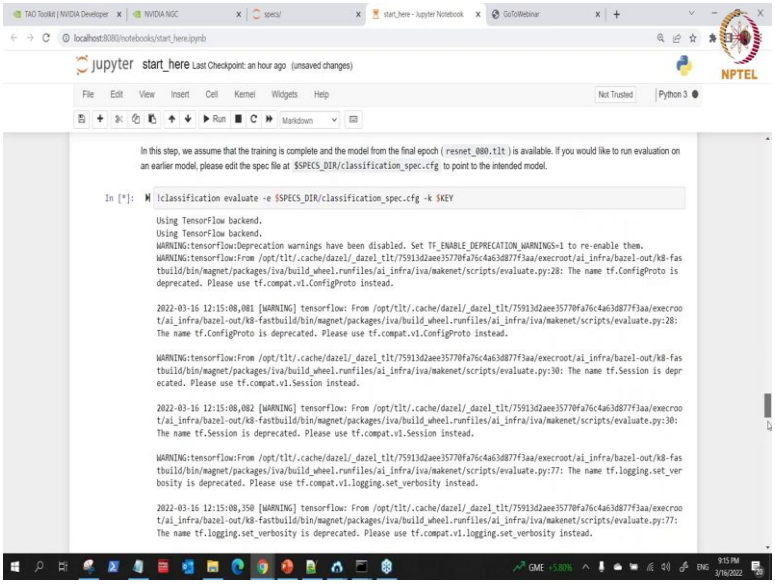
(Refer Slide Time: 11:36)



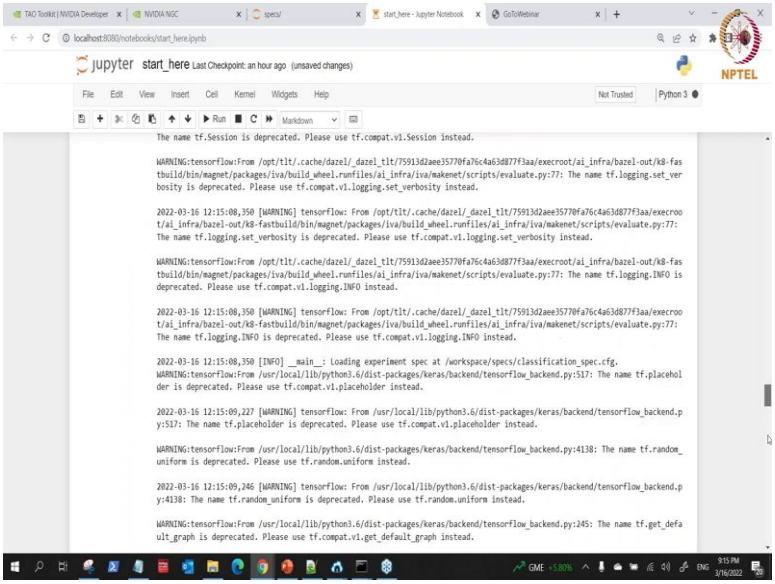
So, you can continue from where it stop by word, specifying the word initial epoch. So, let us say we stop at 4 epoch now. If I decide to continue this training tomorrow I do not need to start from 1 epoch again, I will just continue from the where do I stop 4 epochs. But will not be needing that for now, for this demo.

So, we are, it is almost done here, it is almost done. So, you see it is how do you know it is almost done? When this is it the star. So, now, when it gives it the number everything is done that cell has completed. So, it has finished successfully. We can see this is the loss and this is the accuracy 0.4, but you know because we trained for 4 epochs. So, we are not expecting the accuracy to be that good so, but that is just for a demo, it is just for a demo. If you train with your 80 epoch the accuracy would be fantastic.

(Refer Slide Time: 12:21)



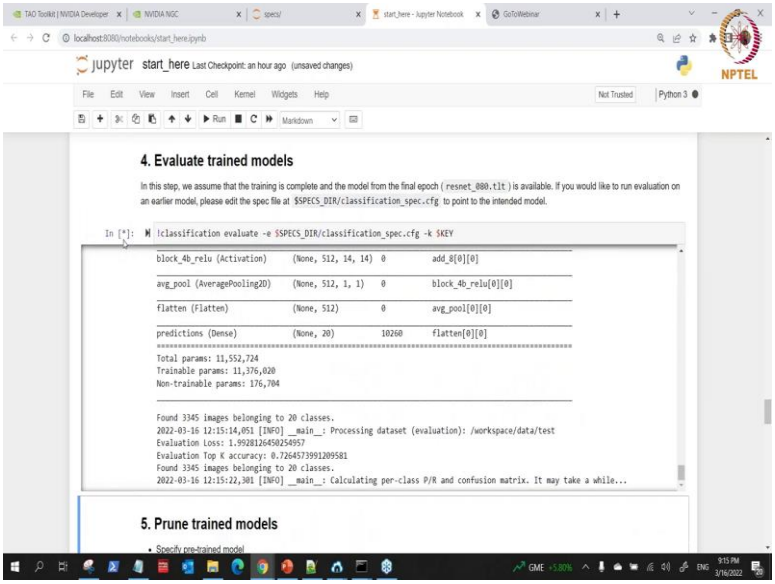
(Refer Slide Time: 12:47)



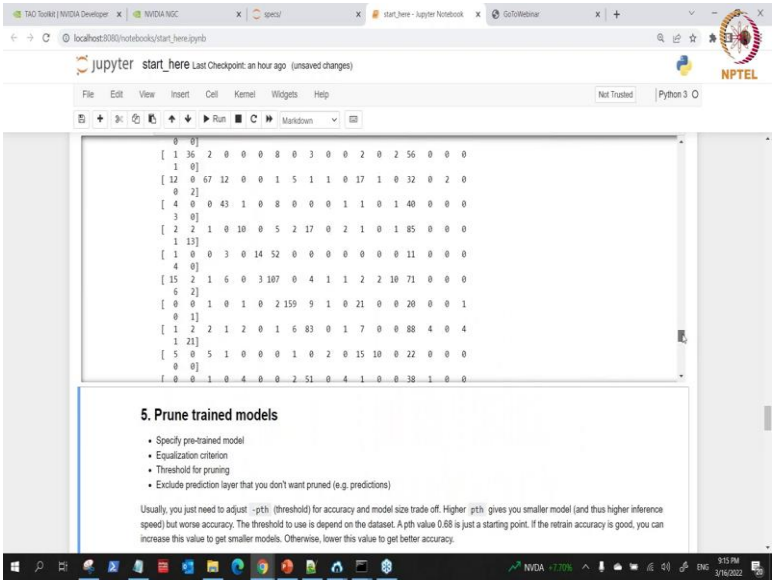
Then we need to evaluate what we have trained. So, how do we evaluate? You have what this to evaluate, then classification, then evaluate you have you specified the word these configuration file which is the specification file and the key also this is what we specified then.

Let me around that is going to do the evaluation. So, during the evaluation if we plot the confusion matrix and also give us the precision and also the recall and all that everything will be given so, it is almost done.

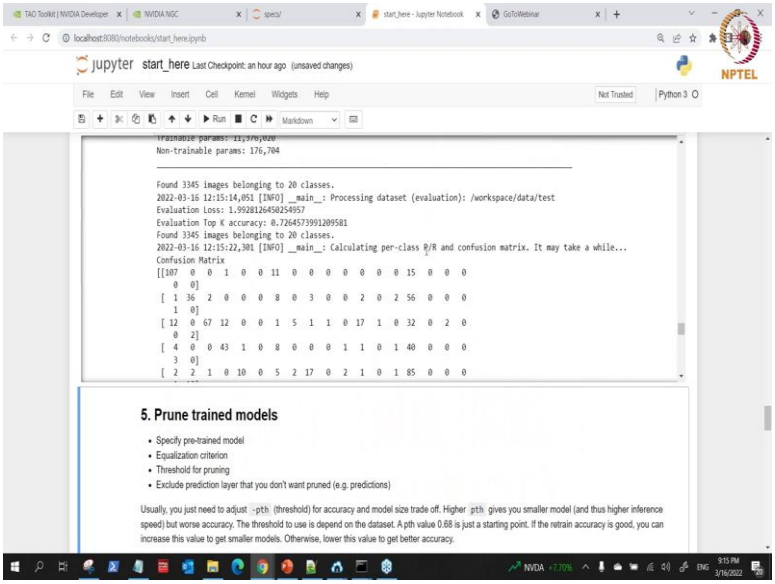
(Refer Slide Time: 13:11)



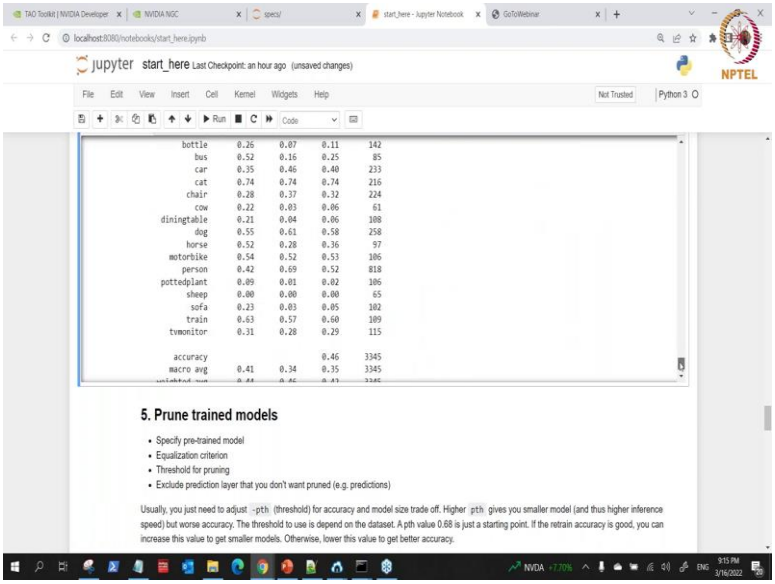
(Refer Slide Time: 13:22)



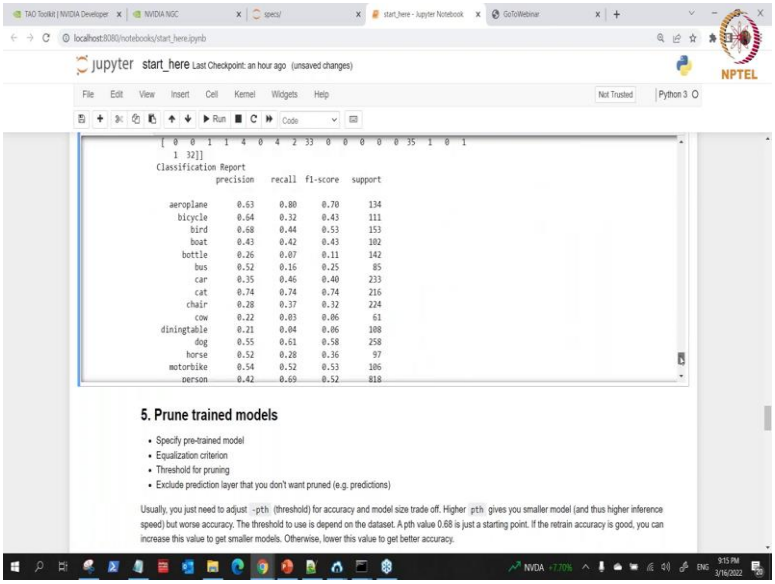
(Refer Slide Time: 13:22)



(Refer Slide Time: 13:26)

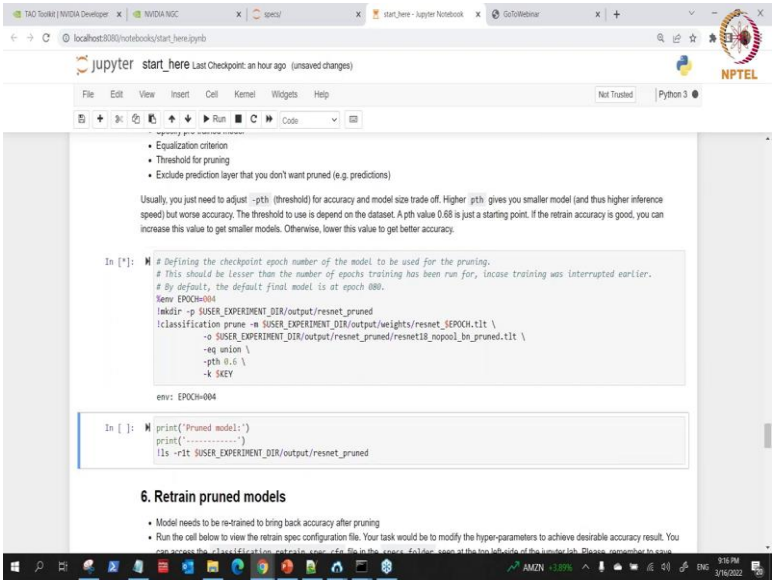


(Refer Slide Time: 13:28)



So, it is trying to what, calculate the precision. And we see all that and once it is done you see it will give it a number here. So, yeah, this is the see the confusion matrix, then it will then generate go to classification reports. You see the classification report here, precision, recall, f1 score, support and all that. So, I think it is done now, yeah. So, we can proceed for that.

(Refer Slide Time: 13:44)



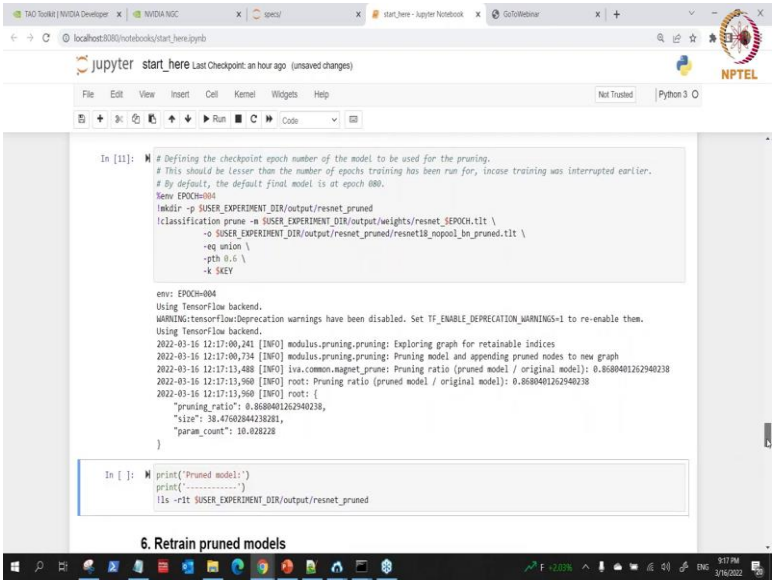
And then, so we can now what prune, yeah, this is where you prune. So, for you to prune we need to create a folder where we store our pruning, create a directory where we start

our pruning. This is the epoch the default epoch is 80. So, because I am using 4 epoch, I am going to swap these to 4. And then, so how do you prune? You have your classification and the keyword prune, yeah.

So, you supply where is the model that was trained. This is where the directory it is. So, where do you want to store when you prune? The prune you will store it here in this directory output. And what percentage do you want to use to prune? We just want to prune 60 percent, this means 60 percent.

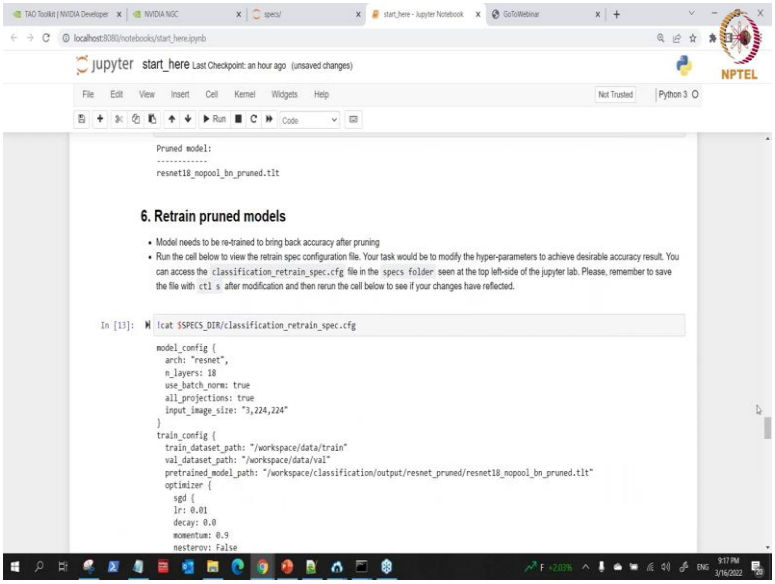
If you specify 0.2, that means, 20 percent of the model will be prune. So, we can run these now and it is going to do the pruning.

(Refer Slide Time: 14:34)

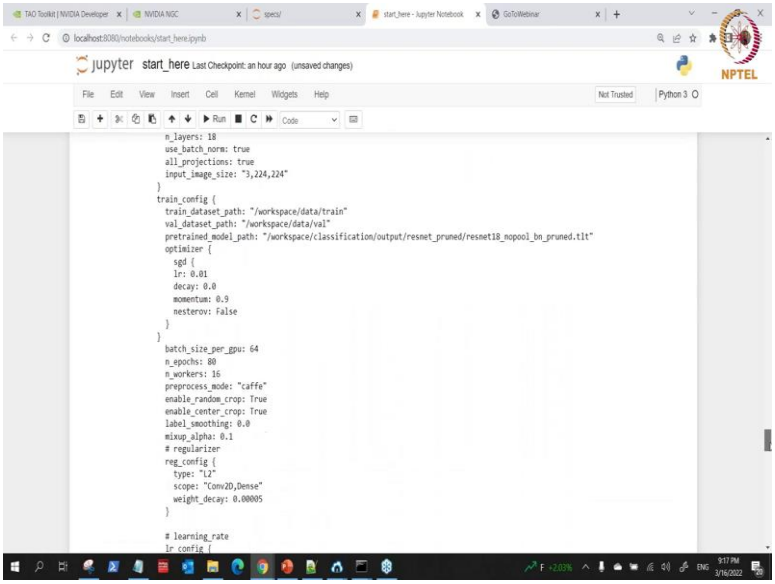


So, why that is going on? So, after the pruning is completed then we can view the file here. We will be able to feel, ok that is done. If that is given it number 11, then we can now see the pruning, yes, ok. This is what it saves. This is the name of what it has saved.

(Refer Slide Time: 15:01)

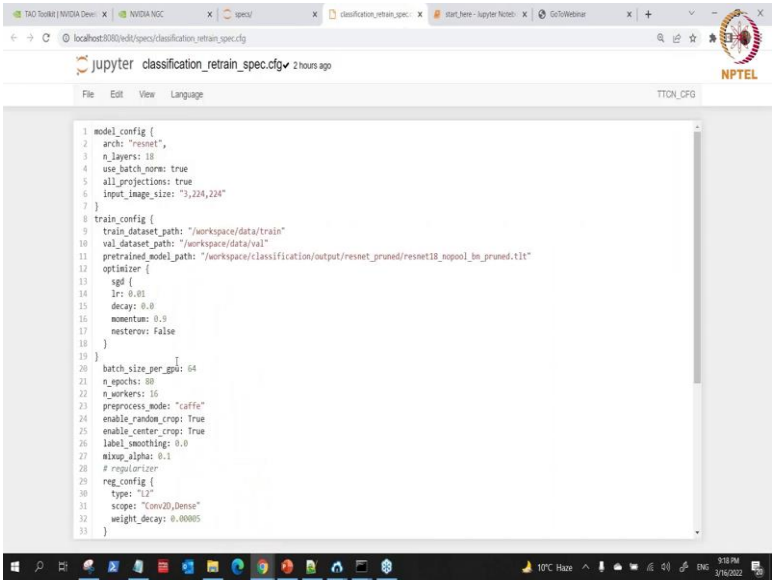


(Refer Slide Time: 15:26)

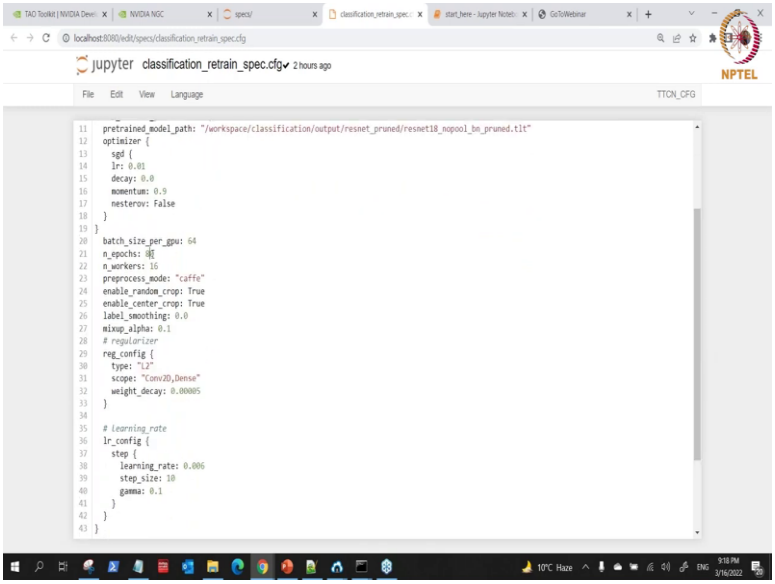


Then so, because we have pruned there is need for us to retrain again. So, when we want to retrain we have to use the word the specification file which is for work for retrained.

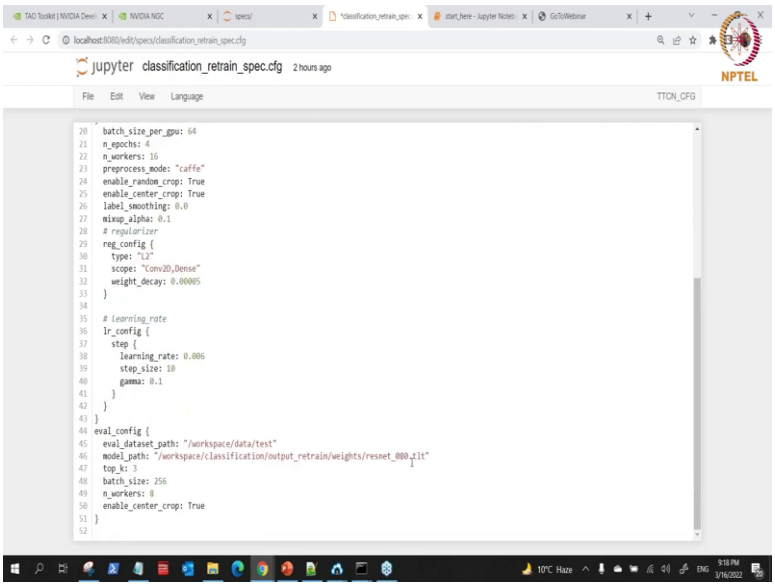
(Refer Slide Time: 15:35)



(Refer Slide Time: 15:41)



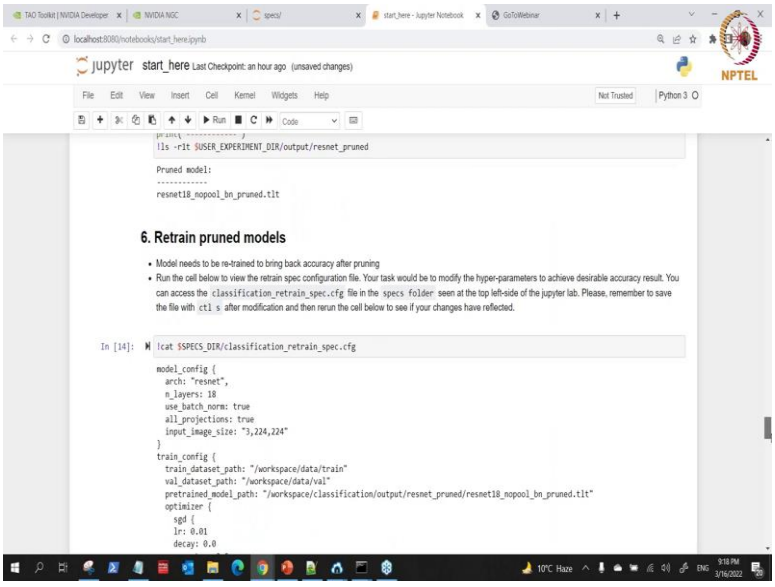
(Refer Slide Time: 15:50)



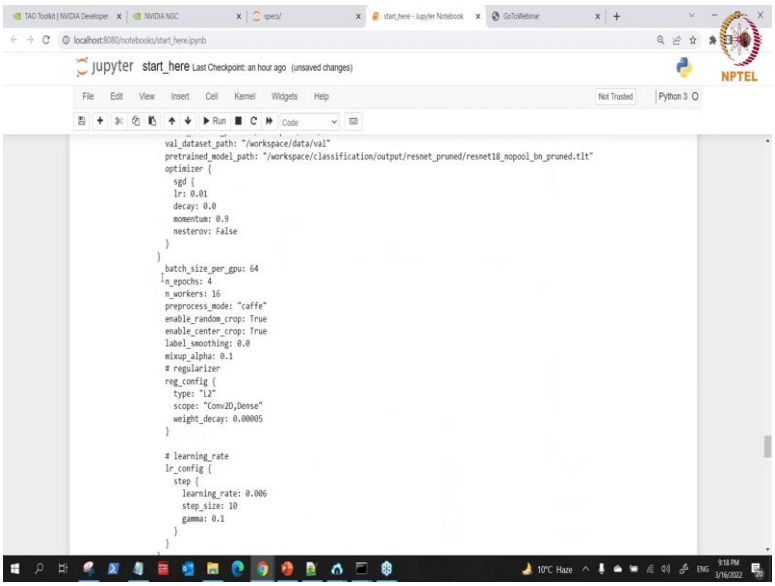
And then so, let us cut this and see what is inside. This is what is inside is the same thing as for the train, but the parts that are different is the retrain path and the evaluation path.

Also, here we cannot use 80 because we cannot wait for long. So, I would just I will come here again and look for, ok this is the retrain and change the number of epoch to change the number of epoch to 4. So, that we can just change with that and then the evaluation here, I would change that 1, ok as well. So, I can save, come to save, then close that, come back to here.

(Refer Slide Time: 16:05)

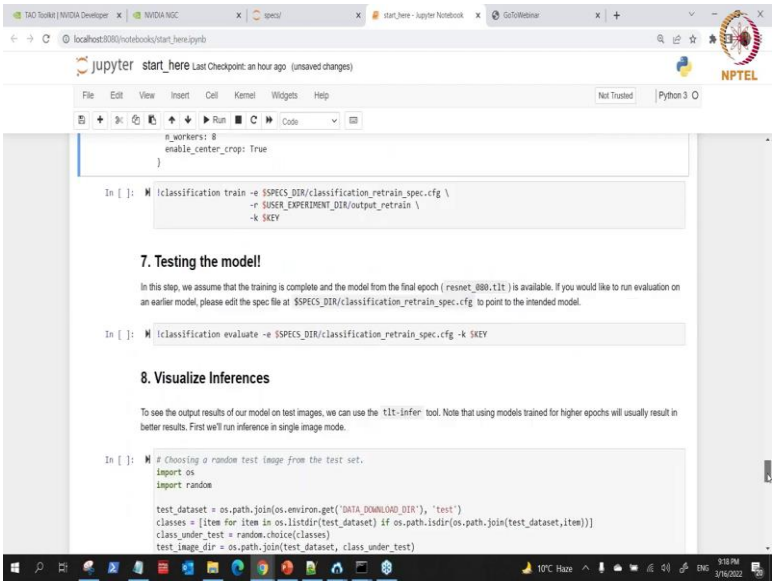


(Refer Slide Time: 16:08)



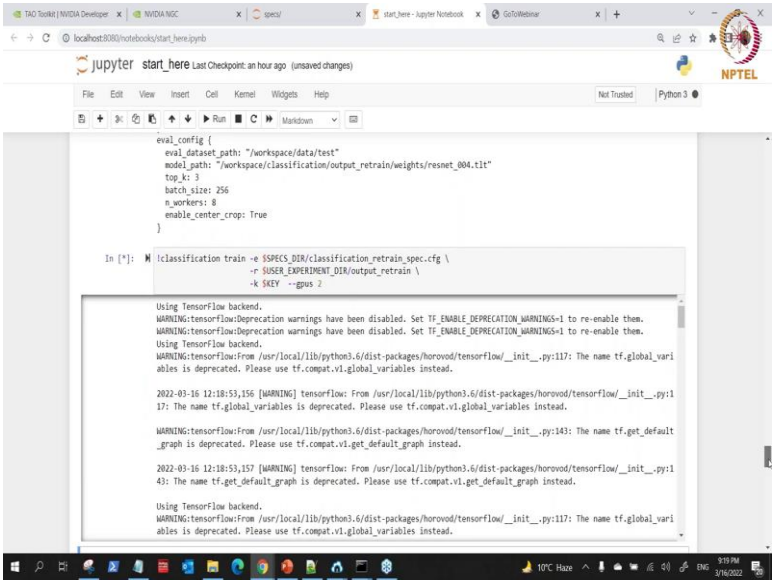
Then, I need to do is to let me check here whether that has been affected I think. So, yes it has been affected, here you can see, yeah.

(Refer Slide Time: 16:13)



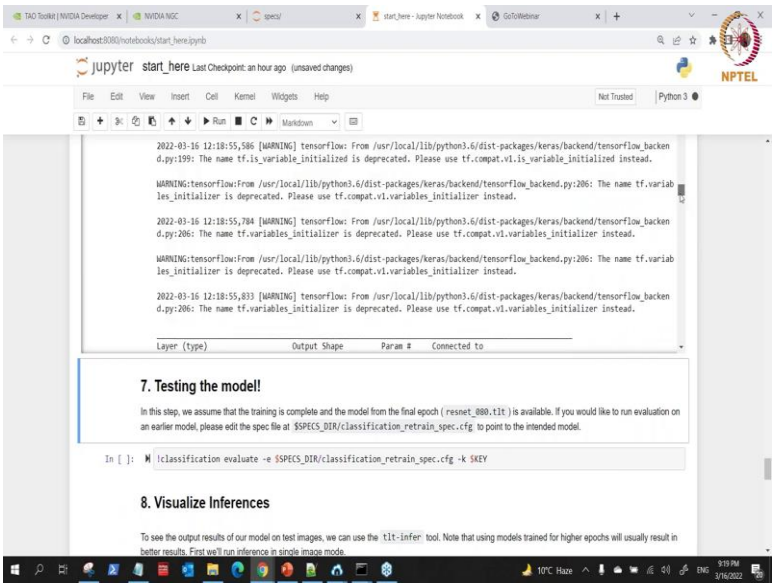
Then, we can proceed further here to retrain. So, to retrain here I would like to retrain using two GPU as well gpus 2. So, that can be faster than do that we are almost done with this. So, you can see this making things effortless.

(Refer Slide Time: 16:35)

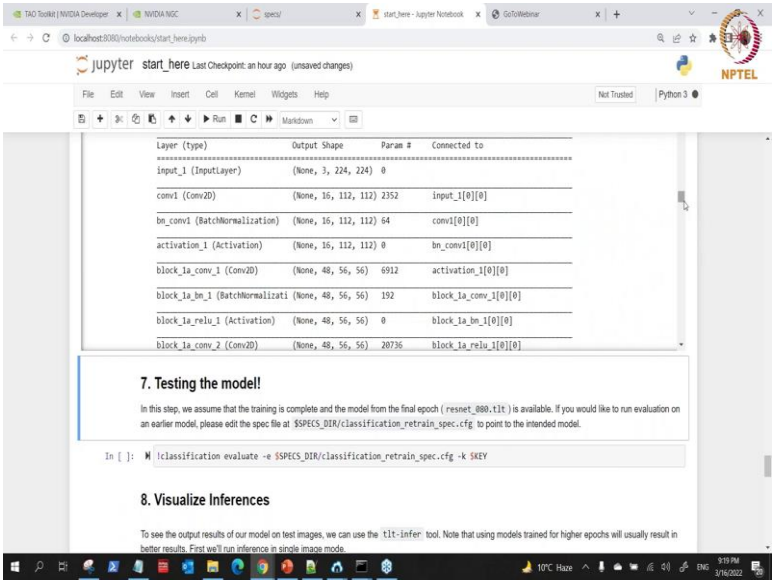


I must say because we are using multi GPU here, if you have to write it from the scratch it is going to be a lot of effort. Now, we are not writing any special code, yeah. And we are training our model.

(Refer Slide Time: 16:55)

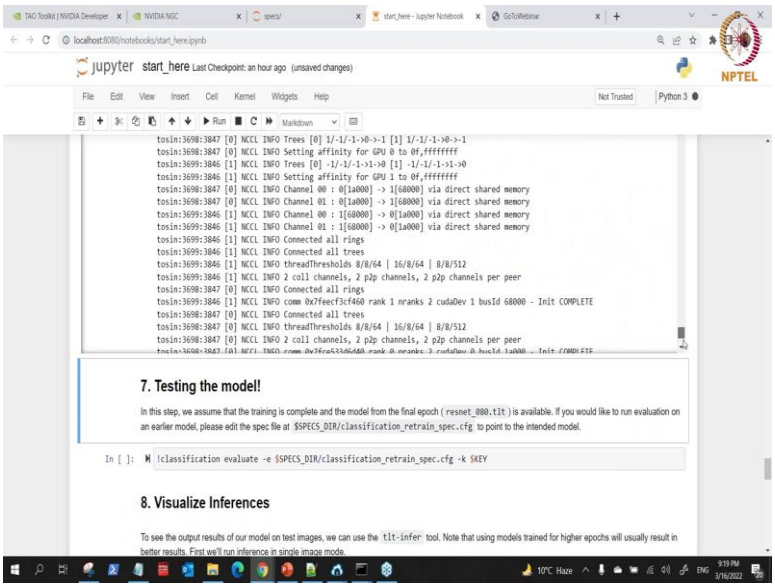


(Refer Slide Time: 16:54)



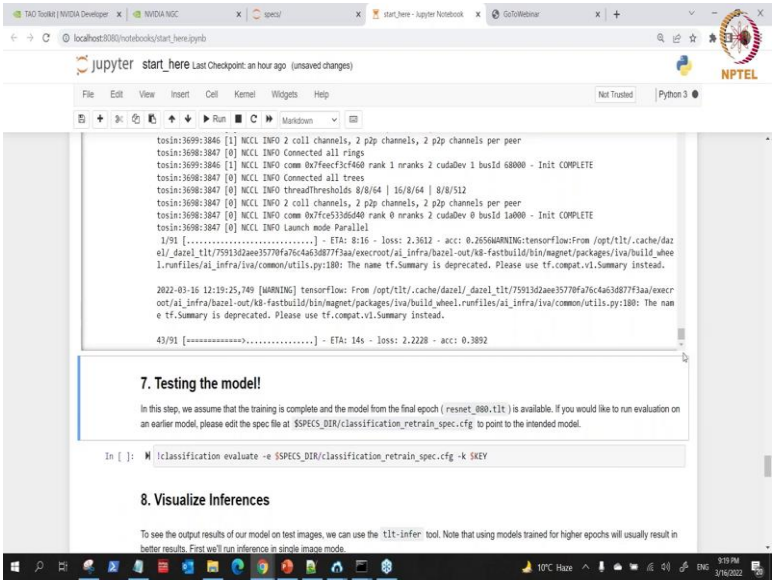
We are not writing any special code. What we only did was to just bring our raw data and do some configuration with the specification file. So, it is loading the model already there.

(Refer Slide Time: 17:05)

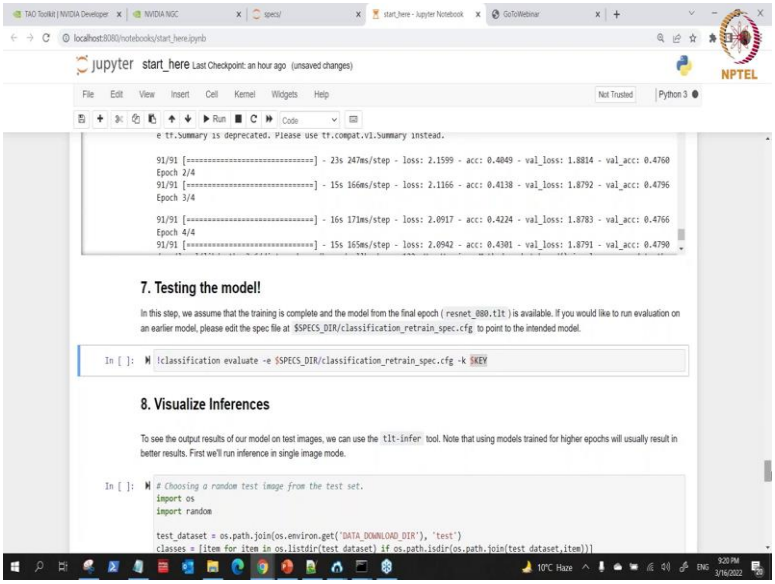


So, then it commence the training. So, you can see how it is been trained, yeah.

(Refer Slide Time: 17:11)



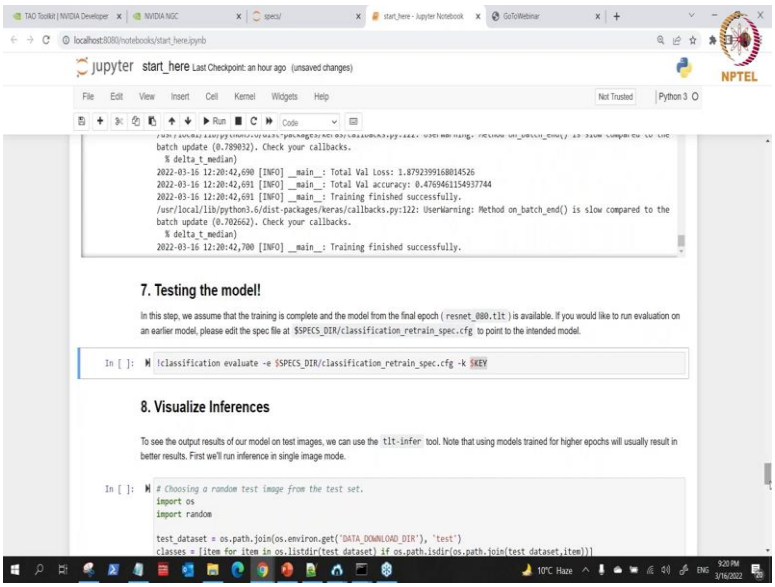
(Refer Slide Time: 17:14)



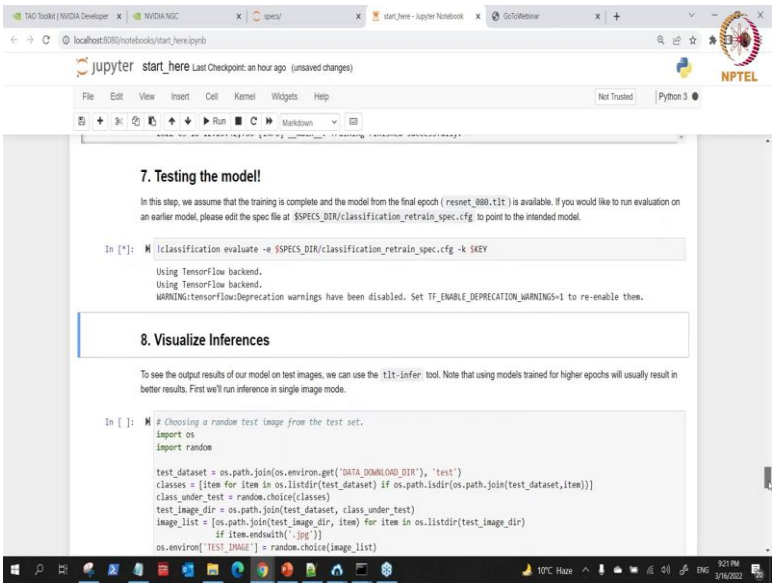
So, once this is done the next thing to do is to proceed to testing. So, to test is also evaluation. So, you have just the keyword is just classification, then evaluate. So, with what specification file? So, with this specification file, but retrain specification file then you supply the key as well while second epoch, so in less than 2 minutes I am sure be done with that.

So, if we once it finishes, then we test then we can now do our visualized inference here. We can visualize our, do our inferencing. And once we have done with the inferencing, then we will be able to export our model and that is done for.

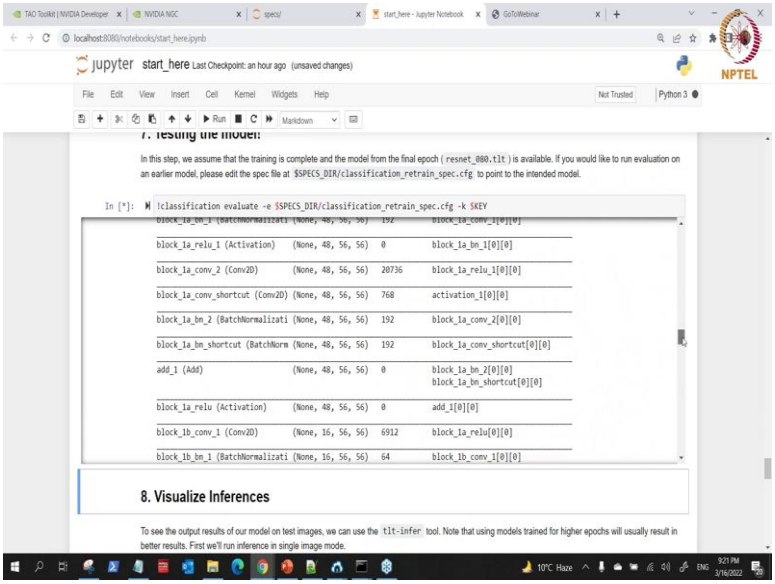
(Refer Slide Time: 18:24)



(Refer Slide Time: 18:38)

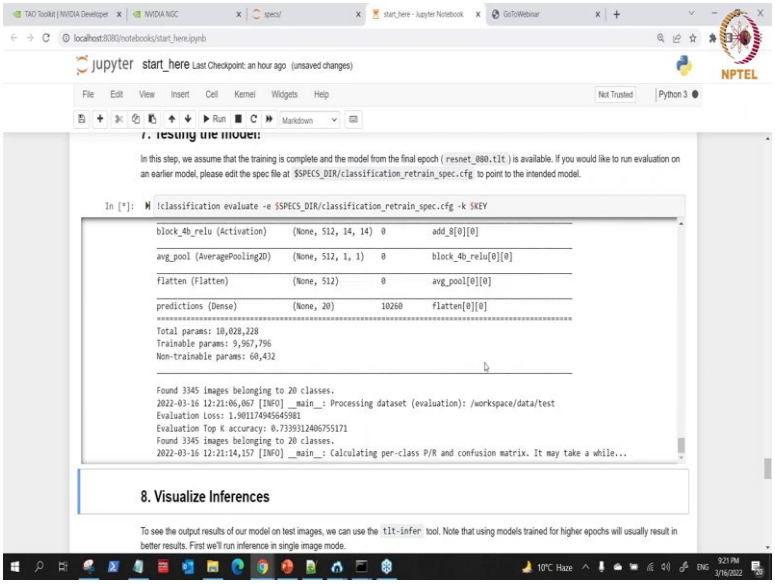


(Refer Slide Time: 18:47)

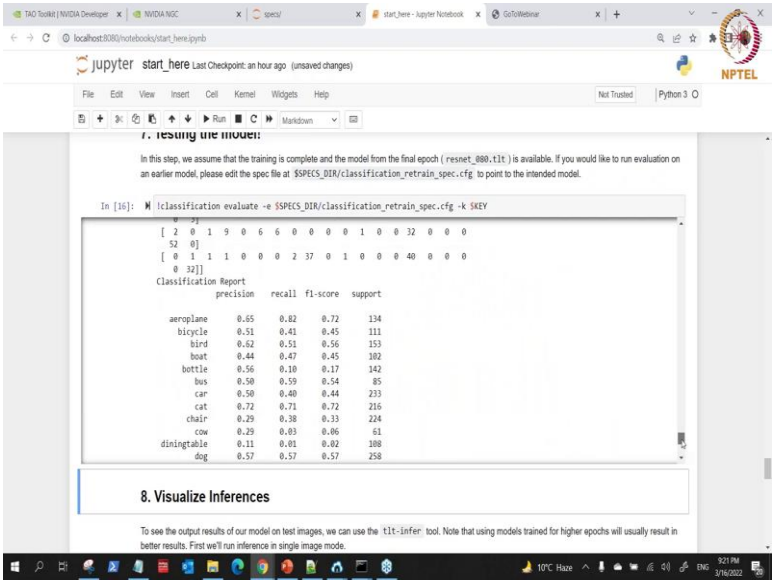


So, successfully retrain, that has been retrained successfully; you can see it that has given that then. What do we need to do? We need to test that. So, what we do is what we test that. So, that we also go into testing. It is just the what we do even when we write it manually ourselves. You know when you train a model you have to test it with some data sets. Since we have a test data sets, so it is good to test with that.

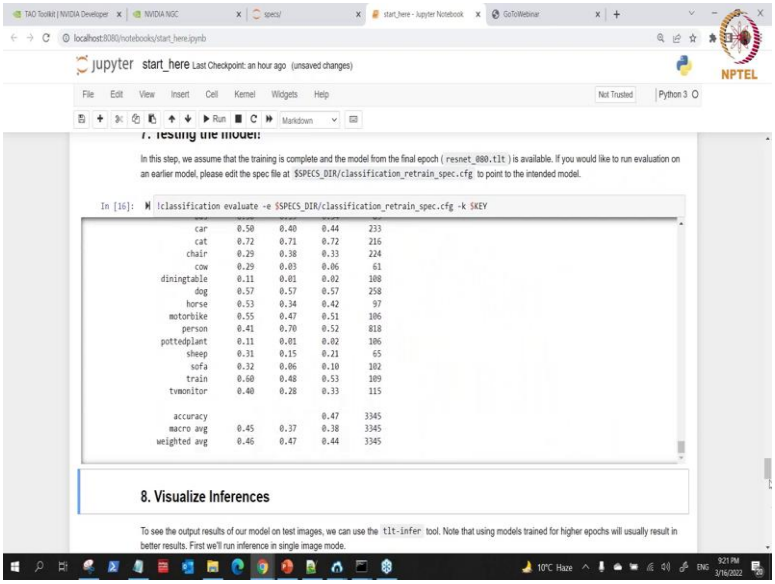
(Refer Slide Time: 18:57)



(Refer Slide Time: 19:13)

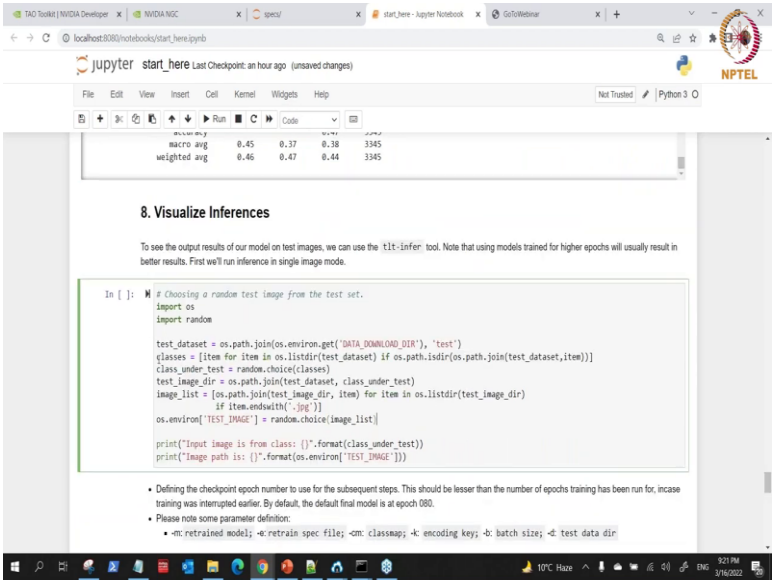


(Refer Slide Time: 19:15)



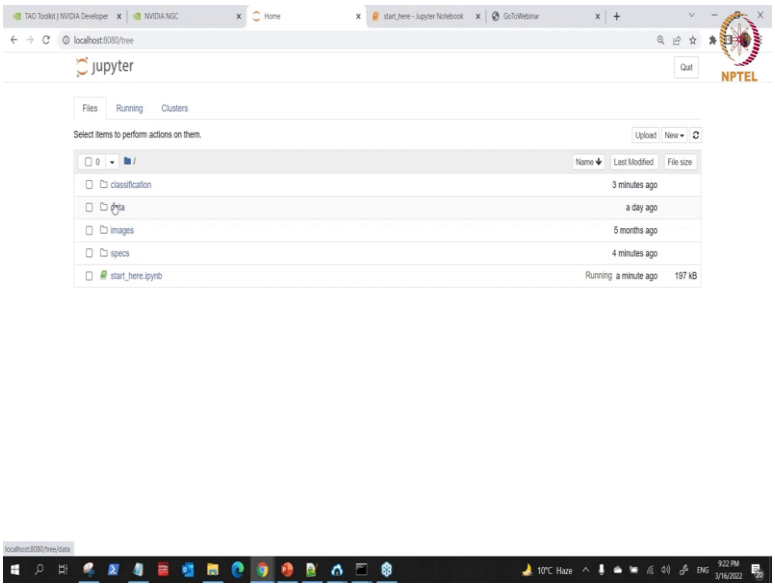
So, it is going to test and then once it the for the testing is just evaluation if we plot the confusion matrix and then gave us the classification report there.

(Refer Slide Time: 19:20)



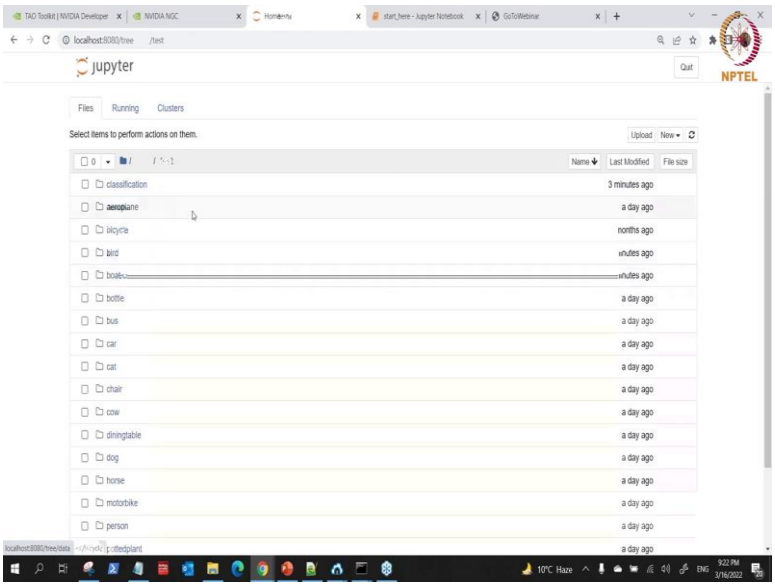
So, after that is done the next thing to do is what, ok let us start to inference. So, what we are trying to do here you can write this on your own, it is just that we want to select a particular image at random from the test folder.

(Refer Slide Time: 19:34)



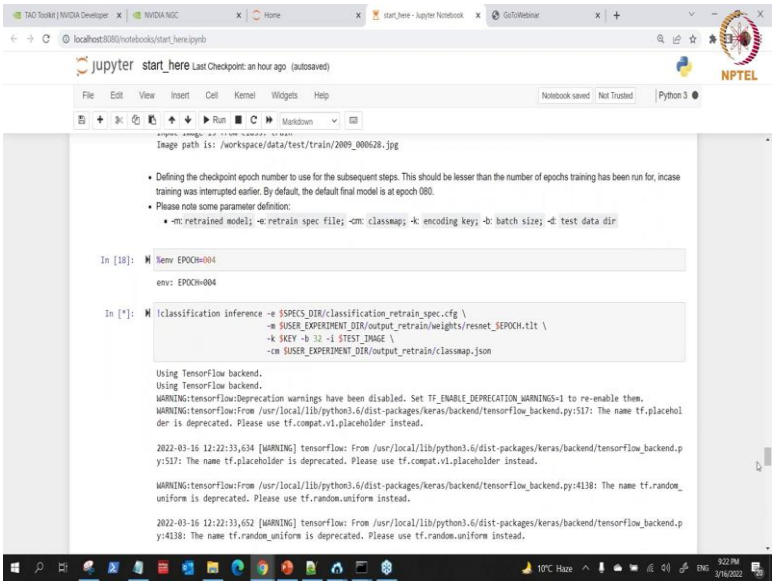
So, from the test folder here, let me come here, this is data and from the test folder here we just want to select the random, want to select a random data from there.

(Refer Slide Time: 19:36)



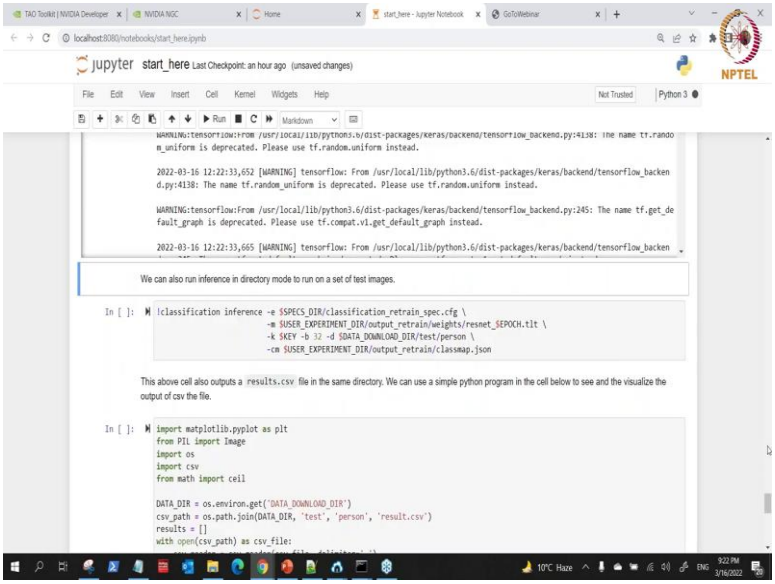
So, if I run this that is what he is doing there. So, you just speak the class, the class is trained and it has pick a particular image from there.

(Refer Slide Time: 19:44)

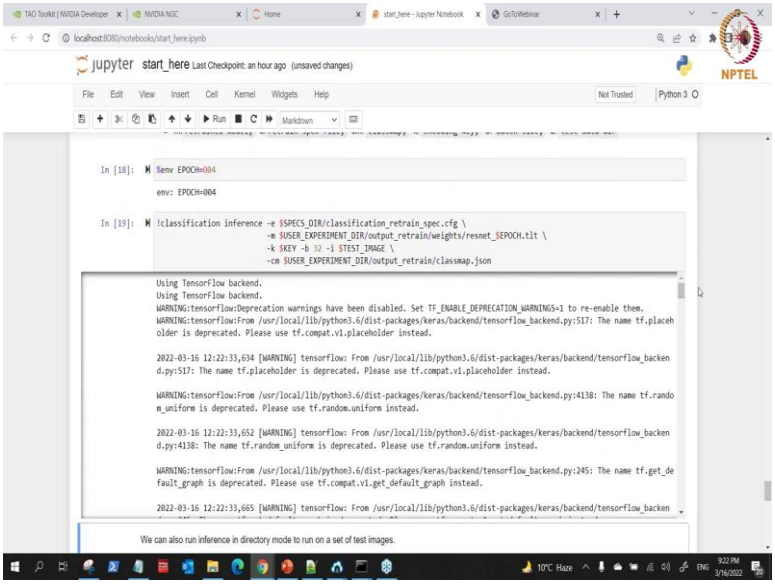


So, I am not expecting it to be able to identify these very because we trained with 4 epoch, mind you. But this is just a demo. It is just for you to know the flow. So, I do that, then here to run the inference here. So, even if the inference is not, the result is not correct with I am not bothered with that it say because we are just using it for demo here.

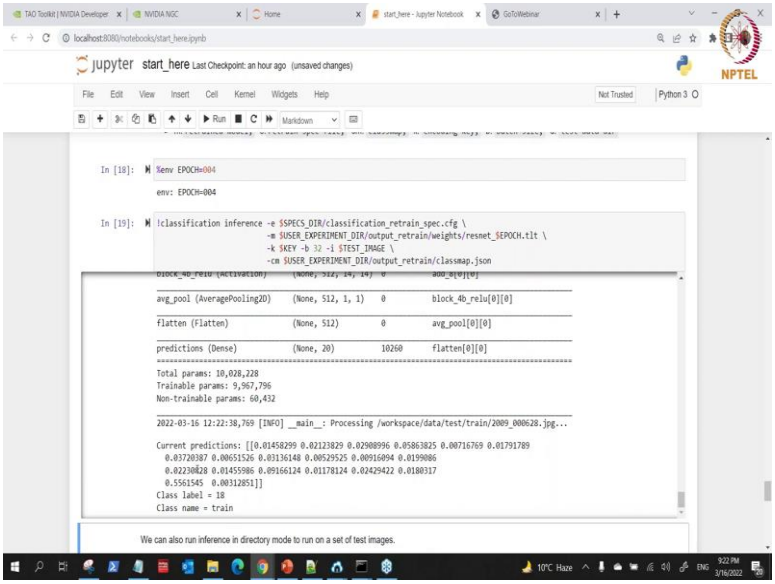
(Refer Slide Time: 20:17)



(Refer Slide Time: 20:22)

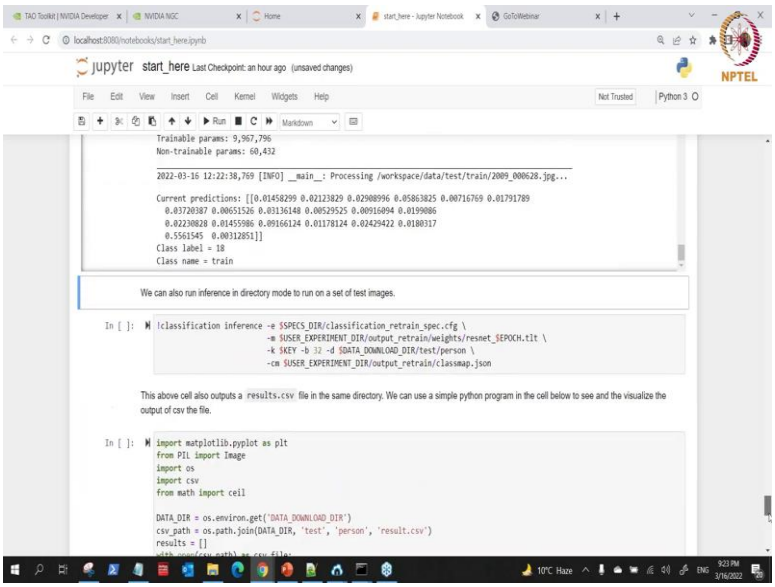


(Refer Slide Time: 20:28)



Then, we will able to see the reported get it loads the model, oh luckily he is able to get it as train. I think it has. So, what did we selected is a train, ok. The class is trained. This is the image. It was able to identify it also as trained, the label, ok. We are lucky there.

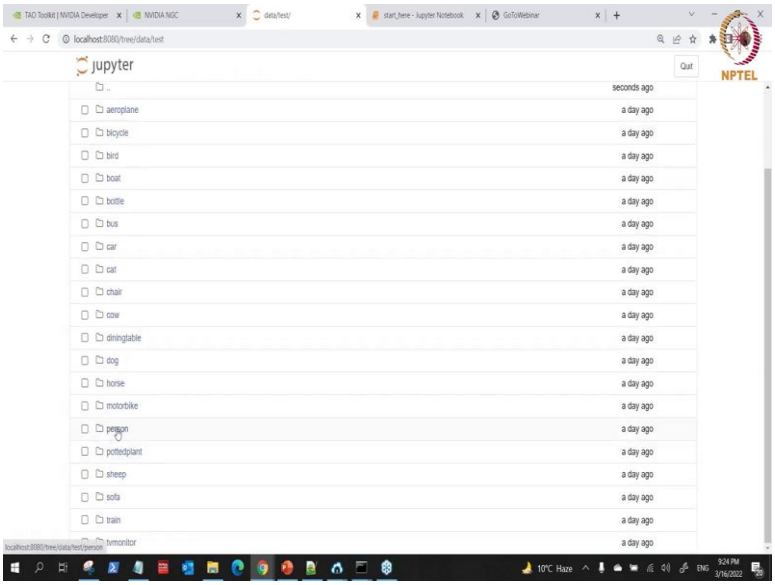
(Refer Slide Time: 20:44)



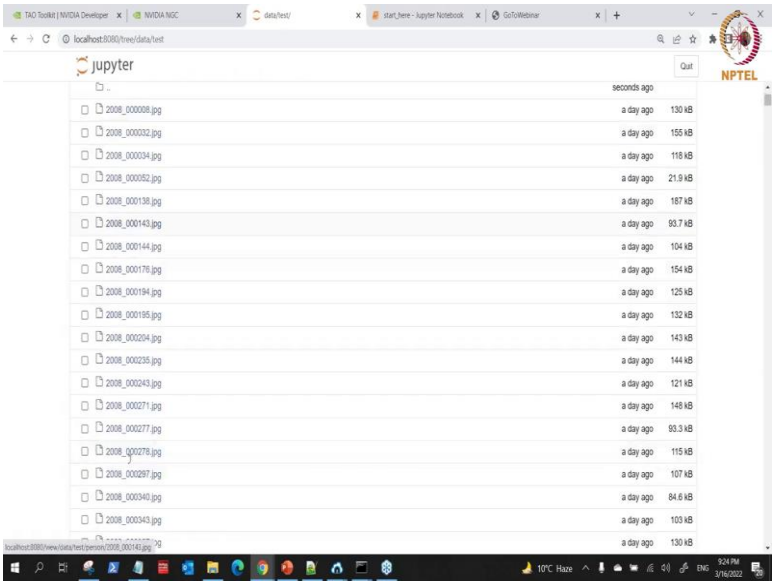
So, we are lucky there with 4 epoch is able to recognize that. Then also we can train we can also evaluate it inference on set of images. So, on set of images to inference that you can do you have classification and the keyword inference, then where which specification file to use retrain, where is the weights of the retrain model the paths, the

key here, the batch size start to. Then, data where do you load the class, the name, they have know which for that you want to use to evaluate or inference.

(Refer Slide Time: 21:42)

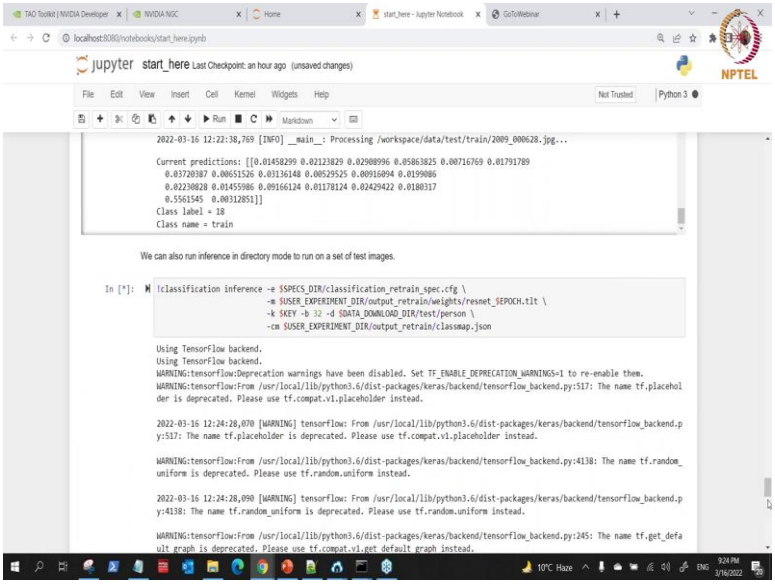


(Refer Slide Time: 21:44)



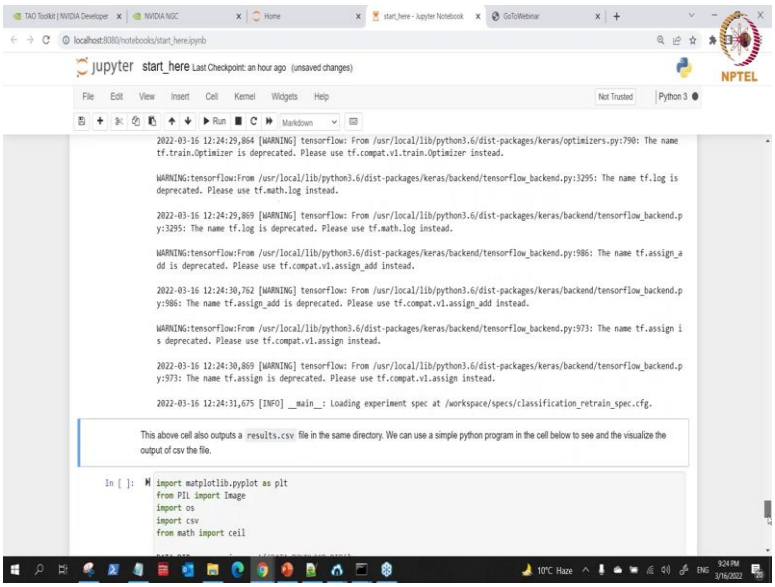
So, the folder the folder want to use the test 1 and we want to get a person from the test 1. So, from the test 1 here, you see if you come to data test, if we look for a person this is person. So, it will want to use it to test all these persons here. That is what we want to use it to want to test with that.

(Refer Slide Time: 21:58)

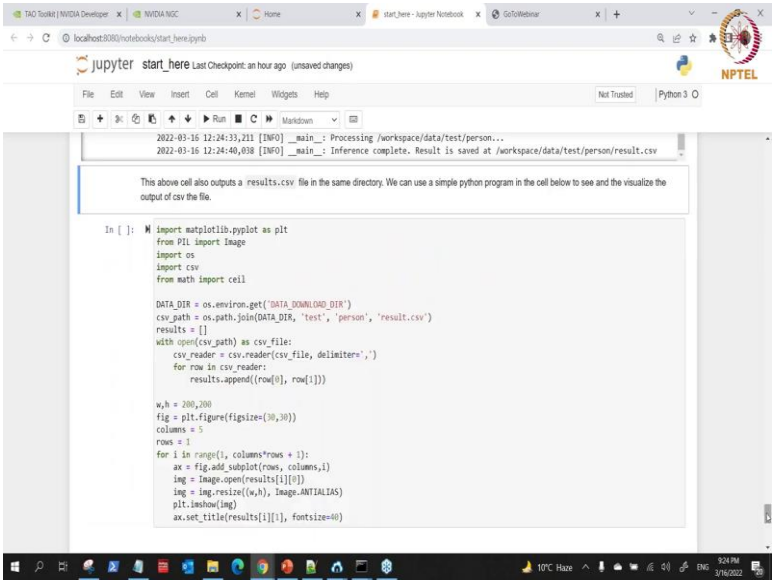


So, come back to data. So, that is what we are doing here. Then it is this one we seem we indicate the class name. So, it is we displayed graphics face and we can be able to see whether it is classifying well or not so, and that we go and then, ok. It is done.

(Refer Slide Time: 22:10)

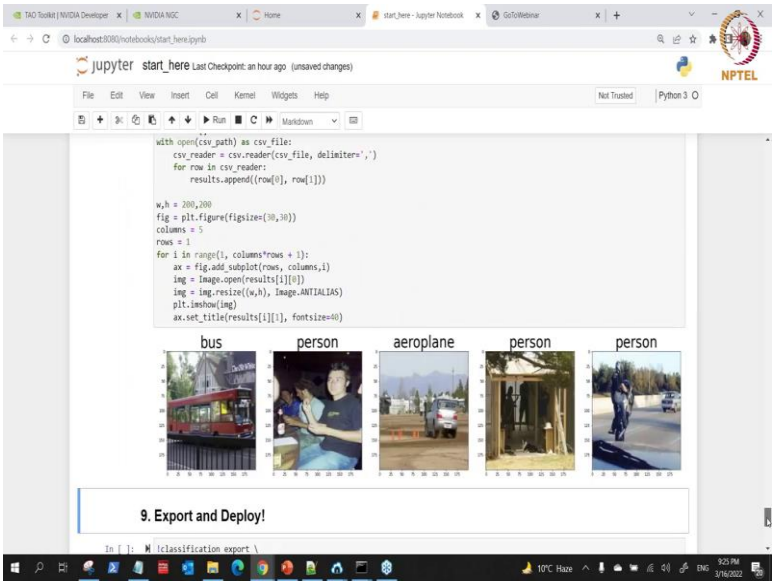


(Refer Slide Time: 22:24)



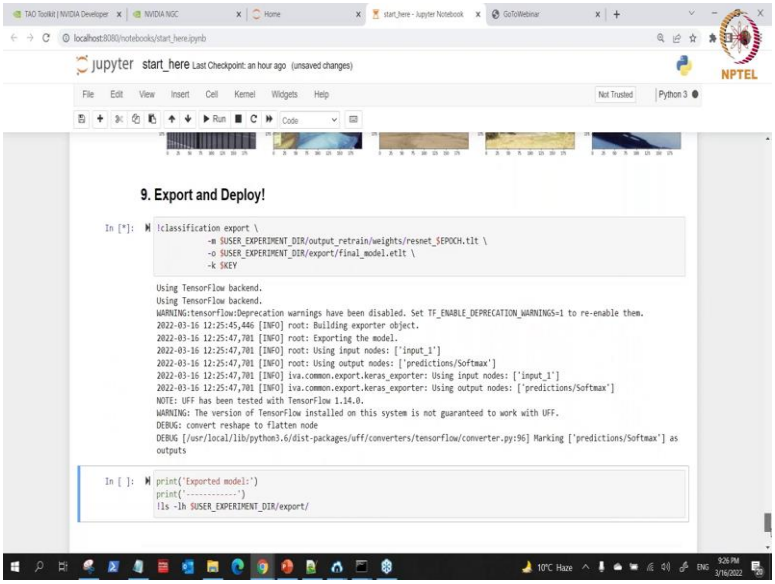
So, what we need to do is to what visualize the result. So, we can visualize the result with this, running this cell to visualize the result you can write yours, ok.

(Refer Slide Time: 22:33)



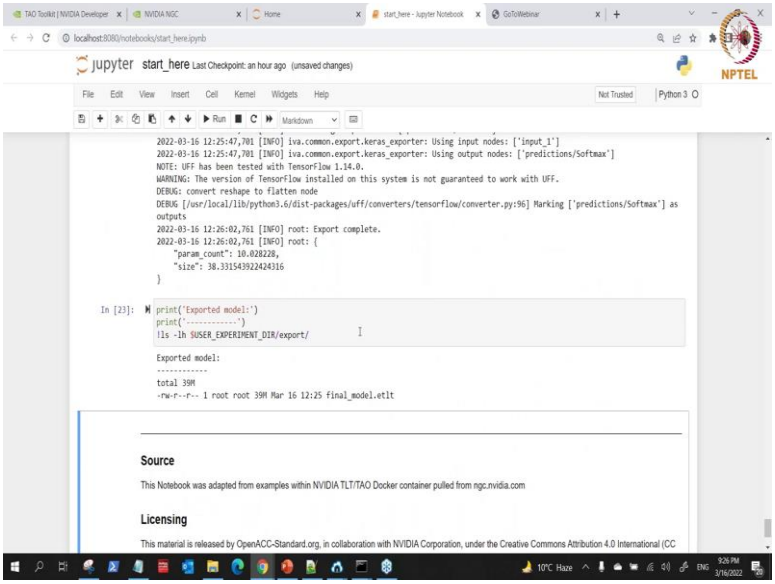
Let us see is able to classify that very well, ok. Get this is bus, this is person, this is not aeroplane, this is person and this is person because we train with 4 epoch. By time if we train with maybe 60, 80 epoch, everything will be the answer would be perfect there.

(Refer Slide Time: 22:55)



So, then, so the next thing to do is to what export our model. So, how do you export your model? You have the flag a which is classification then the keyword export. So, export which model where is the path of the model, this is where the model save the path. And where do you want to export it, which folder? This is the path of the folder here. There is a folder that I will call export here. Then, you need the key also to do that.

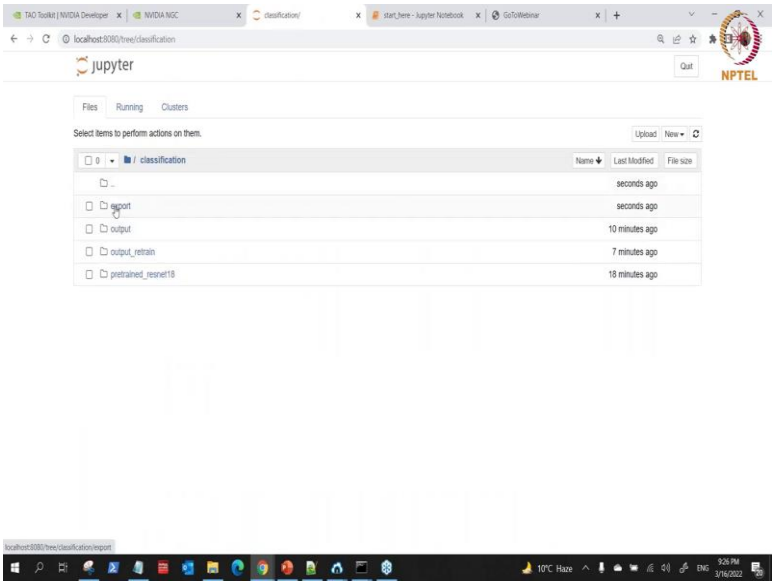
(Refer Slide Time: 23:40)



So, we just run these and then we will have I am using on how. So, this way run. Then, after this run we just validate that whether it is actually saved in the path we specify shall

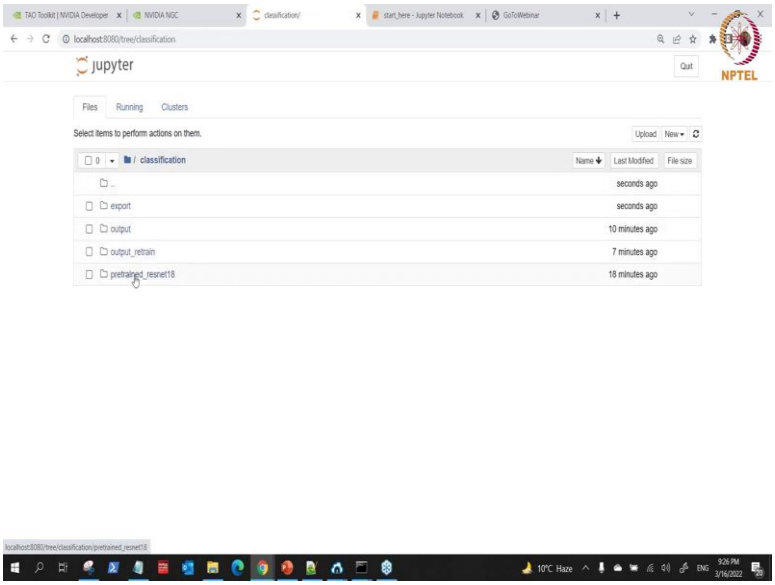
we do that, ok. I think that is done. Then let us check here we check that it is, ok. So, that is that about NVIDIA Tao tool kit to use it that, save it here.

(Refer Slide Time: 23:58)

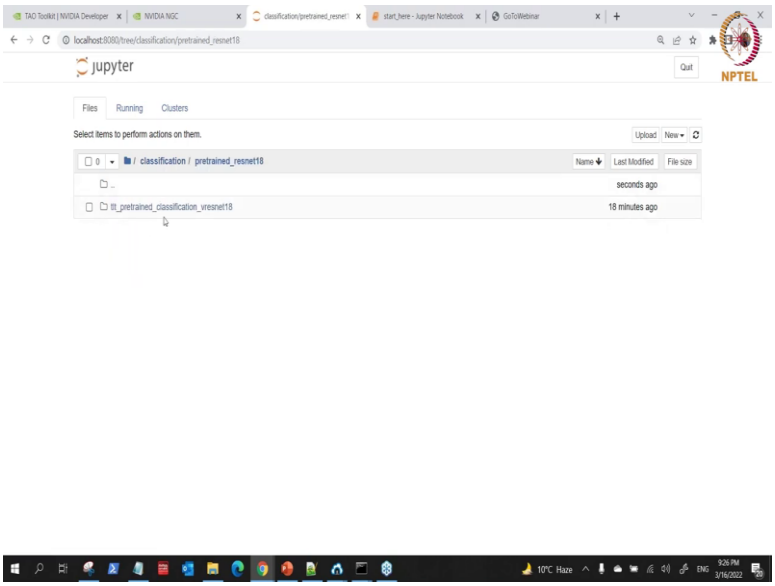


So, you will be able to check your folder, you see classification, you will be able to see export that this is the model there. I have been exported, this is the pretrained model, you can also see that there. So, all of that is done.

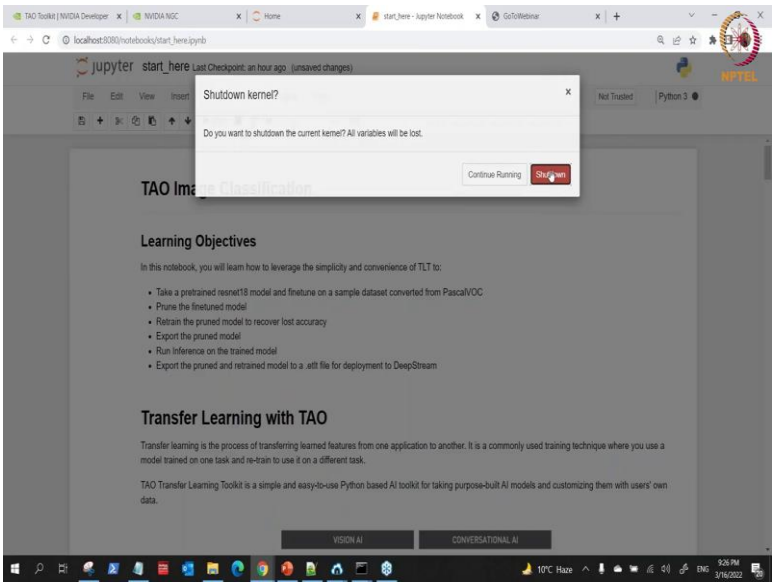
(Refer Slide Time: 24:03)



(Refer Slide Time: 24:04)

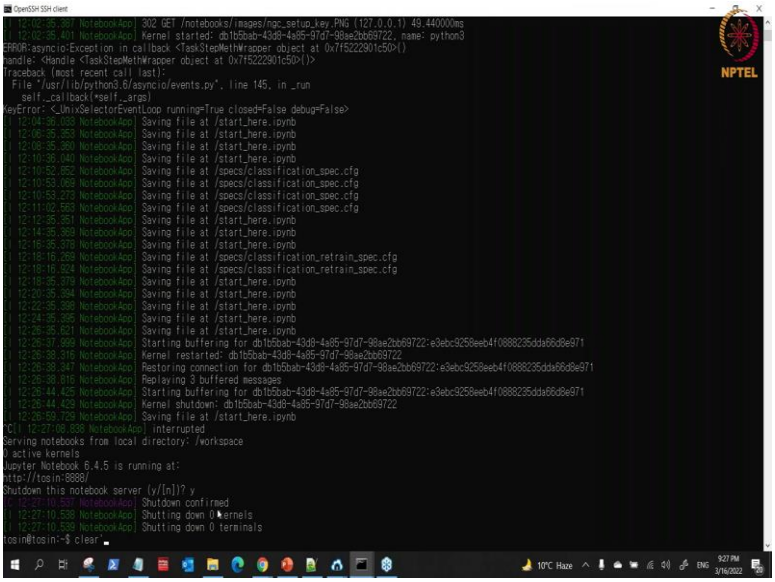


(Refer Slide Time: 24:23)



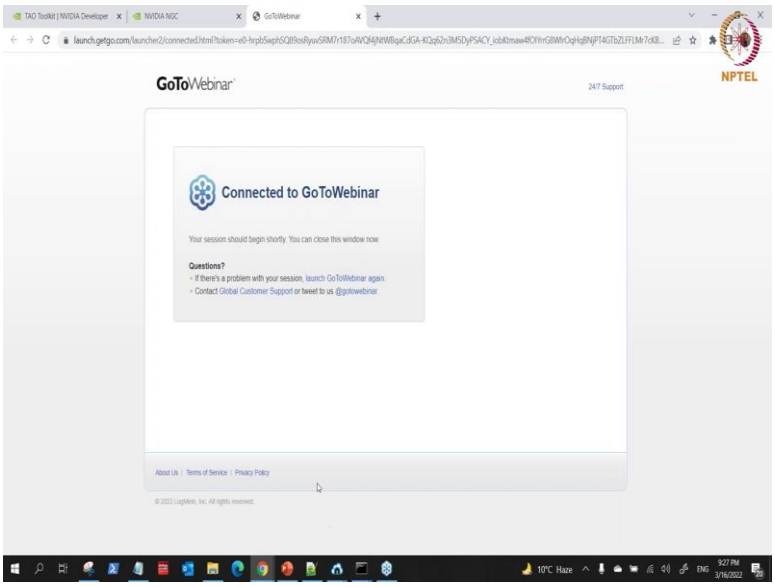
So, what I would do now is just to clear this, then turn down this kernel because we need to move folder. So, after shutting down this kernel, I can just close these, ok, alright, yes.

(Refer Slide Time: 24:44)

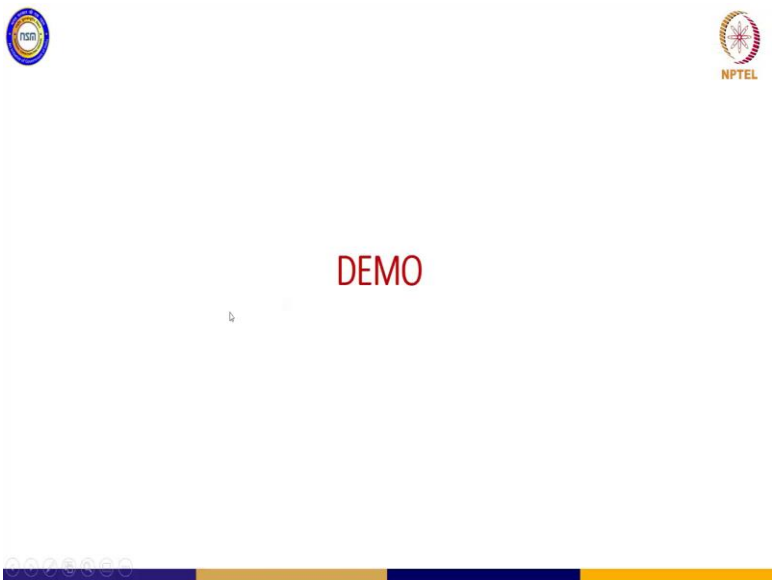


And then come here, I just I am stopping this, ok. Have clear, ok. And I can close this.

(Refer Slide Time: 25:04)



(Refer Slide Time: 25:11)



So, we will proceed. So, we have run that demo, then let us look at a little time we have if we can move to YOLO5.

(Refer Slide Time: 25:16)

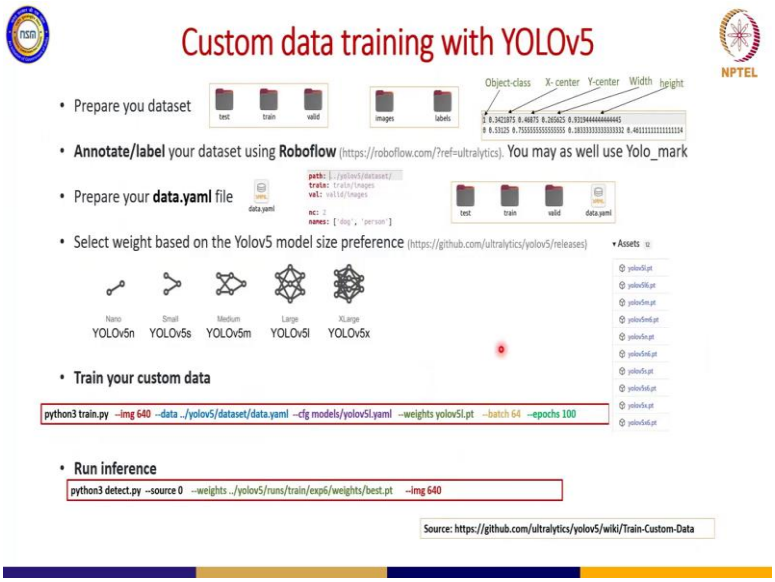
So, YOLO5 for computer vision, YOLO is a family of object detection. So, we use YOLO for object detection. It is highly based for computer vision. So and where do you get that? You can get that by visiting this website here. It is the original website. And then this is the GitHub repo. So, you go to GitHub, we clone, you can clone this repo.

So, now how do you do that? You need to clone following these steps. After you clone then you have you cd into the file the folder the directory, then you keep install some requirements. So, after you have done that, then you can now run your YOLO.

You can run because it can detect up to 80 objects. So, you just run python, detect.py and then source; source means the input where do you want to get your input from. If you want to supply input from a webcam, a single webcam you put it 0 here. So, it shows inputs will be coming and you will be seeing the computer vision the data will be what object detection running live.

If it is images you supply image here the path of the image rather than putting 0 here, you put the path of the path of the image, you put the path of the image there. If this video you can also what do detection on video. So, you put the path of the what video there then if you want to do object detection on set of images, you also put the path there. And you can do as well for YouTube and also for RTSP also. You can also run that there.

(Refer Slide Time: 27:03)



So, and now, so I have not done that. So, where does the transfer learning it comes in? It comes in when you want to use that particular model for your custom data. So, they call it custom data training with YOLO5. So, you bring your own data, not the data in the YOLO. So, YOLO5 was trained using coco dataset.

Now, you have your own data. So, how do you prepare your data set? You have your data in this format here. So, test, train, and validation in this set here. So, in each of them you will have folder in each of them which is called images. This is where your image will be. And you will have another one label, although this label will be empty at first.

So, the next thing to do is to what? Is to label these images so, how do you label or you annotate these images? So, you go to what we call the roboflow. This is the website here. So, in that website you upload this your data there and you do labelling and after then if you generate this label file for you. Or, you can also use what we call the YOLO mark, although you may need to fine tune some other things using YOLO mark because YOLO mark is specifically dedicated for YOLOv4.

However, the only thing you need to do is just the way it arranged the classes. So, if we generate the label. So, what is inside the label? This is what is inside the label, yeah. It has 5 features. Now, the first one is to give it the object class. So, if you have if you are classifying 5 objects. So, it would be, you would give them value of 0, 1, 2, 3, 4.

So, this is an object class. The second one is the X-center of the object in the image and this is the Y-center of that object, and the third one the third one is the Y-center. The fourth one is the width of the image, and the last one is the height of the image. So, if you see the second one here is 0, this is another class.

Then, the next thing to do is to prepare your data.yaml file. So, the data.yaml file is what you have here. So, you have to what you add these paths. So, based on your folder, the folder the YOLO folder you clone, YOLOv5, then you include data set path. This path is to this data. Then, your train, you have your train and validation. So, it will be train/images train/validation.

Once you use Robo, Robo will help you to generate these. So, the nc here means what? The number of classes. Let us say we are considering two classes and the name of the classes is just this. So, you have two classes, dog and person which is two classes. So, after you have done that. So, your data set will look this way. You have test, you have trained, you have validation, you have your data.yaml there.

So, the next thing to do is to select the weights you want for YOLO5 because YOLO5 has set of weights, different set of weight based on the model. So, you can have this is

for YOLO, if you want to consider nano YOLO, if you want to consider small YOLO, you want to consider medium size, large size or extra large size are there. And these are the data set, these are the weights that you can download. This how they look like.

So, after you have done that then you can train your custom model. So, how do you train your custom model you can train them using this python 3 because I am running on Bluetooth, that is why we have python 3 train.py and this img is talking about the dimension of your image. Then, data, that is where is the path to this data yaml. You specify that.

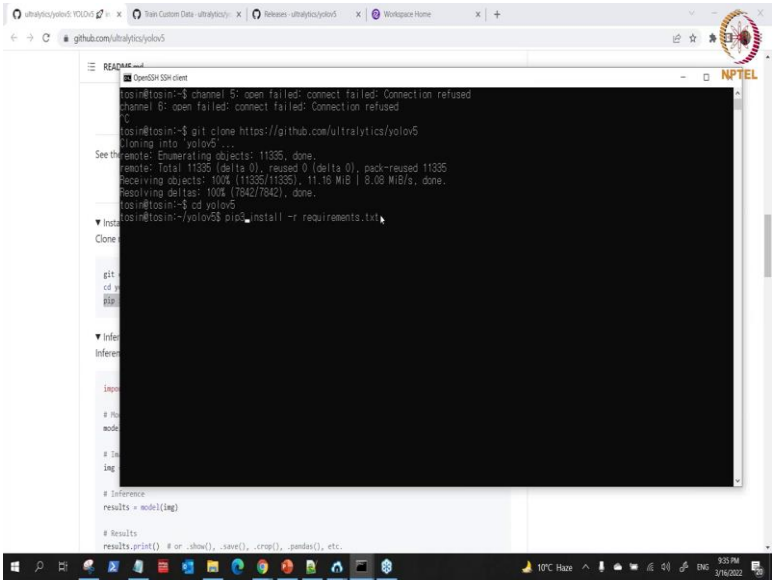
And then there is the word the configuration file for YOLO5. This is the path you specify it. Then, the weight you want to use based on any of these so if we select large, so the weight will be YOLOv5l.pt that is the weights. And then the batch size and then the number of epoch you want to.

So, after you have done that then you can also run; if after you have trained successfully you can run your inference using this python3 detect.py. Then your source, if this is an example for using camera webcam camera, so when I say source, its 0. So, that is webcam camera.

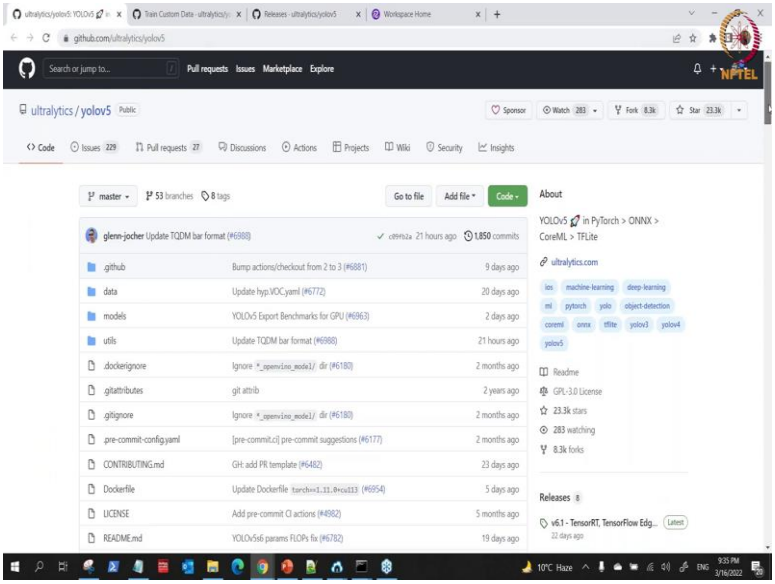
Then, the weight. So, the weight will not be the weight that is here again because this one is YOLOv5 weight. Now, you have trained your model your model the weight will be saved in this paths. So, it will save it as dot best. So, you specify that weight and the image size and then you are good to go.

So, all of that you can see, you can be for custom image, you will be able to access them in this with this link. So, I will give a demo. But unfortunately, we may not be able to use webcam because I am doing necessary I have been trying it, but it is not working with this size.

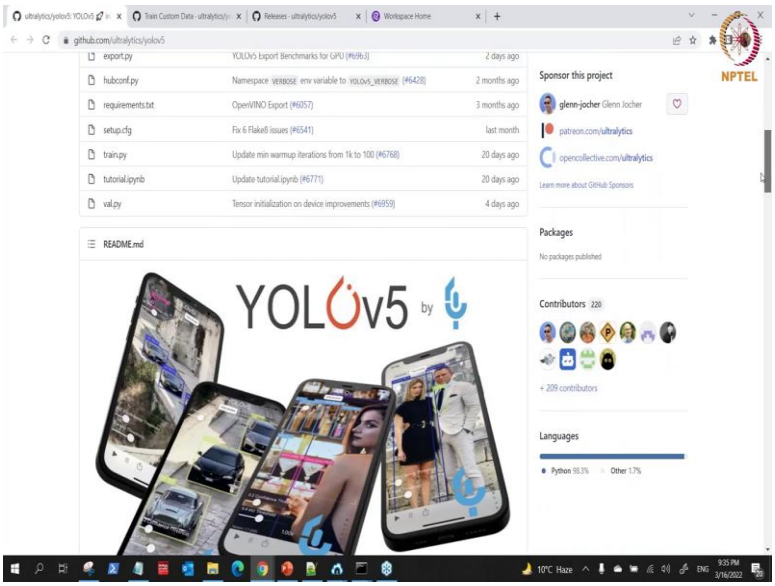
(Refer Slide Time: 32:21)



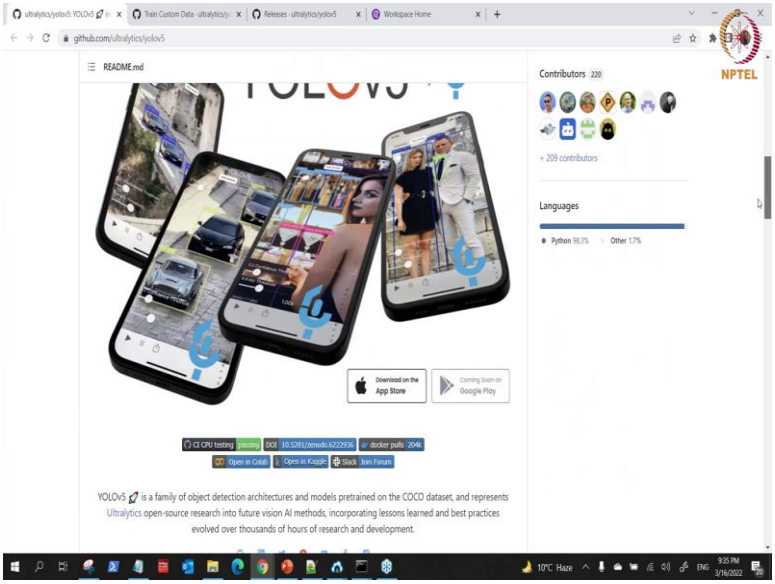
(Refer Slide Time: 32:36)



(Refer Slide Time: 32:43)

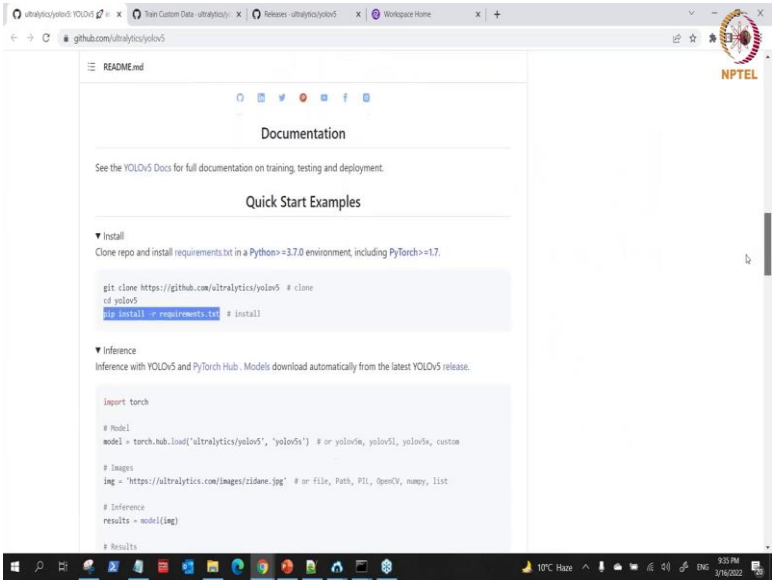


(Refer Slide Time: 32:44)



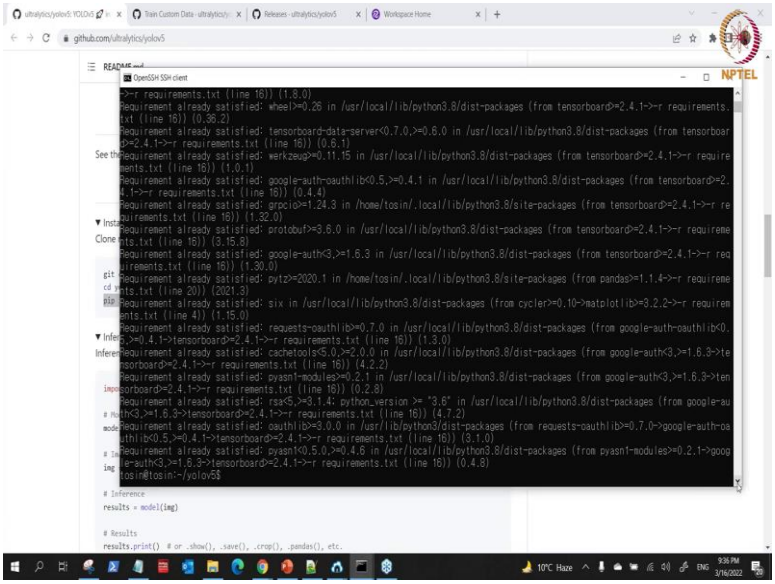
So, what we need to do now is to come here, ok. So, first thing to do is to download. So, let us go here, ok. So, this is the site for the GitHub for YOLOv5. So, you can see it for YOLOv5. They are requirement there the steps to follow.

(Refer Slide Time: 32:45)



So, we can come here and say, ok let us copy this. It is clone you just copy, it clone and you are here. So, I just cannot put us here. So, I just it will clone, hope there is more because I have run this before. So, let us see then cd yolov5, then. So, the next thing to do is to what pip install the requirements. I hope there will be no conflict because I have done this for; so, because I am on Ubuntu my pip will be pip 3.

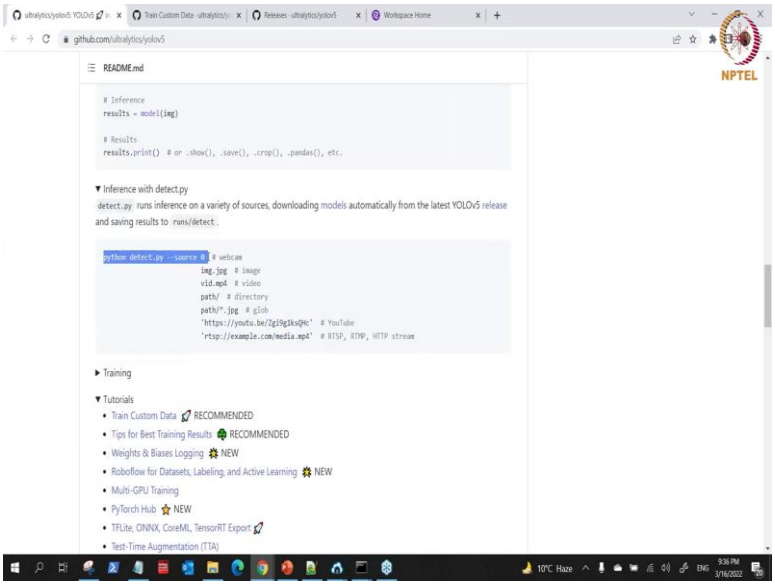
(Refer Slide Time: 33:24)



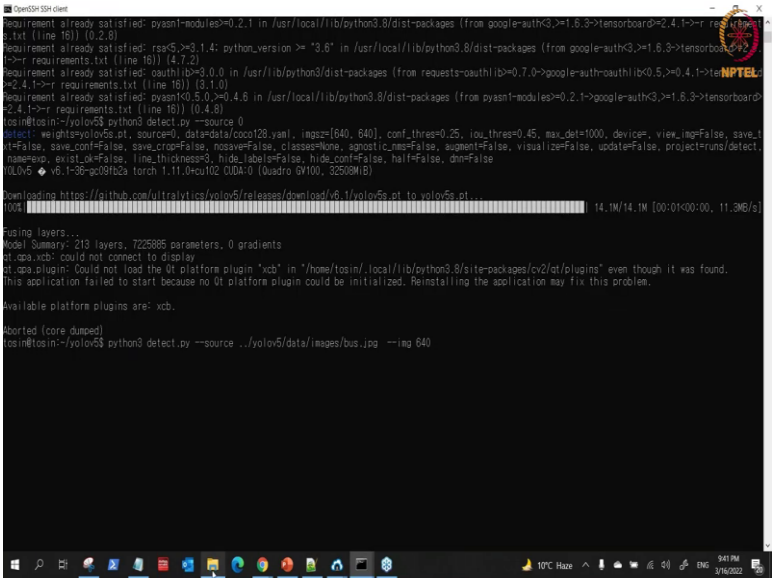
So, it we do that, ok since it has been stopped before with that. So, now, YOLOv5 is ready, not the custom one, but YOLOv5 is ready now. So, how do we run YOLOv5

now? So, if I want to use, I want to use what we call the webcam that will be an issue, but I will show you.

(Refer Slide Time: 33:54)

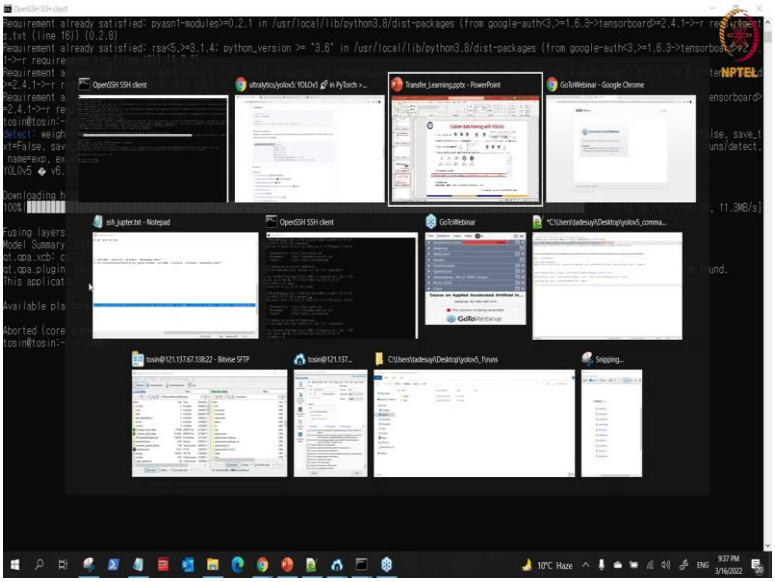


(Refer Slide Time: 34:02)

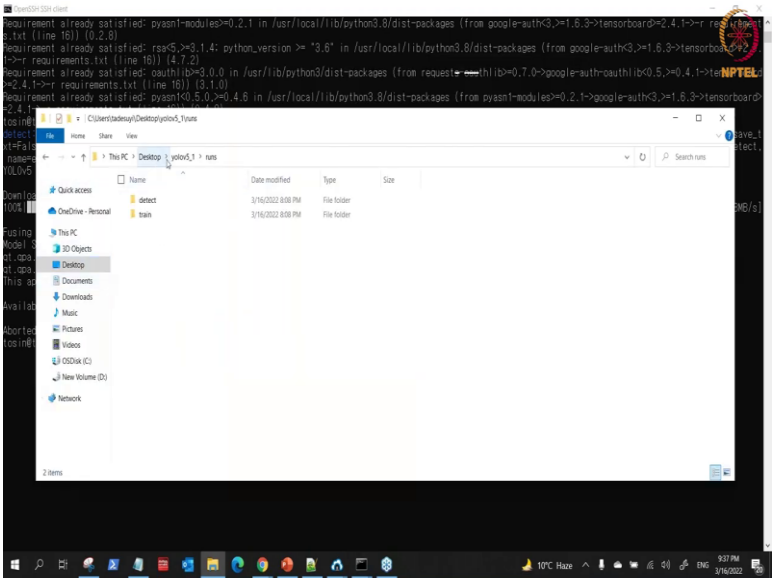


So, if I say python detect. So, this there will be it will complain of something, so it will be python 3 rather, ok. So, python 3 detect. I hope you can see my screen. So, that is, so this is python 3 detect, so, ok. So, it is running now. So, it is downloading the weights. So, it aborted because ideally there is we cannot use webcam.

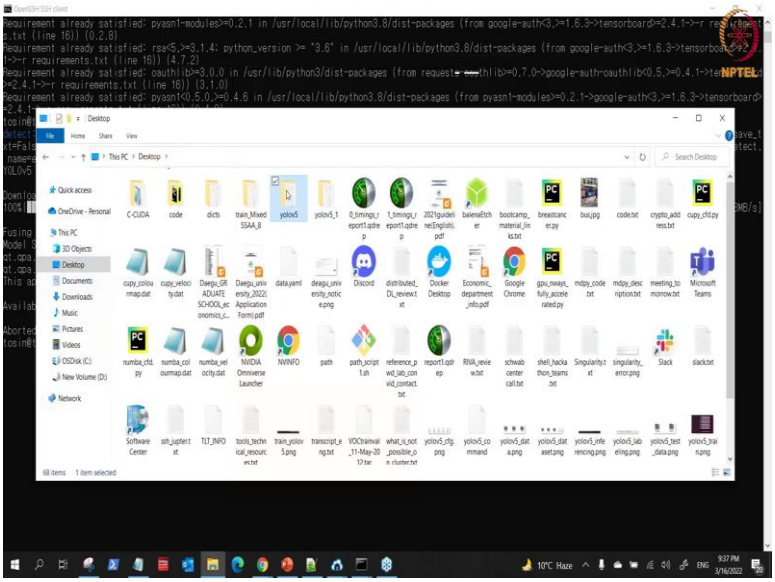
(Refer Slide Time: 34:45)



(Refer Slide Time: 34:52)

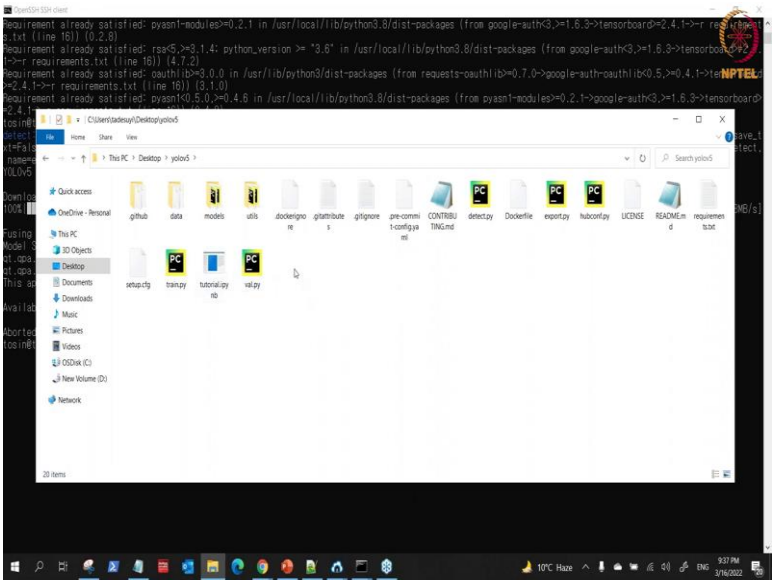


(Refer Slide Time: 34:56)



So, what we need to do is let me just show you let me go to my folder here. So, once you download your YOLO, this is YOLO. So, once you download it this is how the folder looks like.

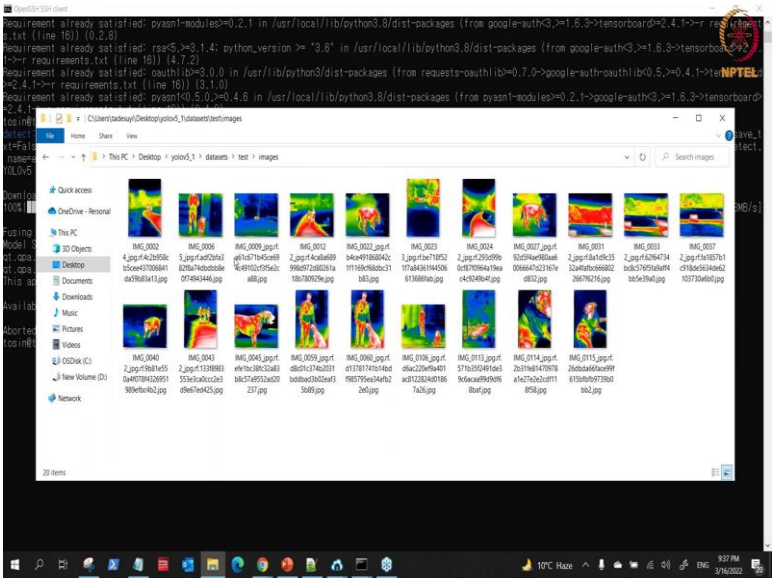
(Refer Slide Time: 34:58)



So, the folder looks like this, then what we need to do? In order to run that is if you go to, you need to include a new for your a folder for your dataset. So, the one I have done before is that I included the new folder, this is the one I have done before. So, I included this, this is the data set.

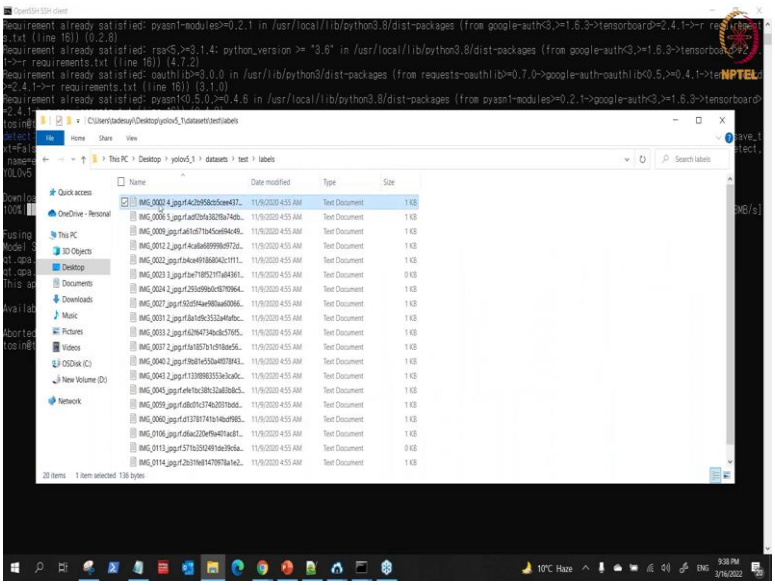
So, what happened here? This is the test, ok. Let me show you this is the test. So, you have its image and what; label here.

(Refer Slide Time: 35:28)

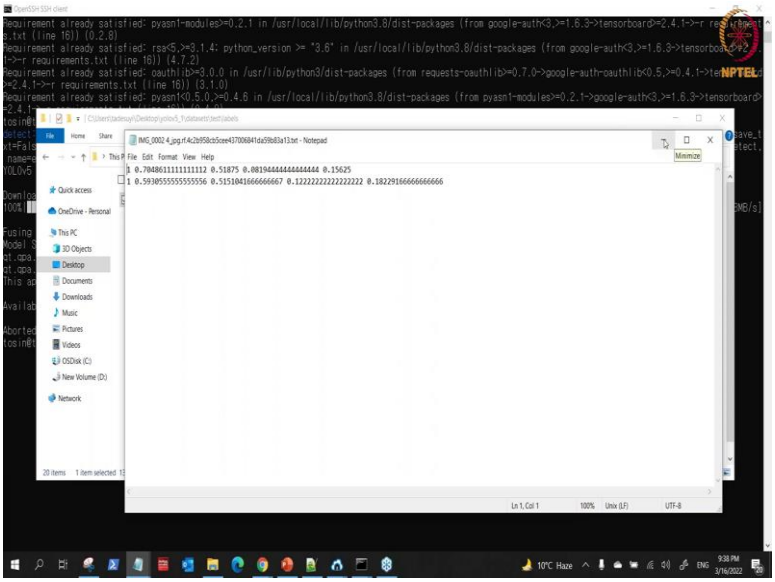


So, this is the image. I am just using this. This image is for image for dog and person.

(Refer Slide Time: 35:40)

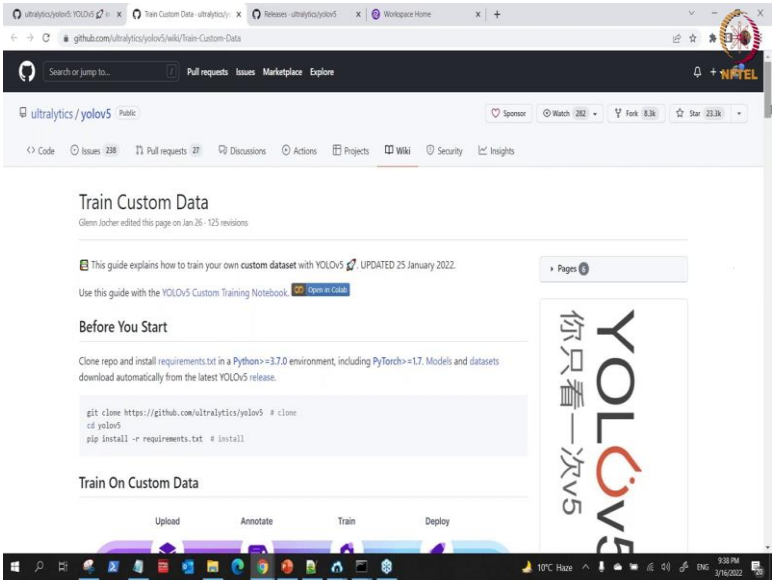


(Refer Slide Time: 35:44)

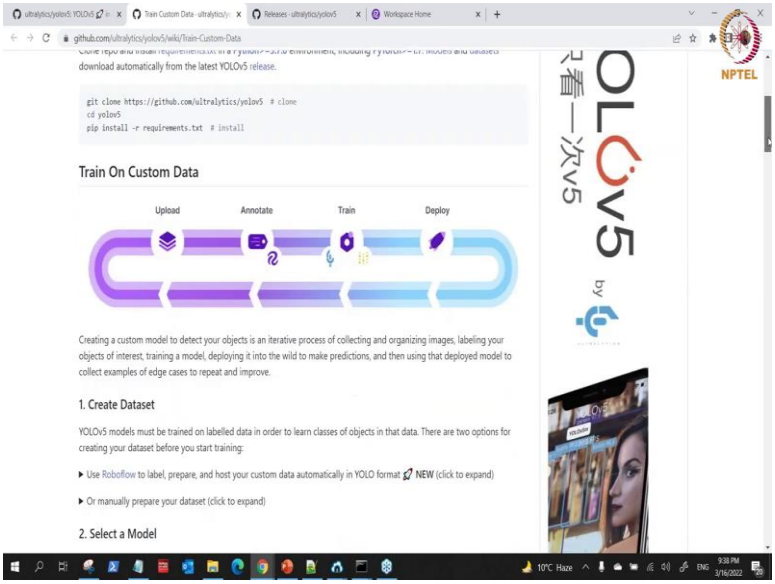


So, then the label is in this format. So, if you look at the label here, this is the format here. So, in this class which I have shown you before. But how do you get this format? So, for you to get this format what you need to do is to, what you need to do is to go to come here please; I want to waste the time. What you need to do is to go to Robo, here there is Robo here.

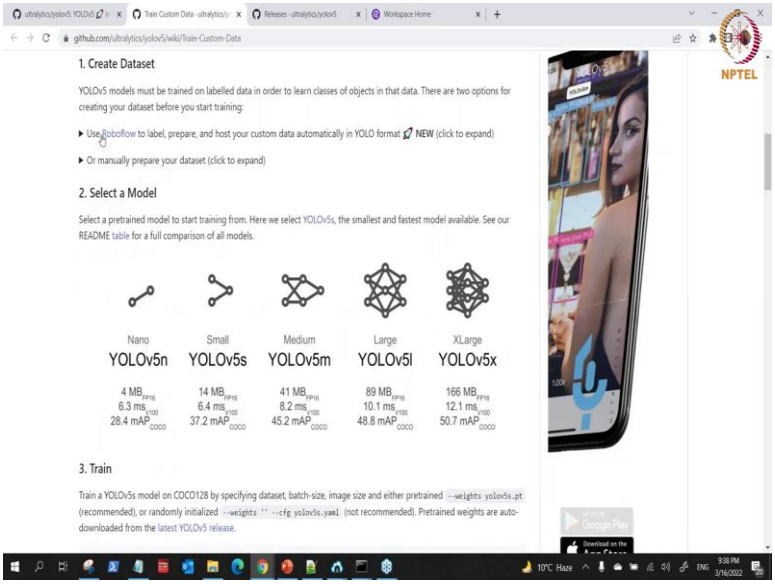
(Refer Slide Time: 36:12)



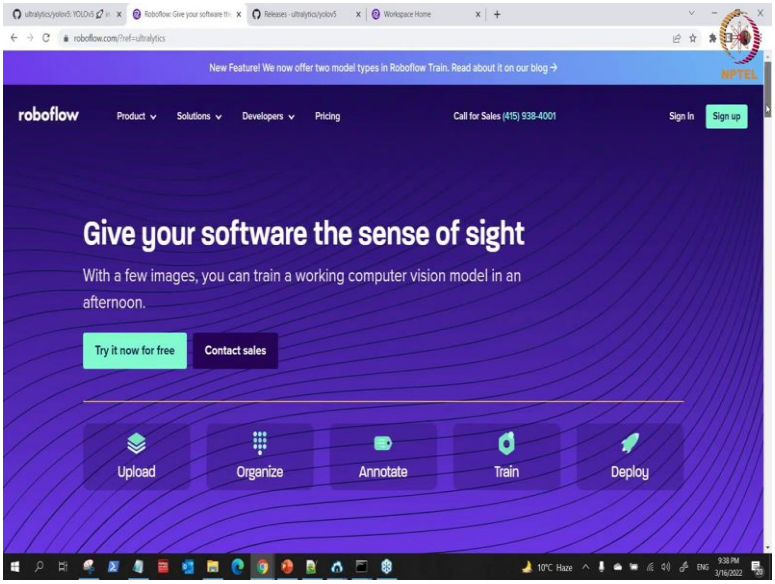
(Refer Slide Time: 36:16)



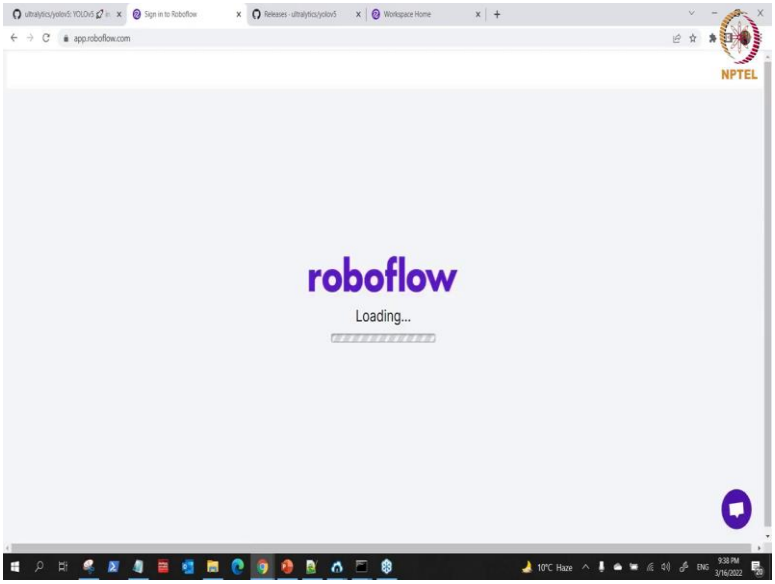
(Refer Slide Time: 36:17)



(Refer Slide Time: 36:21)

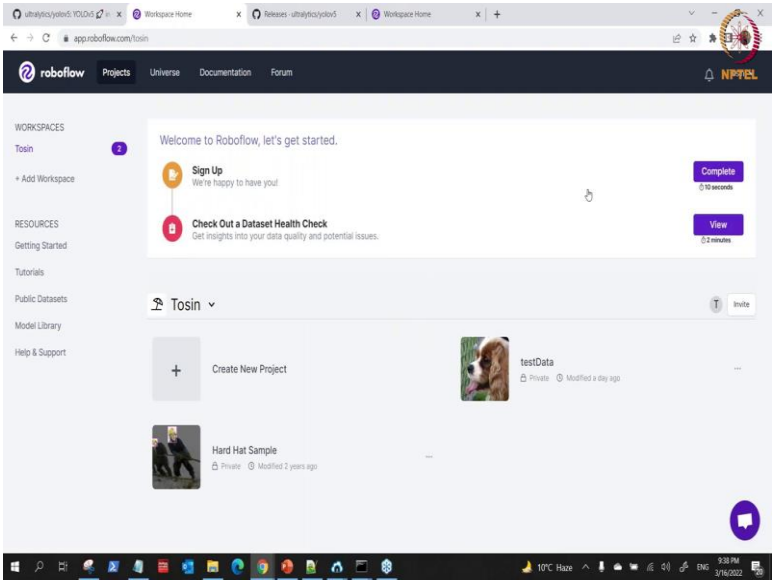


(Refer Slide Time: 36:26)

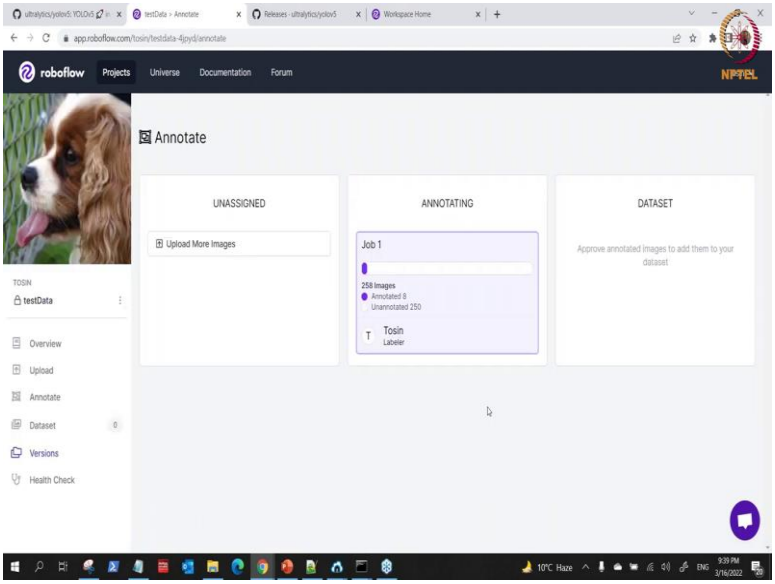


So, this is custom data training, if you want to do custom data training. So, when you get to custom data training then you go to Robo, this is Robo. Then, you need to sign in. I do not know if it allow me to sign in because I have been signing in before, yeah.

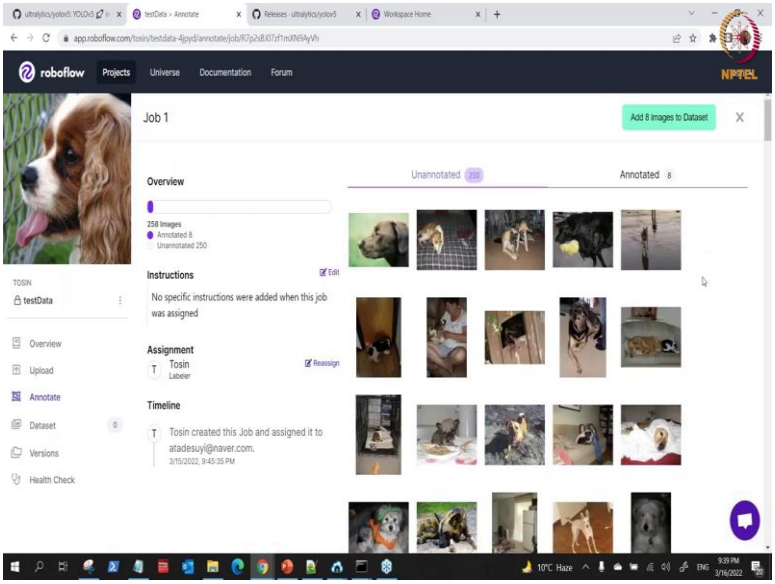
(Refer Slide Time: 36:29)



(Refer Slide Time: 36:41)

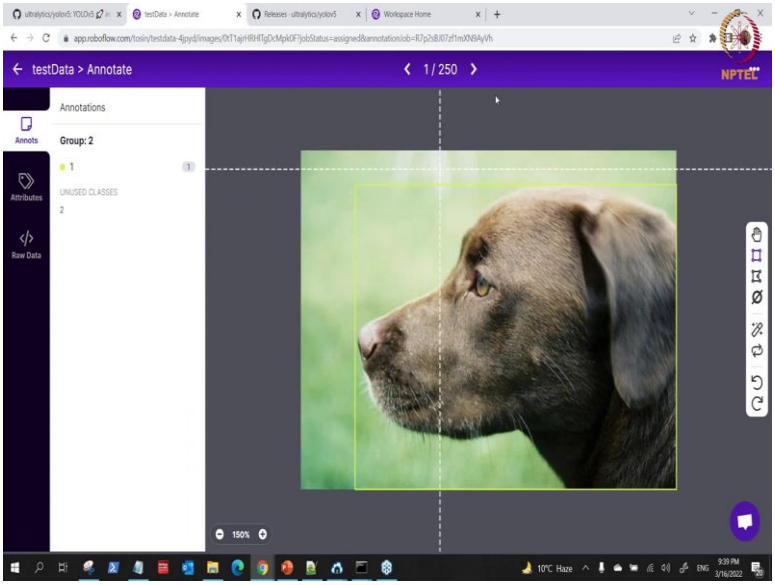


(Refer Slide Time: 36:47)



So, if you do not; if you are not signed up before you can sign up. So, once you sign up what happened is you upload, you create a new project and upload the data. So, when you upload the data for example, the one I have here then you go to annotation to annotate the data the. So, this is the data here.

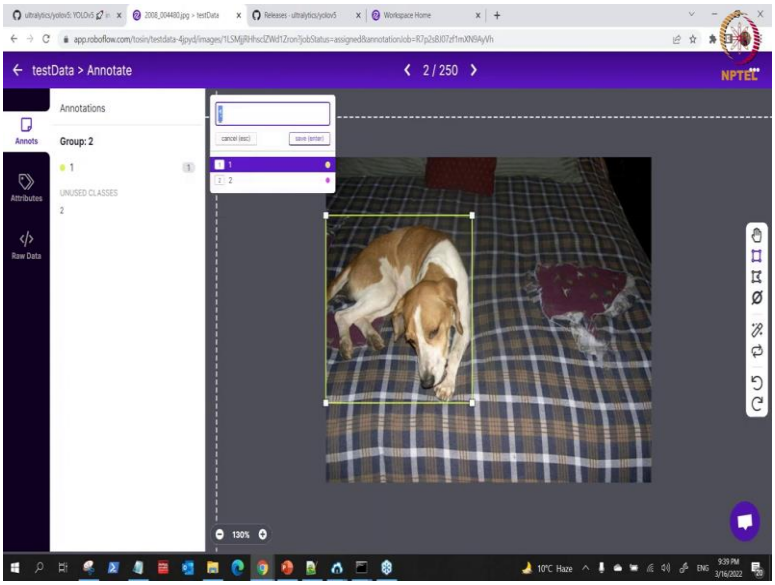
(Refer Slide Time: 36:50)



So, I can click on them you start annotating. You will have specified the number of classes because I am dealing with two classes here. So, how do you annotate this one?

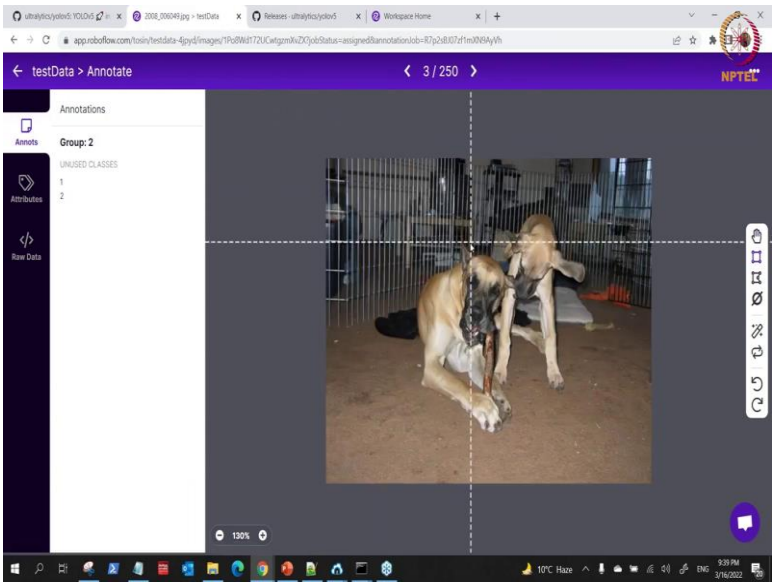
Once you are here, so you just click on the, you just have this. So, I can you draw the you draw the box, it is called box, you draw it there.

(Refer Slide Time: 37:16)

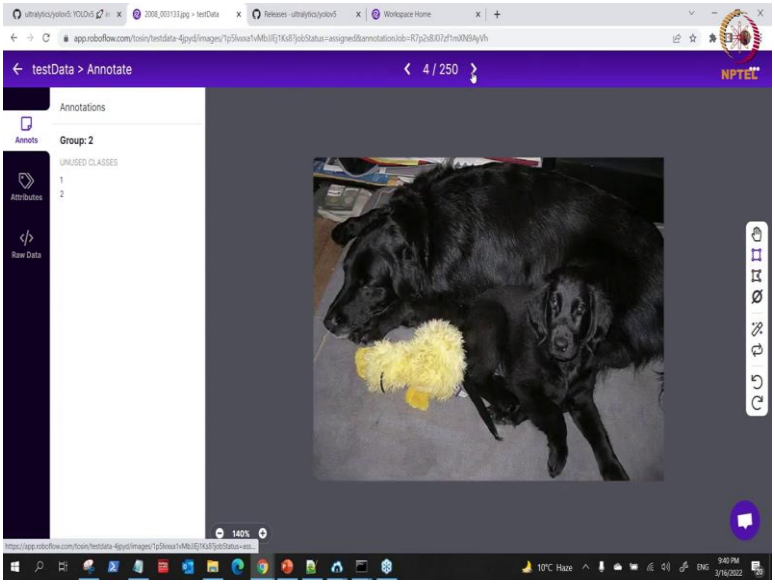


So, once you draw it there, so the class for dog is one you save that. Then you proceed again to the next one. So, you can draw the box on that one as well, draw it there, then you save the class is 1. By the time you get to another object which is not a dog, you continue doing the same way.

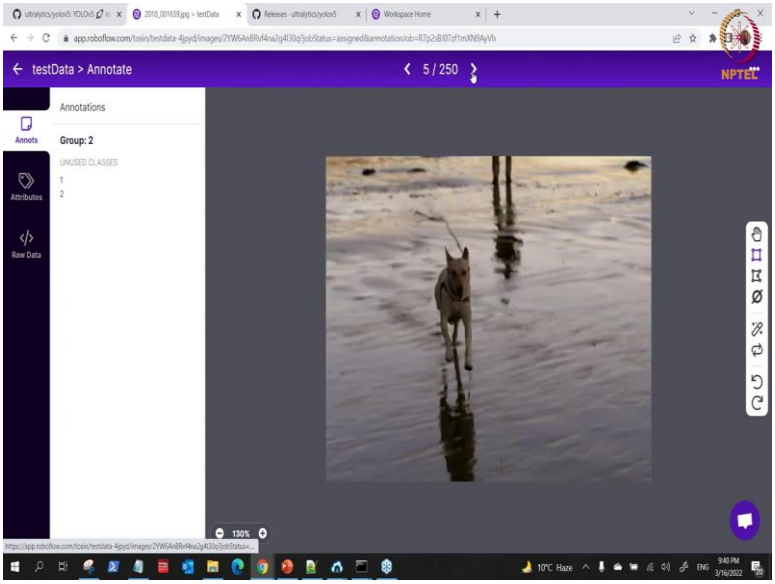
(Refer Slide Time: 37:30)



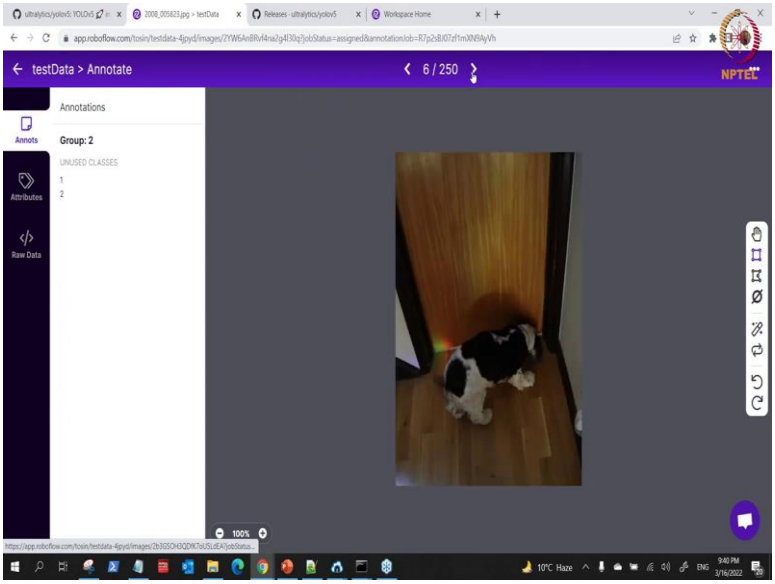
(Refer Slide Time: 37:33)



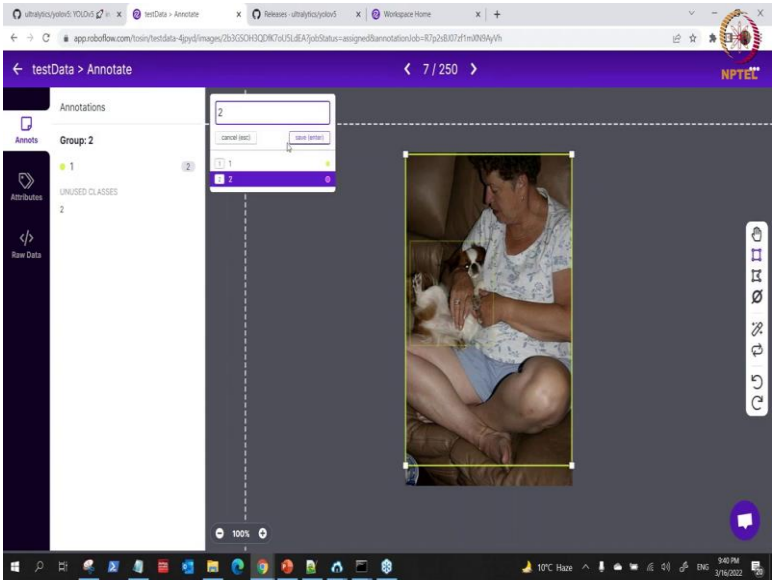
(Refer Slide Time: 37:33)



(Refer Slide Time: 37:35)



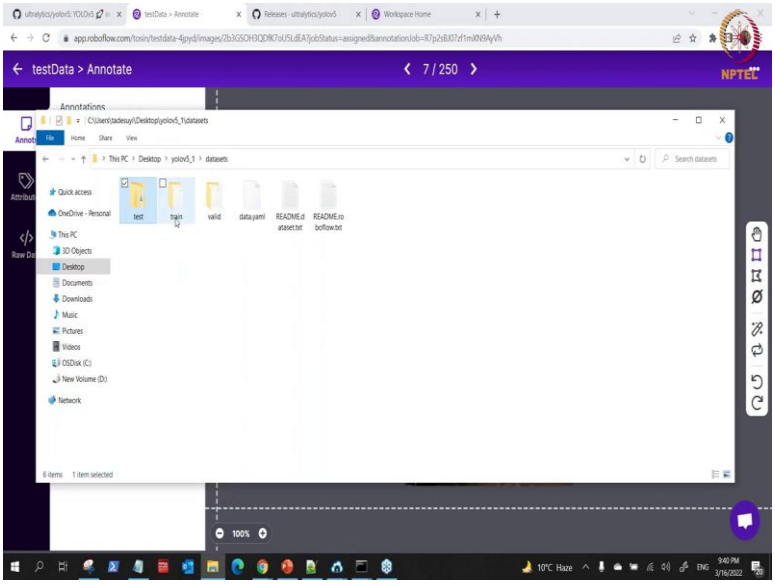
(Refer Slide Time: 37:36)



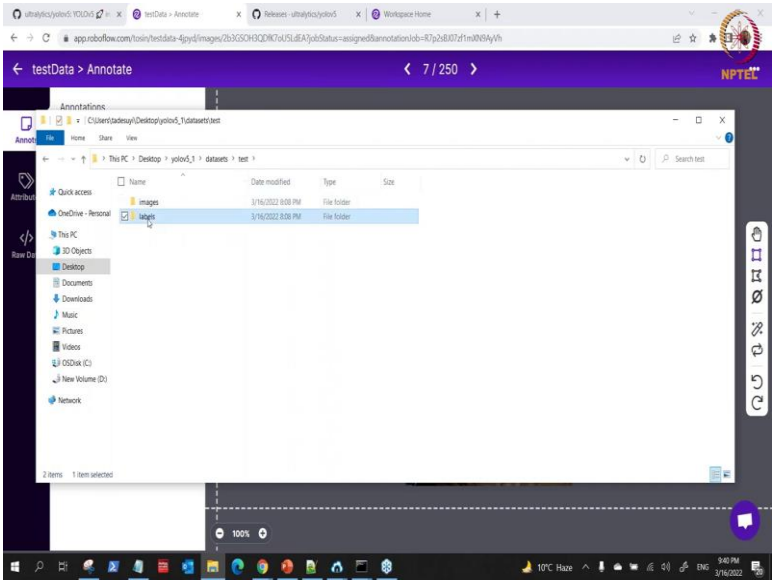
So, let me shift forward and see if I would say the one that include another one, ok. In this case, this is two class. So, what you need to do? I will draw the box for the dog this way the box, the box is can overlap.

Yes, I would say enter that is one for dog, then the class for person now. This is how I would do the one for person, yeah. So, a person which is what the class is 2 and then I can save that. So, that is how you do with Robo there.

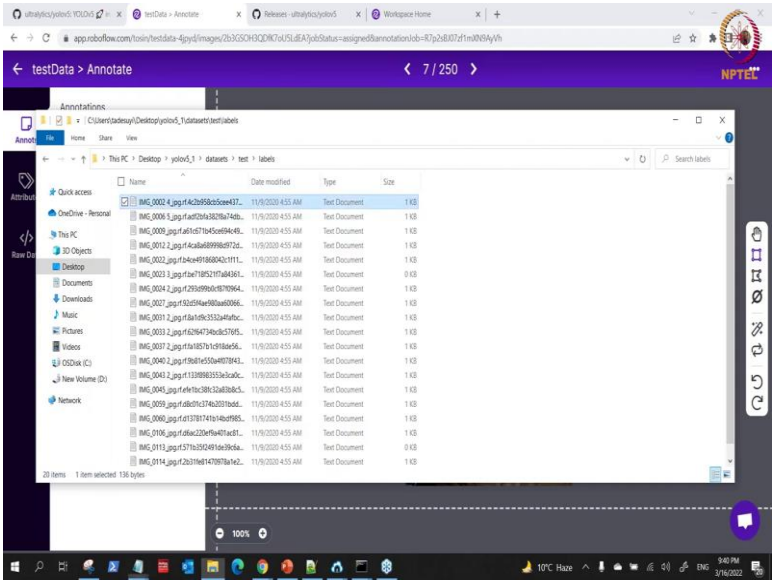
(Refer Slide Time: 38:05)



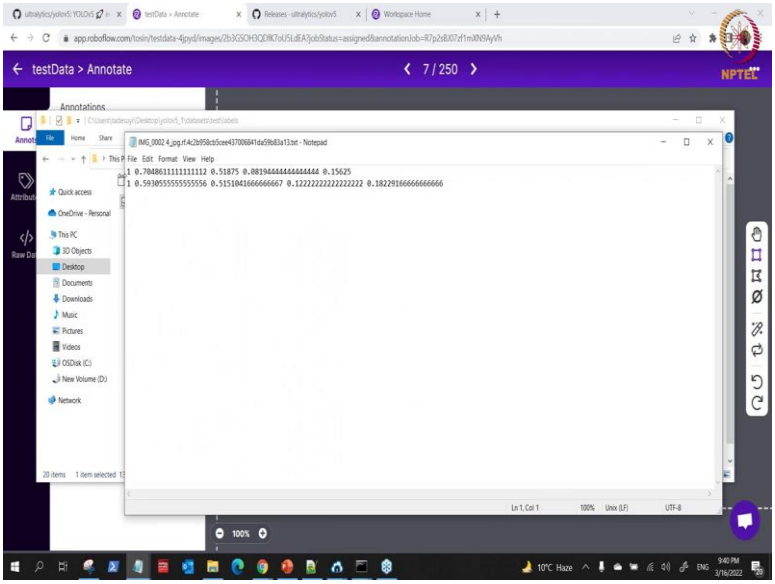
(Refer Slide Time: 38:11)



(Refer Slide Time: 38:14)



(Refer Slide Time: 38:15)

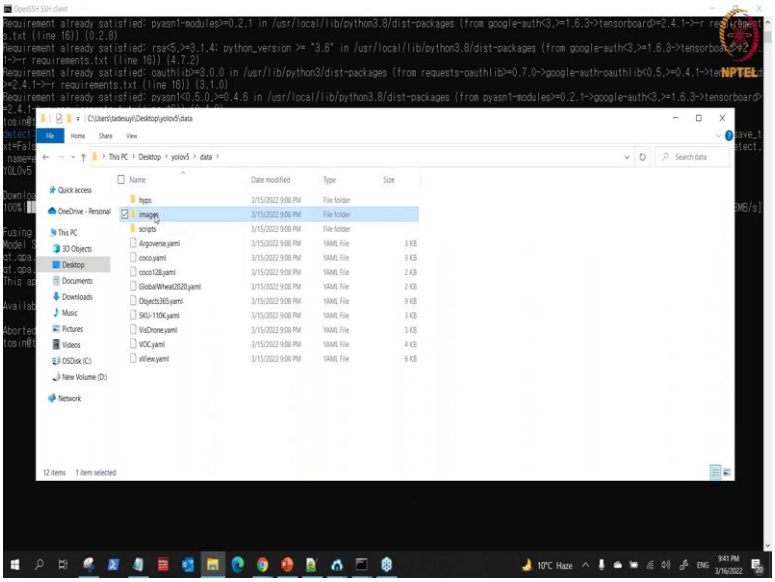


So, let us come back, yeah. I just show you how we came about this how we came about this label, that is how we came about this label. So, it is going to generate it in this way. So, once you draw the box this is the class, this is the for the X-center, Y-center, then the height, the width, and the height of the of the image.

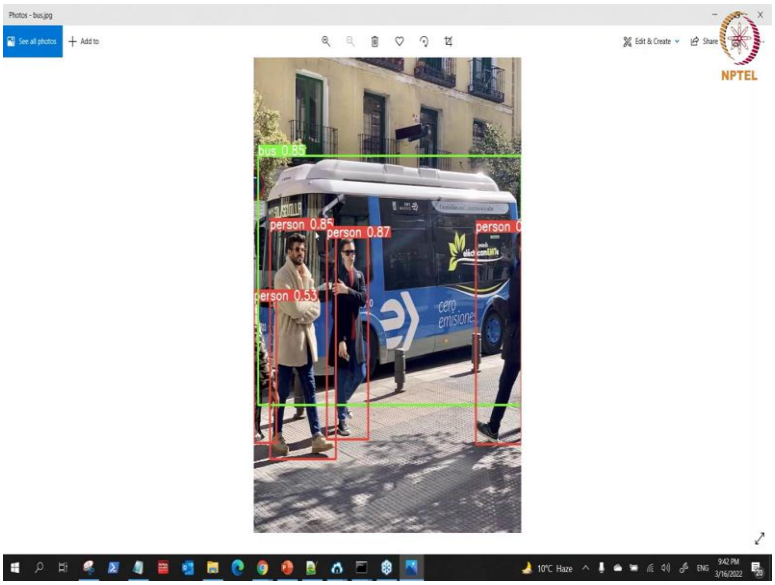
So, we are about to run them up, then before we run up let us just look at this. So, since that one is dumped, so what can we do? I can test with this image here. Let me test with

this. Just do that. We run. So, before I run that is there is an image here. So, in the YOLO which we opened. Where is the YOLO here? We downloaded.

(Refer Slide Time: 39:04)

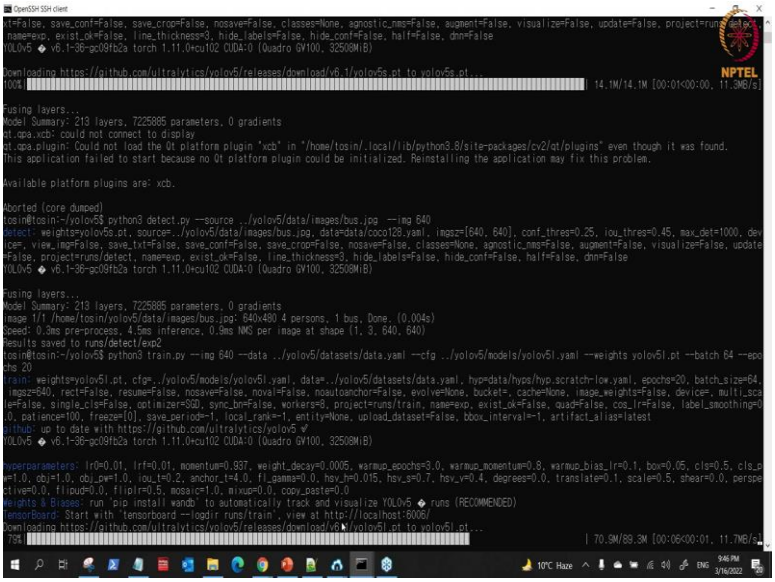


(Refer Slide Time: 39:10)



So, if we go to data then images there is this image here. So, we wanted to identify this images. I can use webcam. We can use it to identify this image and you will see what will happen to that.

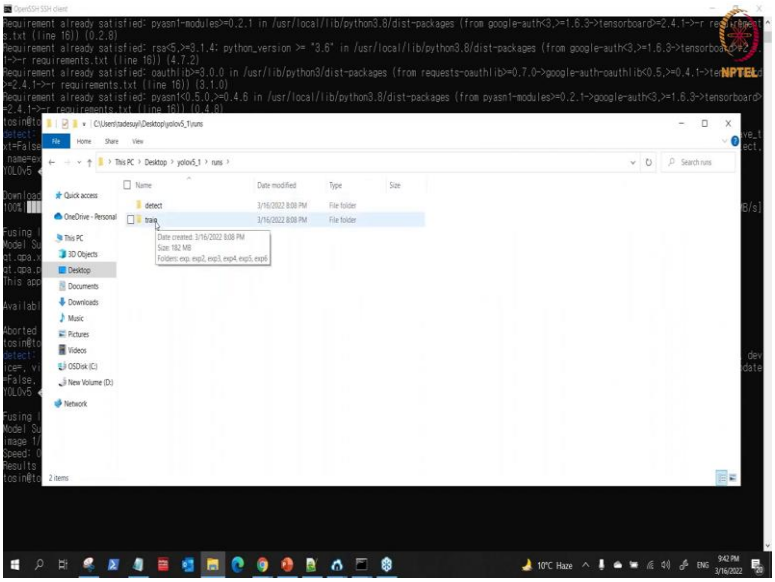
(Refer Slide Time: 39:26)



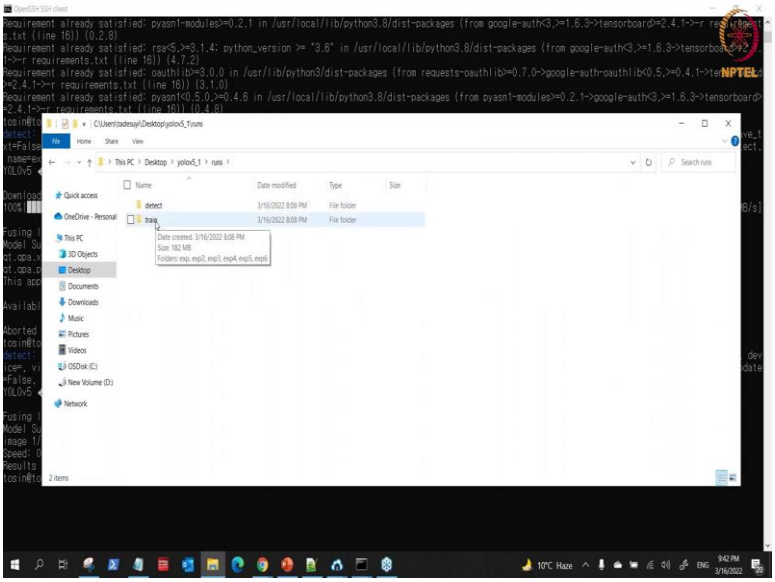
So, and so if I run this from the right track then if we run and save. So, we save it as save it. So, example of that. So, it is on my, it is on my workstation, then what I did is what a put image here somewhere, ok. That the image can be here, which is under the pitch, then I can see where we have bus here, ok, yeah.

So, you can see how it does the classification. It is able to recognize, and this is a bus, this is a person, this is a person again, this also a person, is able to recognize that. Well, if you have a I know I am not using ssh, you can use your; you can use these directs and even with that if you have run that I think I run one before now which is under run dot train, is it train or detects well, yeah, yes.

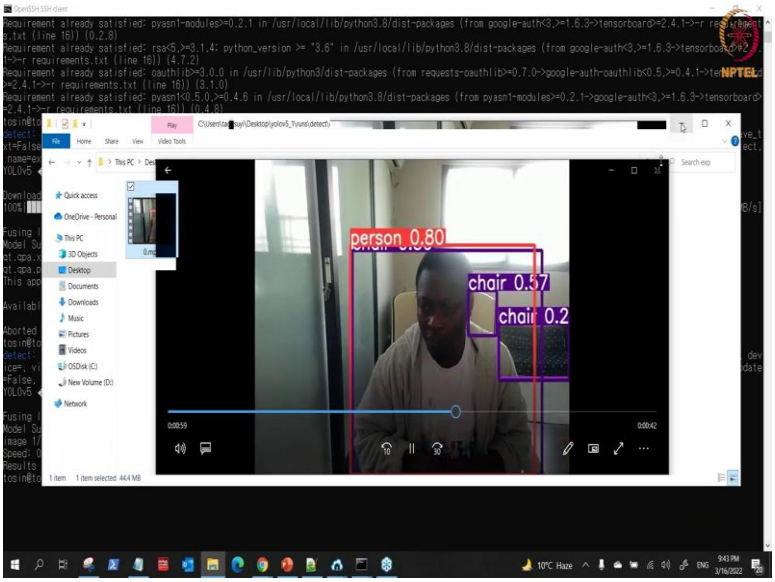
(Refer Slide Time: 40:27)



(Refer Slide Time: 40:30)



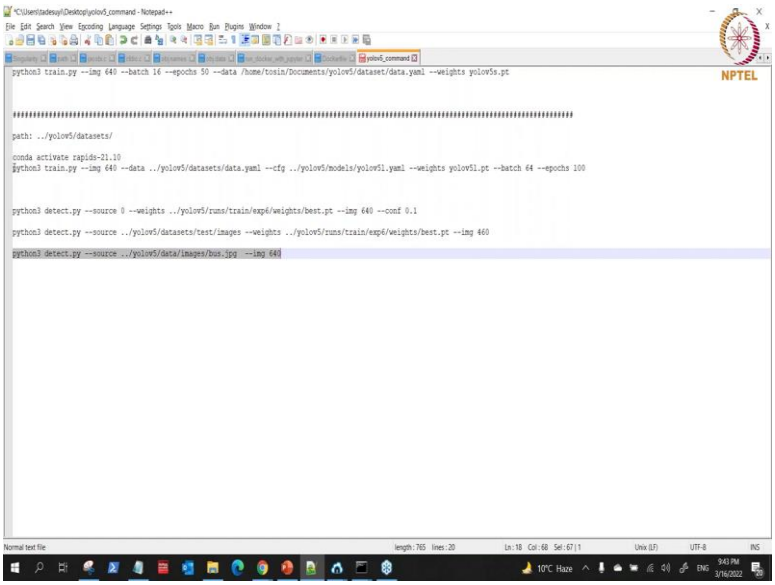
(Refer Slide Time: 40:37)



So, this is the way supposed to run you can see. This was a live one I did, not too long. It will be it will be recognizing this is a couch, this is a chair, that is how it will be running. We will be seeing all of that there.

So, now, let us just show let me just show you how do you train the custom. So, for the custom, what we need to do for the custom?

(Refer Slide Time: 41:02)



(Refer Slide Time: 41:30)

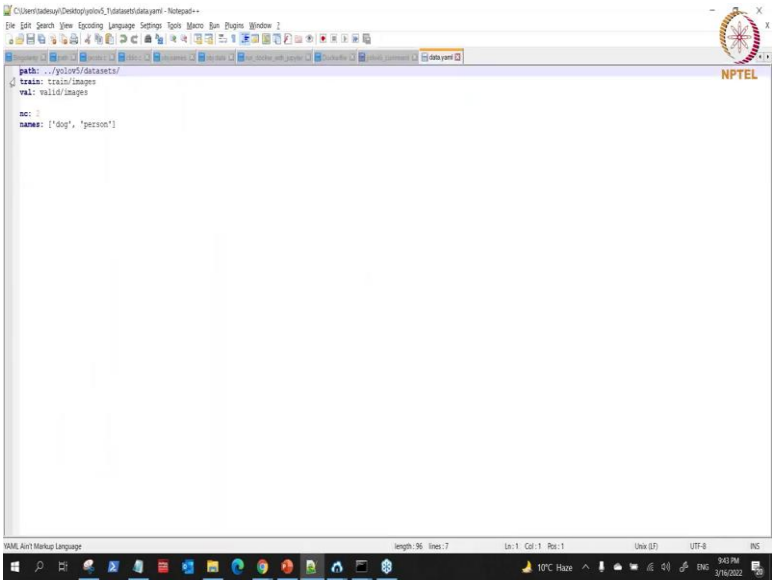
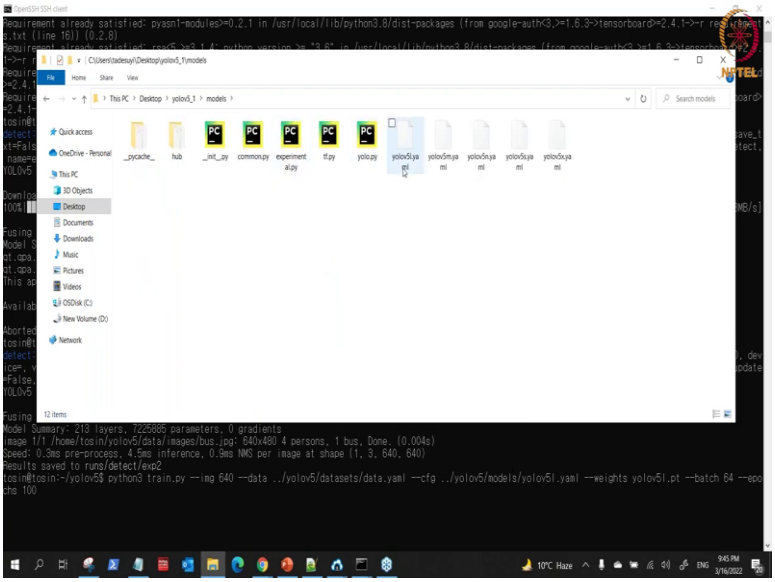


Image to train is to then it this, I will explain this to you. Since, we will set everything here into our YOLO; so into the YOLO what you need to do is this dataset is there. You have put your own data. This is your test, this is your train, this is your validation is there after you have download from Roboflow.

Then, this is the yaml file. Let me open this yaml file for you data.yaml file. So, you add this path there, this is YOLO.datasets, YOLO5 datasets. All this will have been there from Roboflow, will have given you all these train paths, validation path, this is number of classes, if detach your data maybe you are dealing with 10 classes what would be here will be 10. And we will list the names of all the what classes here. So, I am just dealing with two here, that is why I have this one here. So, this is how it looks like there.

So, to complete that one, so what I need to do here is, ok. So, I will need to run that here; I need to run that here then come in please working on this to put it there inside my model. Almost done with that, is copies, and bring it here, then just yeah, ok. So, here you have the train.py. This is the image size, this is the date, this is the path to my data which is datasets data.yaml. This is the configuration file. This configuration file is the one for YOLOv5 . The one for YOLOv5 is you can see from this path here model.

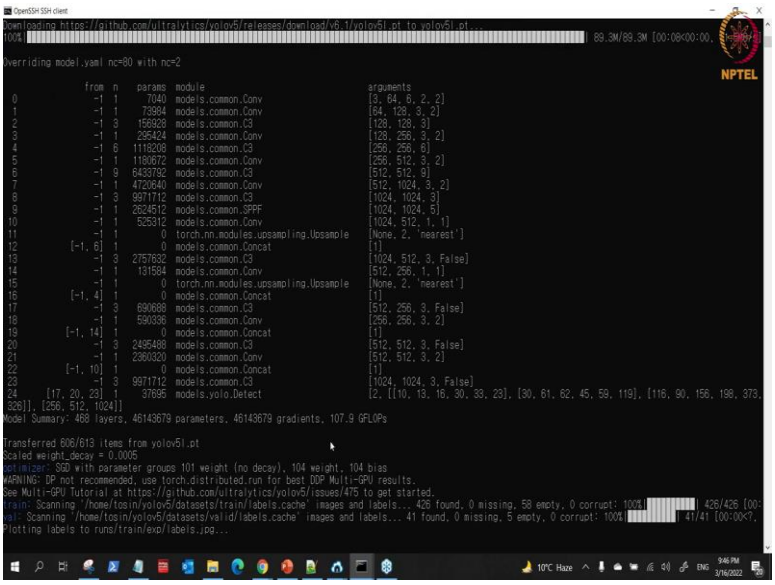
(Refer Slide Time: 43:26)



So, if I come back to model here, this is the path to the file here. Let me increase the view here, so that you can see the view. So, this is the folder here. This is YOLOv5.yaml. That is that for this then the weights. I am using this weights. It is going to download this with by itself. It does not exist 64 batch.

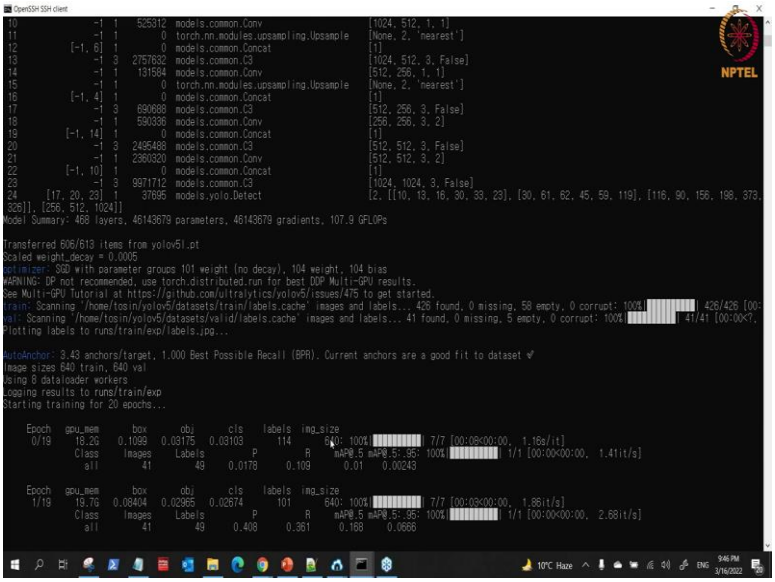
And the number of epoch is just a 100, but 100 would be too much for that. So, let me just say 20, and let us see what happened there.

(Refer Slide Time: 43:55)



So, this is how you would train. So, after you train you can also do inferencing. The same way see is downloading the weights itself or the large.

(Refer Slide Time: 44:02)



So, it is running the configuration and this is how you train with your custom data and you start inferencing just the way others, that is the way we have done the initial one as well. So, you can inference with that.

So, it keeps running well. Keeps running, I know you can do the inference yourself as well by following the presentation that we.

Thank you for listening.