



Currency Denomination Detection using YOLO

Step 1:- Gathering Data

- Gather images of the objects you want your model to detect
- In this Case we Used Indian Currencies of different denominations (ie. 10,20,50,100,200,500,2000)
- For a smaller dataset, take about 30-40 images per object (If you are training on apples and oranges, for examples, find 30-40 different images of apples, and 30-40 for oranges)
- Look for images that have different angles, colors, etc. Regarding the apples and oranges, try to find a variety of shapes and colors.

You Can Obtain DataSets from This Website <https://www.kaggle.com/>

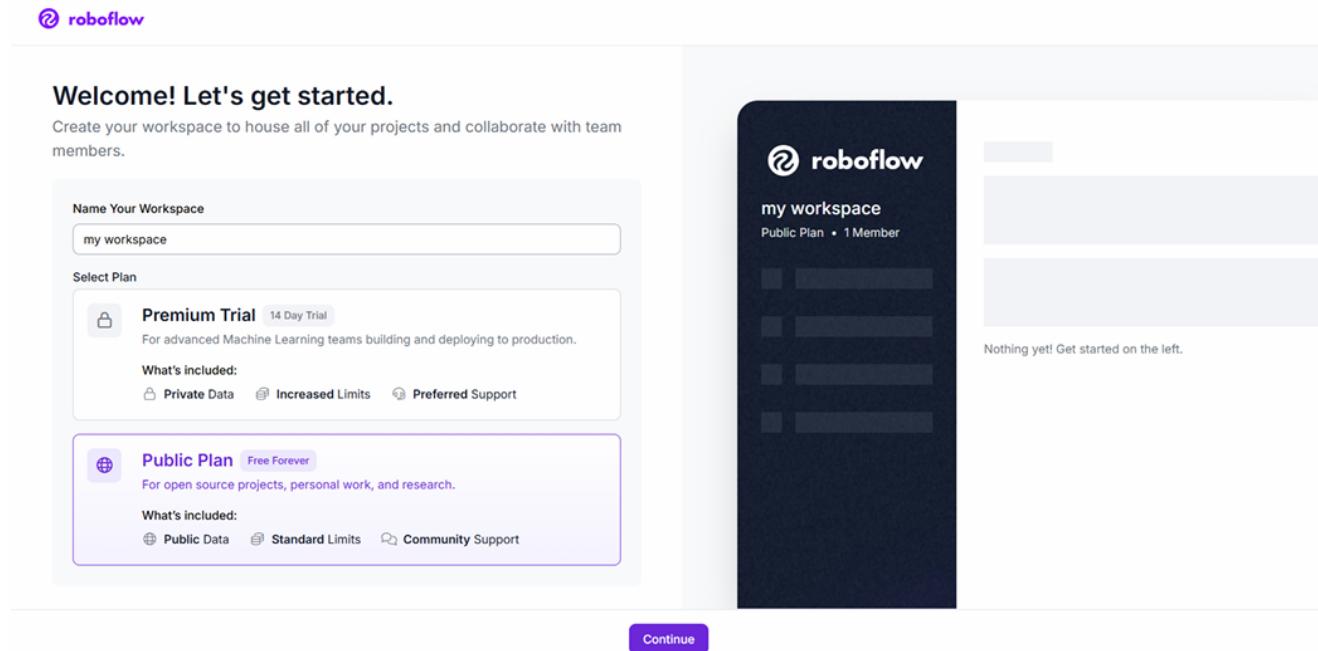
The screenshot shows a web browser displaying a Kaggle dataset page. The URL in the address bar is kaggle.com/datasets/yashdogra/2000-notes. The page has a sidebar on the left with links for Create, Home, Competitions, Datasets, Models, Benchmarks, Game Arena, Code, Discussions, Learn, More, and View Active Events. The main content area features a search bar and a navigation bar with a user profile, a '10' button, a 'Code' button, a 'Download' button, and a three-dot menu. The dataset title is 'Indian Currency Banknotes Dataset' by YASH DOGRA, updated 10 months ago. It includes a preview image of a 2000 Indian Rupee note featuring Mahatma Gandhi. Below the title, there's a brief description: 'Comprehensive Insights into the Design and Features of Indian Rupee Banknotes'. There are buttons for 'Data Card', 'Code (2)', 'Discussion (0)', and 'Suggestions (0)'. On the right side, there are sections for 'Usability' (rating 10.00), 'License' (Apache 2.0), and 'Expected update frequency' (Never). The overall layout is clean and organized, typical of a Kaggle dataset page.

Step 2:- Annotating

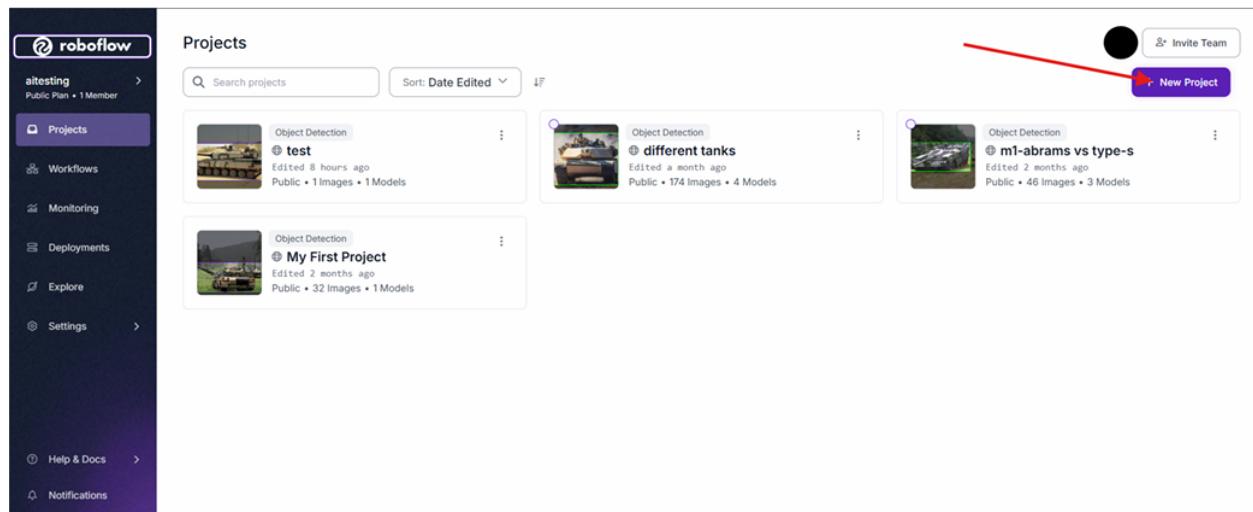
- Annotating is the process of drawing a box around the object in a photo, and giving it a name.
- This process tells the AI model that "This object is in this location, looks like this, and is called this"
- We will be using a free website called Roboflow to annotate, but there are others like it.

How to Annotate

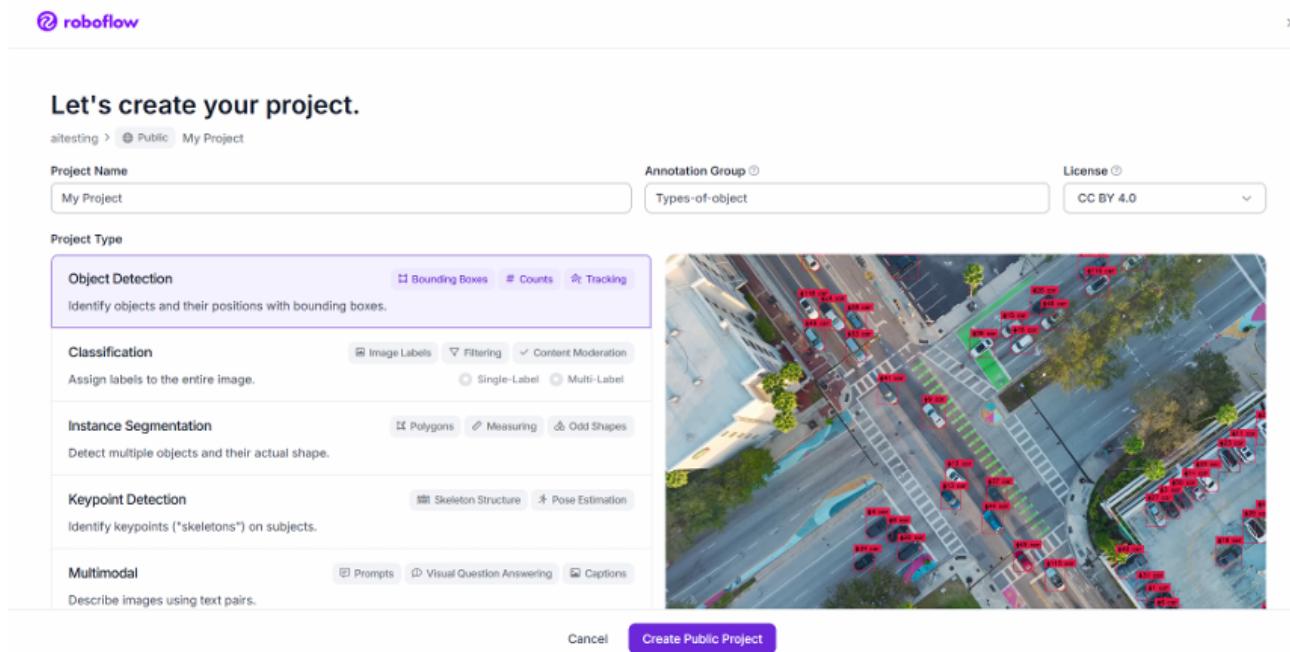
1. Open <https://roboflow.com/> and create an account
2. Create a workspace, Select public plan and continue.



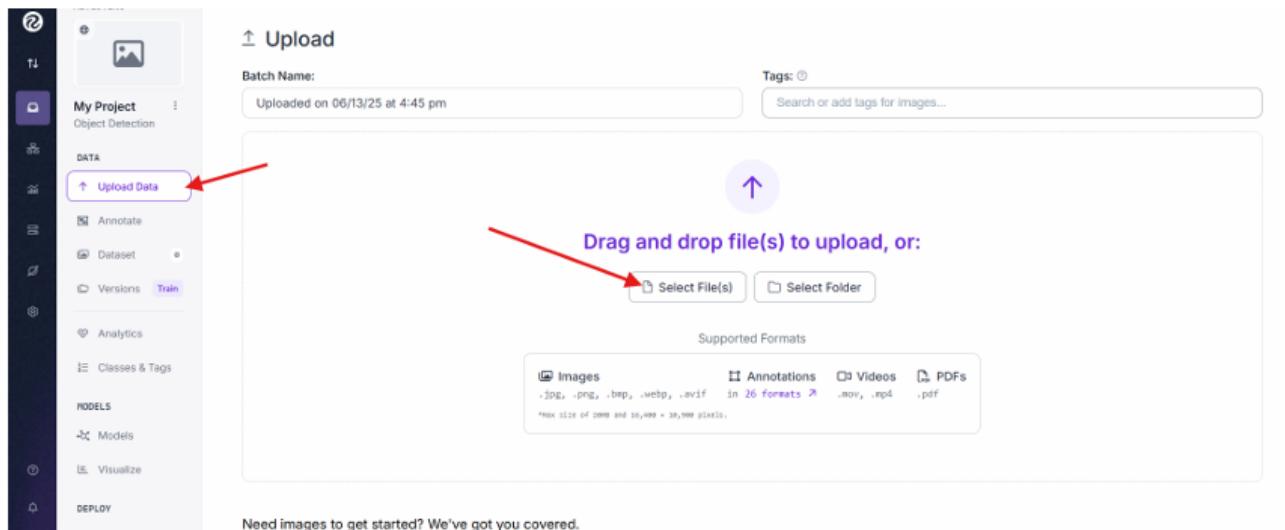
3. Create a project



4. Fill in name, Annotation group, select "Object Detection," and create project:

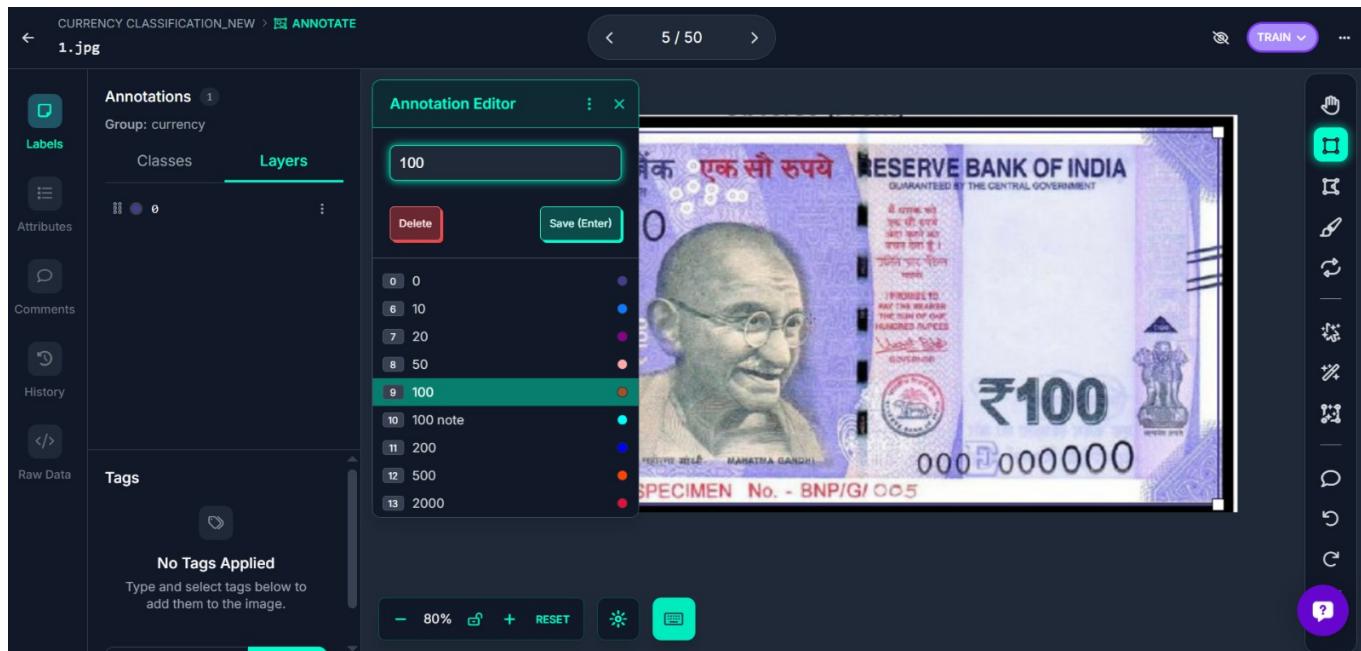


5. Navigate to the “Upload Data” tab and upload your images



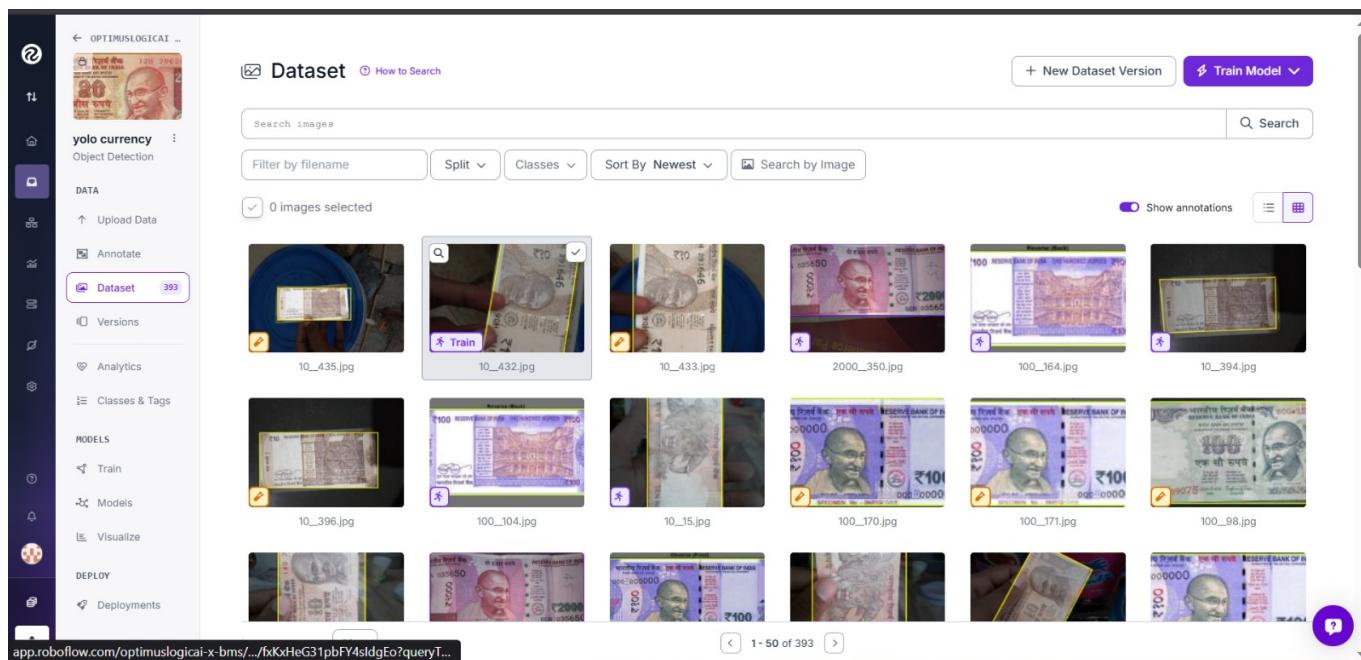
6. Click on “Start Labeling,” “Assign to myself,”

7. To label the image, draw a box around the object and enter the name of the it



8. Once all images are annotated, Navigate to the “Annotated” tab on the left and click “Add to Dataset” on the top right.

9. After Annotating , you can preview your DataSet



10. Split images between a train, valid, and test set. This is necessary. The recommended split is good.

Add Images To Dataset

Total Images to Add: 1

Method [What's Train, Valid, Test?](#)

Split Images Between Train/Valid/Test

Train 70% Valid 20% Test 10%

Image Distribution

Train: 1 images
Valid: 0 images
Test: 0 images

You are about to add 1 images to the dataset.
0 images will be sent back as part of a new job.

11. On the versions tab(left), follow the steps to create a version.

a. Augmentation could prove useful for you, because it duplicates images and introduces imperfections, which could strengthen the accuracy of your model in imperfect situations.

12. Create the version. We will get back to this later.

Versions

+ Create New Version

Version	Created At	Annotations	Notes	Owner
v8	2026-02-04 12:22pm	393		S Pranav Krishna
v6	2026-02-04 11:39am	393	Notes	s.saaswath@gmail.c...
v5	2026-02-04 11:23am	393		S Pranav Krishna
v3	2026-02-03 5:17pm	393		s.saaswath@gmail.c...
v2	2026-02-03 5:11pm	393		Rohit D S

Roboflow Instant 1 [Eval]

This version doesn't have a model.

Train Model

Uses Roboflow Credits View usage ↗

How to Upload Custom Weights

393 Total Images

View All Images →

Step 3:- Training The Yolo Model

1. Open A Colab File ([Colab](https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolo11-object-detection-on-custom-dataset.ipynb))

```

NOTE: To make it easier for us to manage datasets, images and models we create a HOME constant.

[1]: 
  import os
  HOME = os.getcwd()
  print(HOME)

... /content

[2]: 
  %pip install "ultralytics<=8.3.40" supervision roboflow
  # prevent ultralytics from tracking your activity
  !yolo settings sync=False
  import ultralytics
  ultralytics.checks()

Ultralytics 8.3.2 🚀 Python-3.10.12 torch-2.4.1+cu121 CUDA:8 (Tesla T4, 15102MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 41.2/112.6 GB disk)

```

2. Run everything before “Inference with model pre-trained on COCO dataset”(this is not necessary for building the model, but you can run it if you want)

3. Scroll down to “Fine-tune YOLO11 on custom dataset”

```

NOTE: When training YOLOv11, make sure your data is located in datasets. If you'd like to change the default location of the data you want to use for fine-tuning, you can do so through Ultraalytics' settings.json. In this tutorial, we will use one of the datasets available on Roboflow Universe. When downloading, make sure to select the yolov11 export format.

[1]: 
  !mkdir {HOME}/datasets
  %cd {HOME}/datasets

  from google.colab import userdata
  from roboflow import Roboflow

  ROBOFLOW_API_KEY = userdata.get('ROBOFLOW_API_KEY')
  rf = Roboflow(api_key=ROBOFLOW_API_KEY)

  workspace = rf.workspace("liangdianzhong")
  project = workspace.project("-qvdmw")
  version = project.version(3)
  dataset = version.download("yolov11")

```

4. Fill in name, Annotation group, select “Object Detection,” and create project:

Versions

[+ Create New Version](#)

Versions

v9 2026-02-04 3:15pm
Generated on Feb 4, 2026 by s.saaswath@gmail.com

[Download Dataset](#) [Edit](#)

This version doesn't have a model.

Train an optimized, state of the art model with Roboflow or upload a custom trained model to use features like Label Assist and Model Evaluation and deployment options like our auto-scaling API and edge device support.

[Train Model](#) Uses Roboflow Credits [View usage](#)

[How to Upload Custom Weights](#)

393 Total Images [View All Images](#)

caix-bms/yolo-currency/9/export

5. Click on "Download Dataset"

3 [Download](#)

Image and Annotation Format

YOLOv11

TXT annotations and YAML config used with [YOLOv11](#).

Download Options

Download zip to computer
Downloads all images, annotations, and classes.

Show download code
Custom train this dataset using the provided code snippet in a notebook.

[Cancel](#) [Continue](#)

6. Select YOLOv11 and Show download code.

7. Copy the Snippet

8. Navigate back to the Collab Document, select the text highlighted below and paste the text from roboflow website

```

!mkdir {HOME}/datasets
%cd {HOME}/datasets

from google.colab import userdata
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="Your API key appears here from the text you copied")
project = rf.workspace("optimuslogicai-x-bms").project("yolo-currency")
version = project.version(6)
dataset = version.download("yolov11")

```

9. updated snippet looks something like this

```

!mkdir {HOME}/datasets
%cd {HOME}/datasets

from google.colab import userdata
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="Your API key appears here from the text you copied")
project = rf.workspace("optimuslogicai-x-bms").project("yolo-currency")
version = project.version(6)
dataset = version.download("yolov11")

... mkdir: cannot create directory '/content/datasets': File exists

```

10. Next to the "model=", you can chance the size of the model you want to train. Yolo11n is the smallest, yolo11l and x are the largest. For now, its fine to use yolo11n.

```

%cd {HOME}

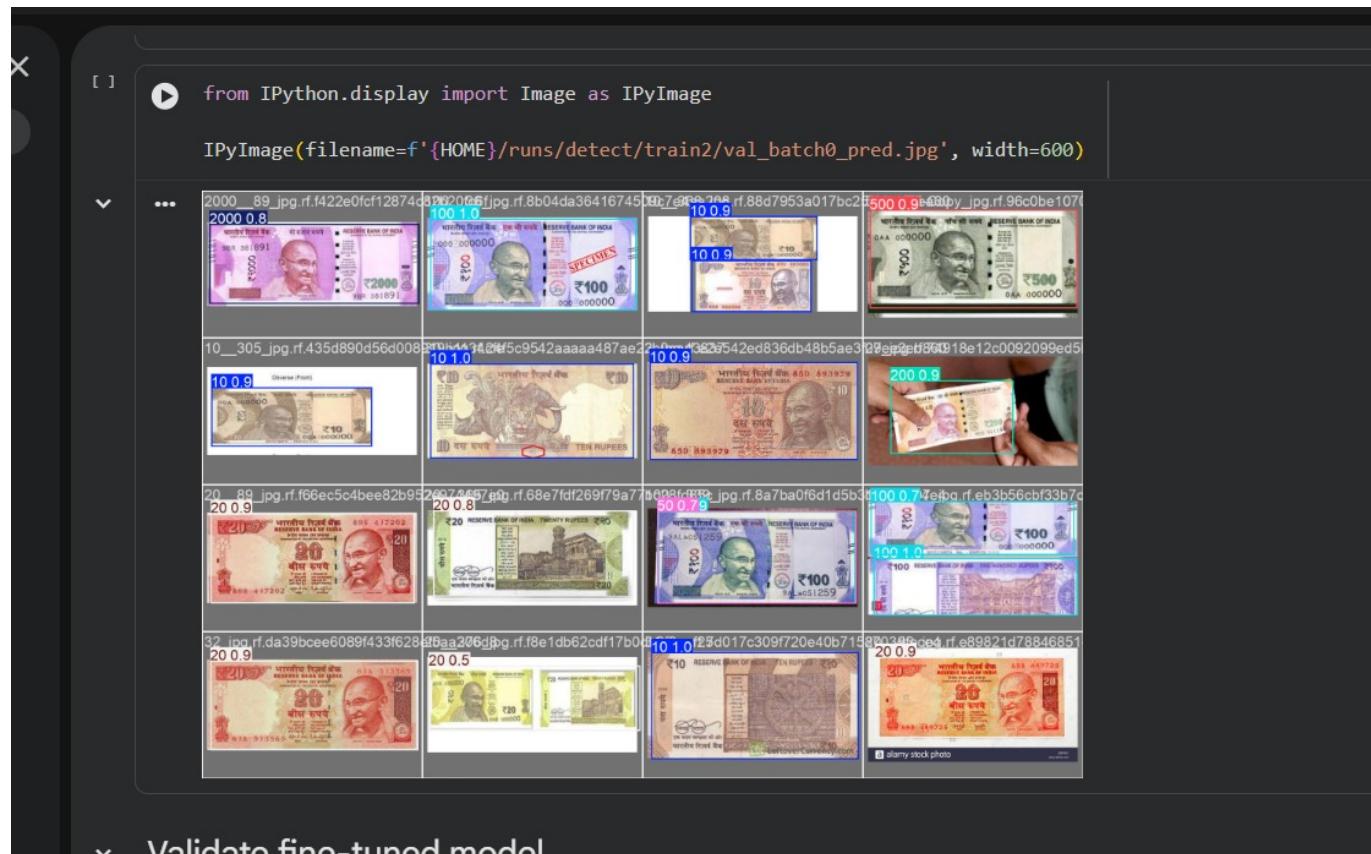
!yolo task=detect mode=train model=yolo11s.pt data={dataset.location}/data.yaml epochs=10 imgsz=640 plots=True

```

11. Next to "epochs='", this determines the amount of cycles your model is trained for. For a final model, 100-300 epochs are usually a good amount. For now, choose anywhere from 10-50 epochs. If you're unsure, use 10.

12. Run the snippet and wait for it to finish. This may take a while.

13. After Running All the Code Snippets you will see different images and their IDs



Validate fine-tuned model

13. Congrats! You now have a fully trained model.

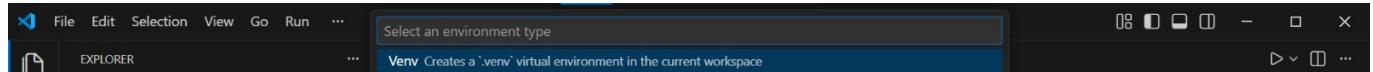
Step 4:- Local Deployment

1. In order to download the trained model onto your computer, open the file browser tab on the left
2. Navigate to datasets → runs → detect → train → weights → look for a file called "best.pt" and download it to your computer.

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
1/1	3.57G	1.634	4.677	0.9326	402	640: 100% 57/57 [00
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100%
	all	259	4652	0.0619	0.113	0.058 0.0312

3. Open a code editor like Visual Studio Code and create a new python file.

4. In Visual Studio Code or any other IDE, Create a Virtual Environment (.venv)



5. Activate the venv and install ultralytics, OpenCV, Numpy using command given below

```
PS C:\Users\Saaswath\Documents\OptimusLogic> ./venv/Scripts/Activate.ps1
(.venv) PS C:\Users\Saaswath\Documents\OptimusLogic> pip install
ultralytics
```

6. Make sure that the best.pt file, and Python file that you are going to create now are both in the same directory

7. Use the code oldeploy1.py given to run the program to test the model and save result to notepad file

8. source = "0" - change this line from zero to "file path" to give any test images to the model instead of taking it from the webcam

9. Open a new terminal on VS Code in the same venv

10. Enter the command

```
python -m http.server 8000
```

```
PS C:\Users\Saaswath\Documents\OptimusLogic> & C:/Users/Saaswath/Documents/OptimusLogic/.venv/Scripts/Activate.ps1
(.venv) PS C:\Users\Saaswath\Documents\OptimusLogic> python -m http.server 8000
```

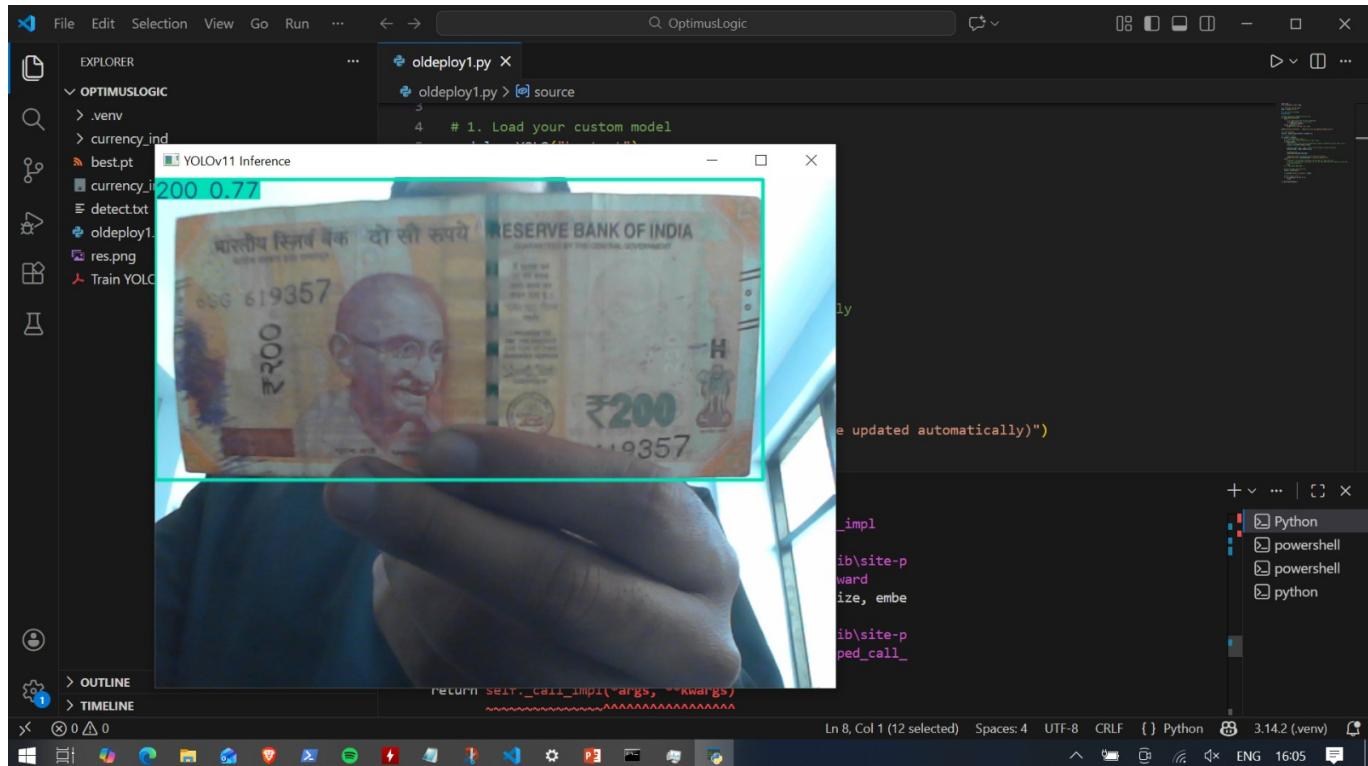
11. This creates a server using port 8000 and the computer acts as server with access granted thru IPv4 address of the computer

12. The txt file can be accessed using

<https://:8000/detect.txt>

Final Results

1. On running the Python Program , your Computer's WebCam will turn on and there will be a live feed on your display as shown below.



2. After Running the Server ,you can open your web browser (Chrome,FireFox,Brave,Edge,Internet Explorer and so on), paste in the Ip Address as mentioned before.You will get the following shown on your web browser.

