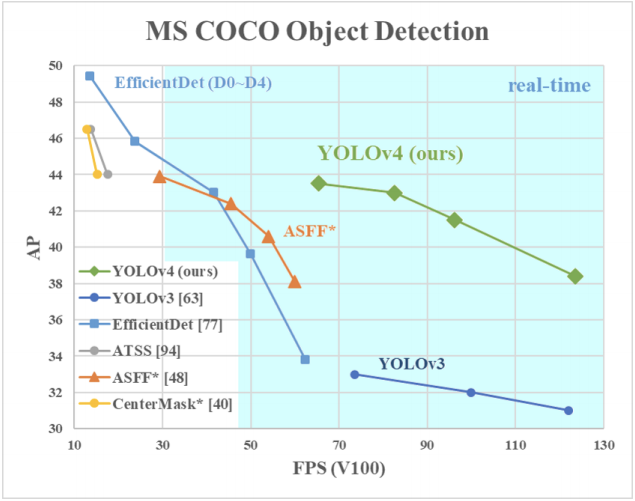
Task 2 - Detection of different colored Face Masks (yellow, black, green) using Yolov4 algorithm

By- Rachit S Garg

(17070124054) (BTech in Information Technology) (8th Semester)

**Introduction to YoloV4**

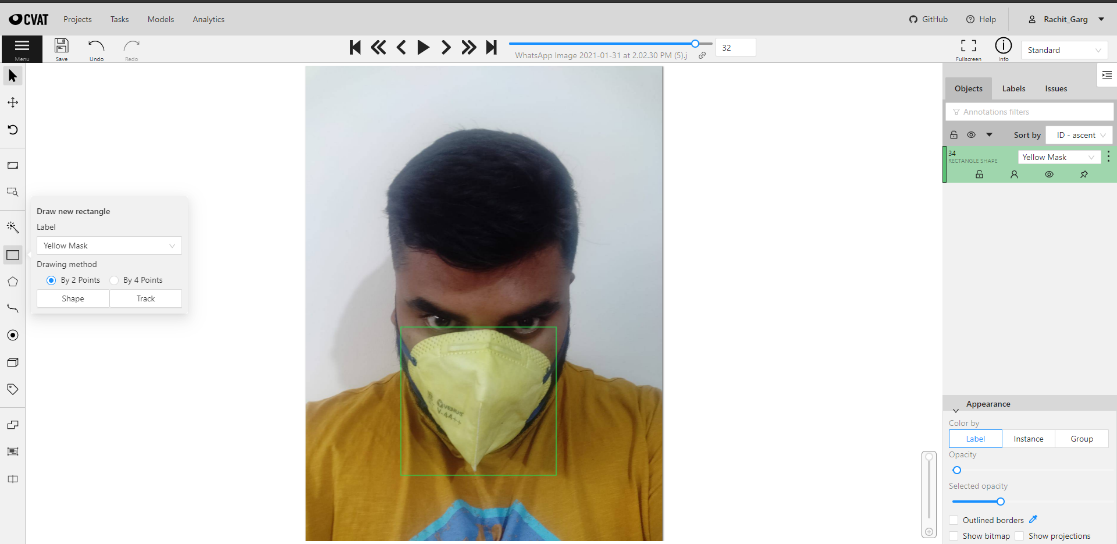
1. By using YoloV4 we get better performance and speed as compared to the previous version of Yolo Version
2. YoloV4 Shown to be the new object detection champion by standard metrics on COCO

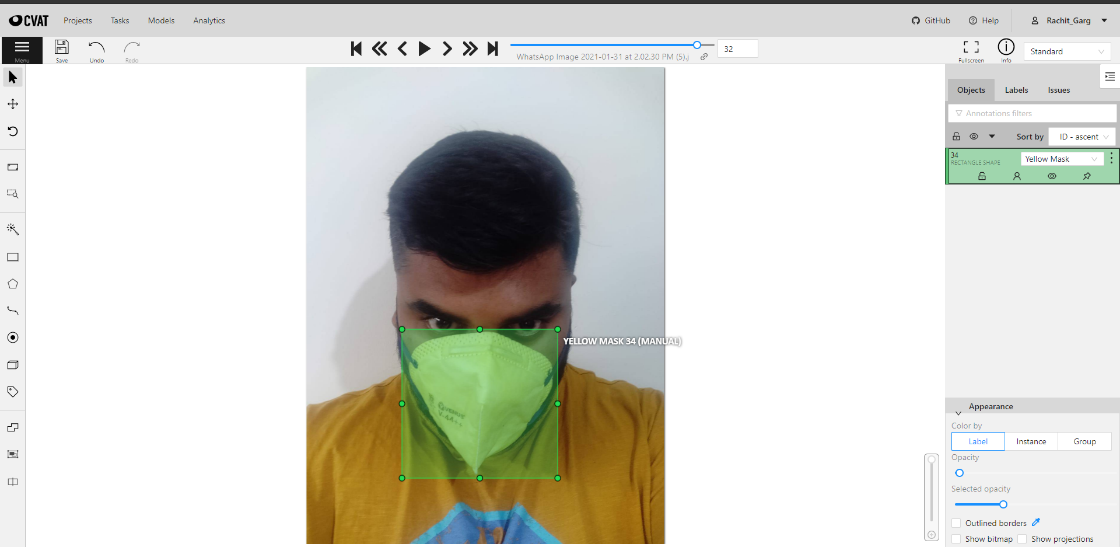


1. I use the [**Darknet framework**](https://roboflow.com/formats/yolo-darknet-txt) because of the ability to train YOLOv4
2. While Darknet is not as intuitive to use, it is immensely flexible, and it advances state-of-the-art object detection results.

**Data Labeling and Augmentation Techniques**

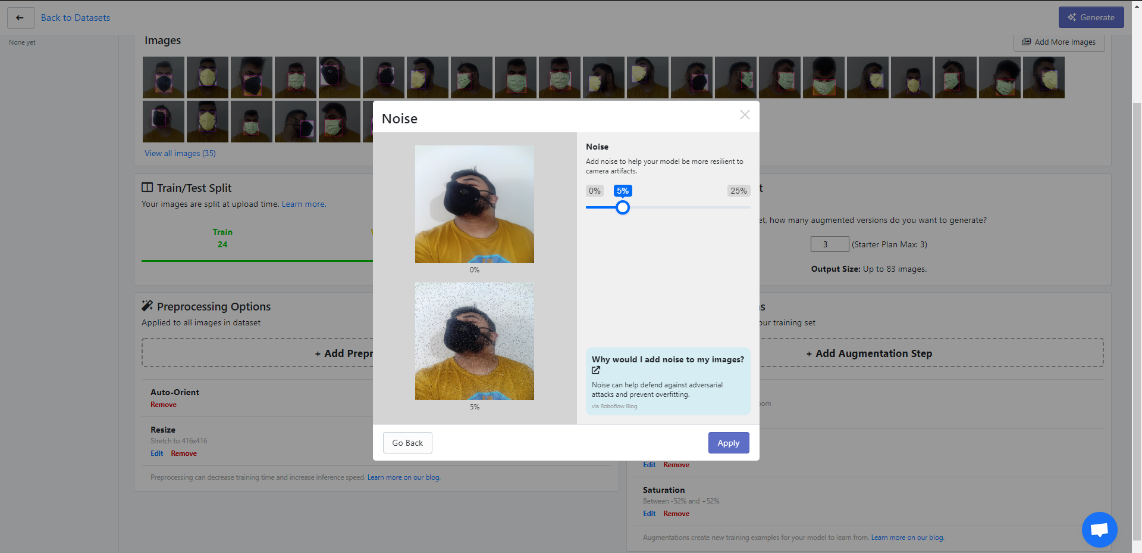
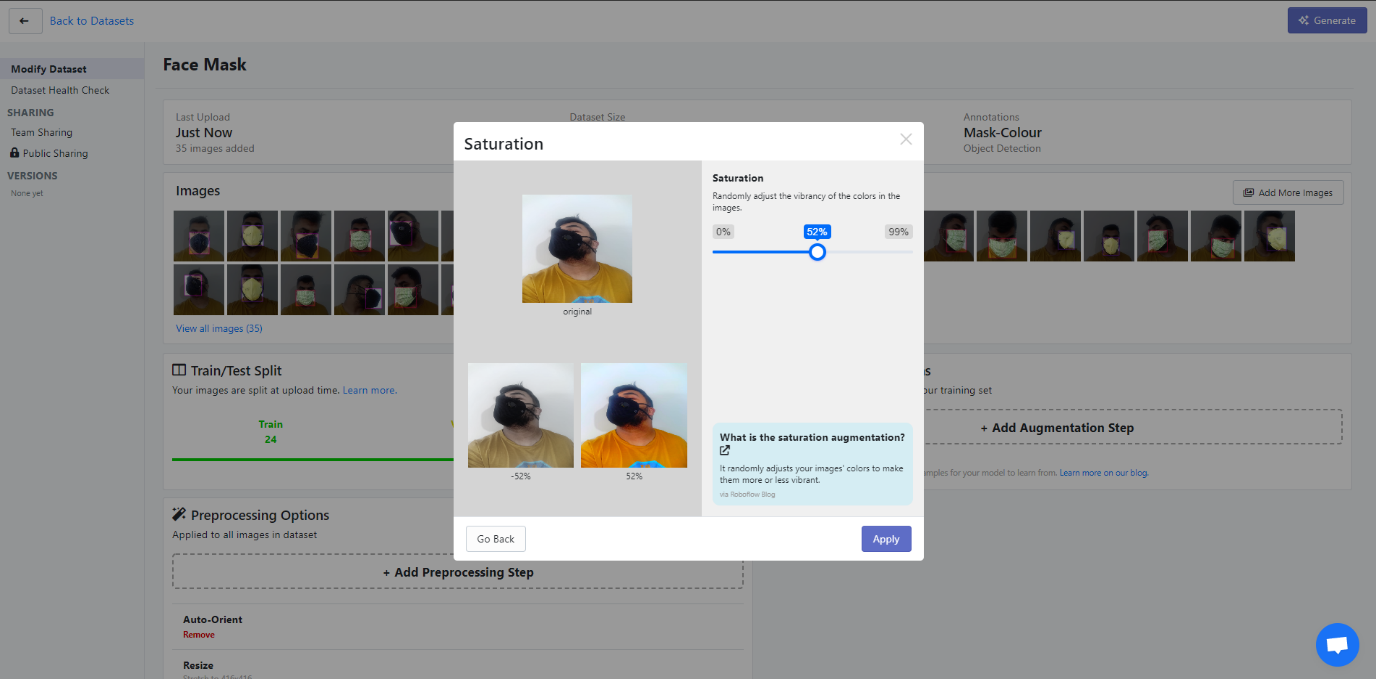
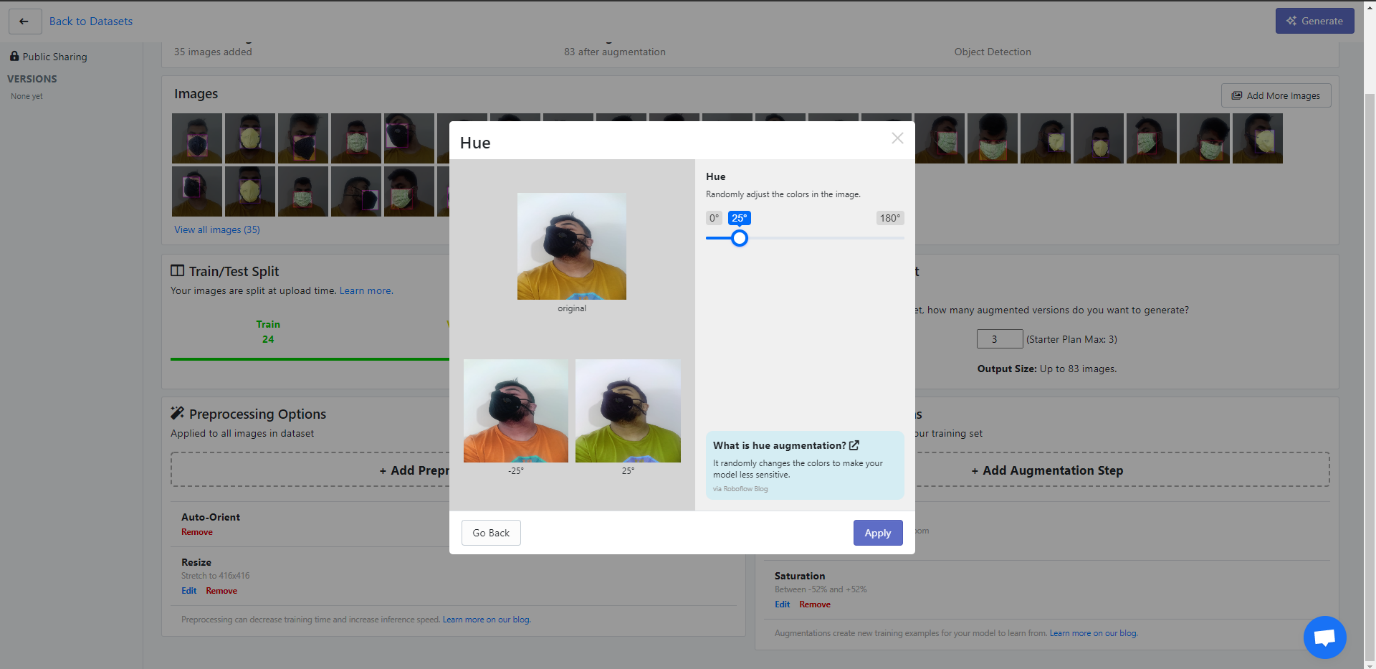
To train YOLOv4 on Darknet with our custom dataset, I need to import our dataset in [**Darknet YOLO format**](https://roboflow.com/formats/yolo-darknet-txt). To import our images and bounding boxes in the [**YOLO Darknet format**](https://roboflow.ai/formats/yolo-darknet-txt), I use **[Roboflow](https://roboflow.ai/)**.



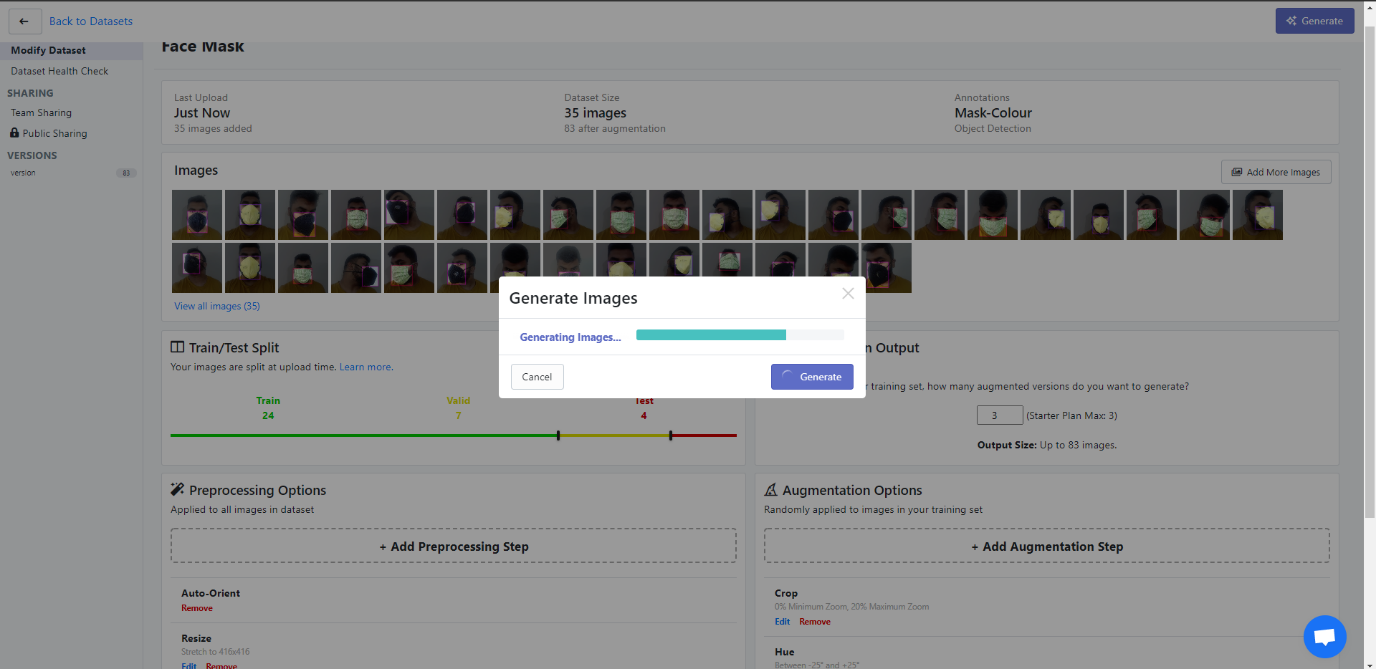


**Augmentation Techniques**

Using Roboflow I used Augmentation techniques for increasing images in the dataset



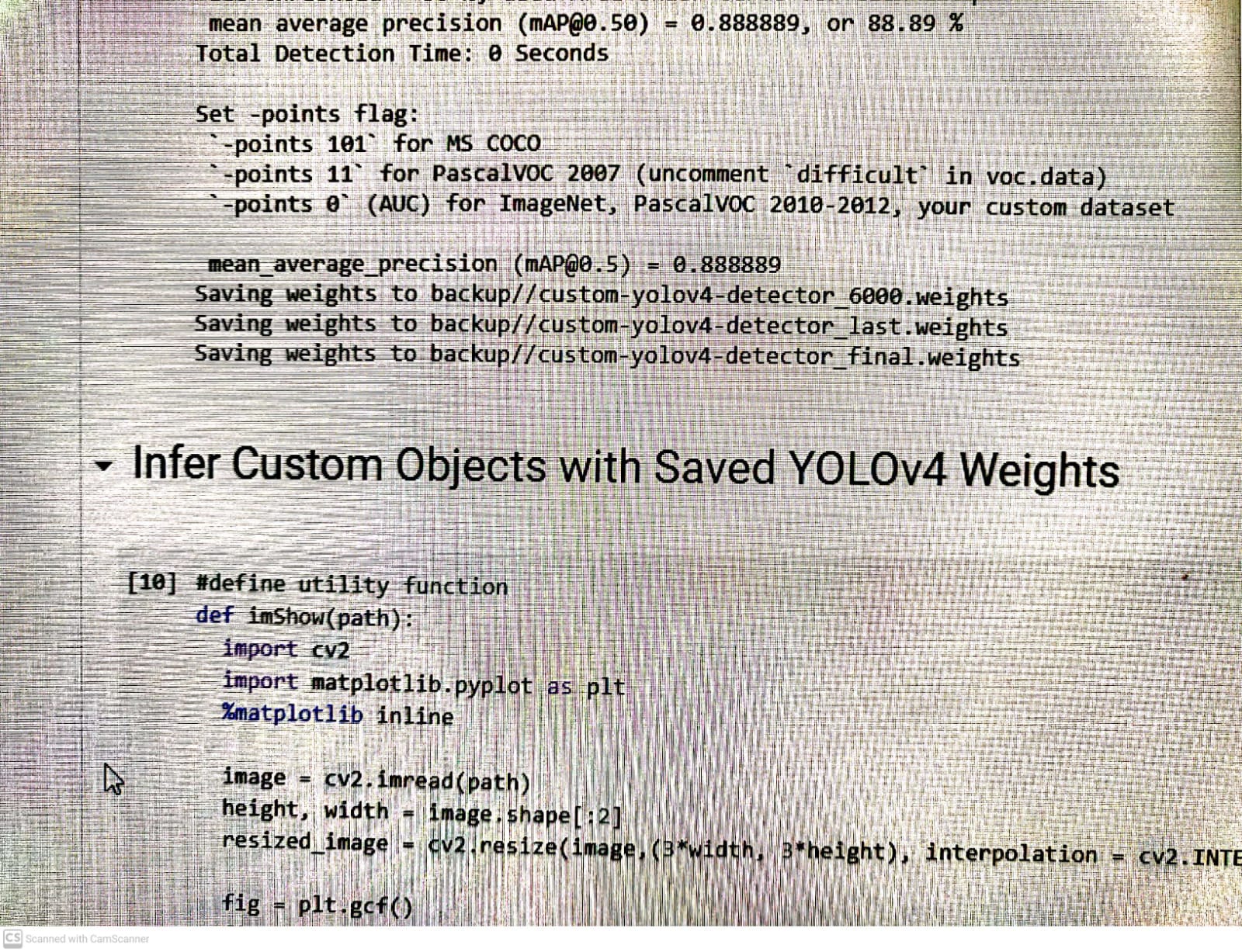
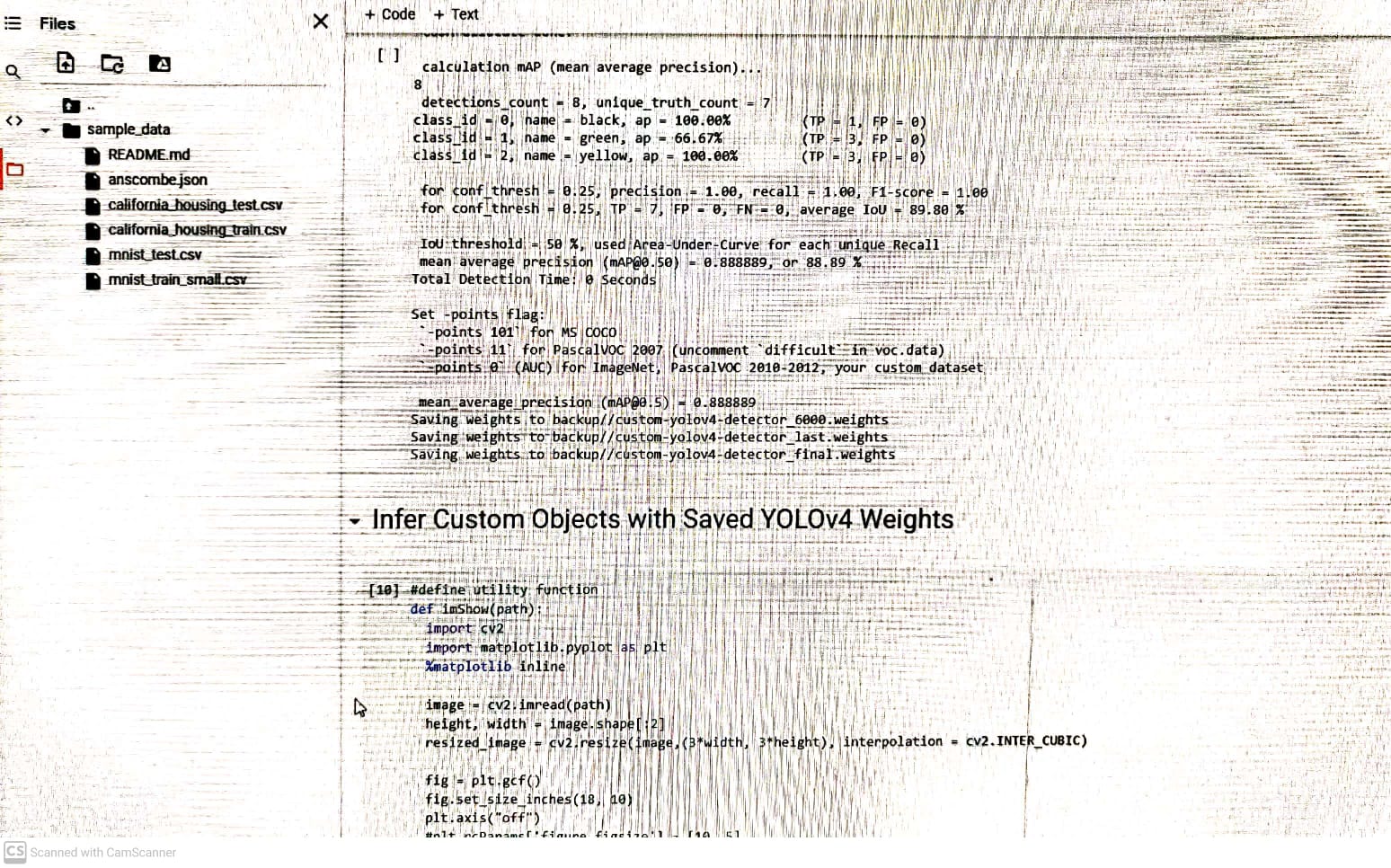
**Generating Yolo Darknet Format Data**



**Steps to implement YOLOv4 on our custom data:**

1. Introducing YOLO v4 versus prior object detection models
2. Configure our [**YOLOv4 GPU environment on Google Colab**](https://colab.research.google.com/drive/1mzL6WyY9BRx4xX476eQdhKDnd_eixBlG#scrollTo=GNVU7eu9CQj3)
3. Install the [**Darknet YOLO v4 training environment**](https://github.com/roboflow-ai/darknet)
4. Download our [**custom dataset for YOLOv4**](https://public.roboflow.ai/object-detection/bccd) and set up directories
5. Configure a custom [**YOLO v4 training config file**](https://github.com/roboflow-ai/darknet/blob/master/cfg/yolov4.cfg) for Darknet
6. Train our custom YOLO v4 object detector
7. Reload YOLO v4 trained weights and make inference on test images

**Images of Weights Getting Train with its Accuracy**

****

**Images of labeled data**



