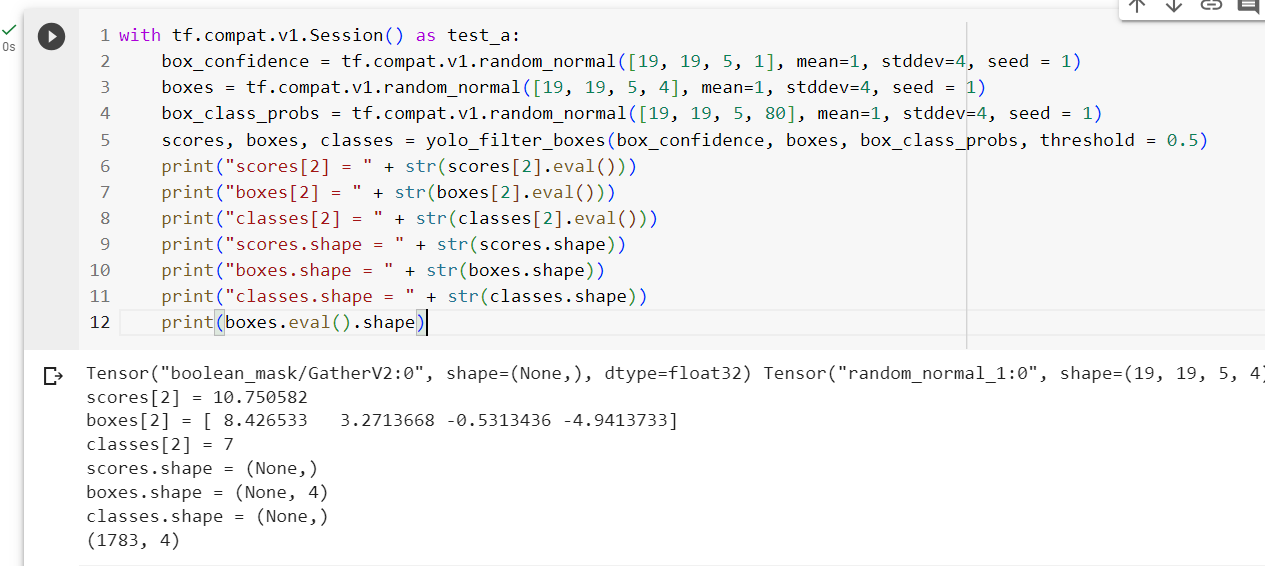
**Lab 04**

**Question 07**

1. In the below given cell, shape of the boxes.eval() is (1783,4). Why are there 1783 boxes? Explain the reason for it. What is the maximum number and minimum number you can get for that? Write these answers in a word file.
   * Change the values like mean and stddev in lines 2 and 4 as well as threshold value in line 5 and observe the different values you get for the boxes.eval().shape.

**Answer:**

The yolo\_filter\_boxes function filters the boxes using the confidence threshold, enabling one to cut out the low ones. Using the confidence threshold, the total number of boxes goes down to 1,783, all of them now having a confidence score of 0.5 or more. Confidence is influenced by the mean and the standard deviation; an increase in the mean or a decrease in the standard deviation may lead to an increase in confidence scores. It would also affect the parameters of class probability distribution since what varies is the threshold: The higher the threshold the lesser the number of boxes increase since this would require higher confidence.

I tried t with these values to see how that may alter the number of boxes: 1,805 is the maximum number of boxes possible, derived from this grid size and the quantity of anchors according to each cell. Then, with a grid of 19x19 based on 5 anchors per cell, that equates to 19 \* 19 \* 5 = 1,805. Theoretically, the minimum would be 0 in case no box ever makes the confidence threshold.

**Question 08**

yolo\_anchors.txt contains 10 values. They can be considered as height and width of 5 anchor boxes. What is the advantage of using such anchor boxes? What was the method used to determine the sizes of these anchor boxes? Give the answers to these questions in the word file.

**Answer**

Method used **-** K-Means clustering

1. Anchor boxes make the YOLO model faster by simplifying the way bounding boxes are predicted, which lowers the complexity in total.
2. This way, anchor boxes can use multiple objects at the same time in one grid cell, thus detecting multi-object situations more precisely and flexibly.
3. Assumed anchor boxes with varying aspect ratios help detect objects of different dimensions, making them identify a variety of shapes and sizes of objects.

**Question 10**

Download the output images zip file from the google drive and observe the bounding boxes in the autonomous driving dataset (i.e., 21 images from 0100.jpg to 0120.jpg). Select 2 images from these 21 images and,

* + Write what you observe regarding correctly detected objects, incorrectly detected objects, undetected objects and incorrect bounding boxes in the word file.
  + Include these output 2 images as well as the original 2 images in the word file.

**Answer**

Original Image

A street with a green light

Description automatically generated

Output Image

A street with a green light

Description automatically generated

* In the image, the bus is detected and has the correct bounding box. The truck, traffic lights, and jeep were completely missed. On the right side of the picture, there is also a car that went unidentified.

Original image

A road with trees on the side

Description automatically generated

Output image

A road with trees on the side

Description automatically generated

* Cars on the picture's left and right sides haven’t been identified.

**Question 11**

Adjusting parameters like max\_boxes, score\_threshold, and iou\_threshold of the yolo\_eval function can potentially address the limitations you noticed in step 10.

* + Change the max\_boxes [integer value] to a different value but use the original values for other 2 variables. Rerun the required cells to get the output images for the autonomous driving dataset. Observe if this result in improvement compared to step 10 for the same two images. If there are any improvements, write them in the word file. Include the new 2 output images in the word file.
  + Change the score\_threshold [value between 0-1] to a different value but use the original values for other 2 variables. Rerun the required cells to get the output images for the autonomous driving dataset. Observe if this result in improvement compared to step 10 for the same two images. If there are any improvements, write them in the word file. Include the new 2 output images in the word file.
  + Change the iou\_threshold [value between 0-1] to a different value but use the original values for other 2 variables. Rerun the required cells to get the output images for the autonomous driving dataset. Observe if this result in improvement compared to step 10 for the same two images. If there are any improvements, write them in the word file. Include the new 2 output images in the word file.

**Answer:**

* After increasing the **max\_boxes** value in the yolo\_eval function there are no changes.
* After decreasing the **score\_threshold** there was a change

**Picture 1**

****

* Output was increased it detected traffic lights, car, and truck but the truck was identified as a bus.

**Picture 2**

****

* Output has increased, and almost all the cars have been identified.
* No changes after increasing the **iou\_threshold** value in the yolo\_eval function
* When running the code to detect objects in all the autonomous driving dataset images within the images directory, I encountered an issue where it was trying to process something other than images—specifically, a “.ipynb\_checkpoints” file. This happened because I had kept the Colab session open, and it automatically saved a checkpoint. To resolve this, I modified the code to ignore that file.

new\_path = 'lab\_5/images'

for file\_name in os.listdir(new\_path):

**if file\_name.startswith('.'):**

**continue**

  # if file\_name[0] == '0': # images from [ 0100.jpg , 0120.jpg ]

  out\_scores, out\_boxes, out\_classes = predict\_(sess, file\_name)