This is the report for my yolo+bert model where I used yolo with darknet

Steps-

1. I cloned the darknet from Github
2. Made GPU and openCV enabled
3. ENabled the darknet

Our 1st part is done so now we will do functions

Functions-

1. I imported the necessary libraries
2. Imported CV and matplotlib for image visualization and manipulation
3. I defined the show image function for showing the image
4. The second function was for showing the image with annotations
5. I installed the transformers for tokenizing and BERT related tasks
6. I defined the next function for calculating the accuracies and other metrics
7. I installed pytessearct for doing OCR in python.
8. The next function was for box size manipulation
9. Defined recall, precision, and F1-scores
10. Now we have our OCR doing function in which first preprocessing then the image to text conversion
11. Then appending the results accordingly
12. Next was a class for tokenizing and iterating the dataset

mAP

This is the segment for seeing the actual accuracies, miss rates

1. This is the function for seeing the miss rate using precision, recall, and the number of images
2. When no detections return 0
3. If there are no common then throwing out the error
4. When the number is between 0 and 1
5. Now converting the files to results
6. As we have results so now showing out the boxes in the images
7. Adjusting the axes accordingly
8. Now showing the boxes
9. Defining the map functions

Now we have all our functions so now downloading the files

Downloading the files

1. Downloading the files and unzipping them
2. Unzipping bert
3. Open The text
4. Now we will use darknet and throwing the output in result.txt
5. Loading the result in json format
6. Making the tokenizer and trainer from transformers pretrained bert model
7. Now we will see the result on train dataset
8. The result is as follows-
   mAp- 68.16
   Precision- 0.85

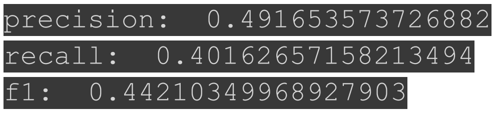
Recall- 0.78
F1 - 0.81

9. After seeing the results in the graph we have 1814 TP and 320 FP
10. Seeing the low miss and map in the graph format
11. Making the directories
12. As we have our model trained

So we will now see the results on test dataset

Detection on test

1. Running on test dataset
2. The results are as follows-
   map= 24.10%



```
precision:  0.491653573726882
recall:  0.40162657158213494
f1:  0.44210349968927903
```

3. Showing the ground truth
4. Final predictions on test dataset and visualizing the forms with boxes
5.