

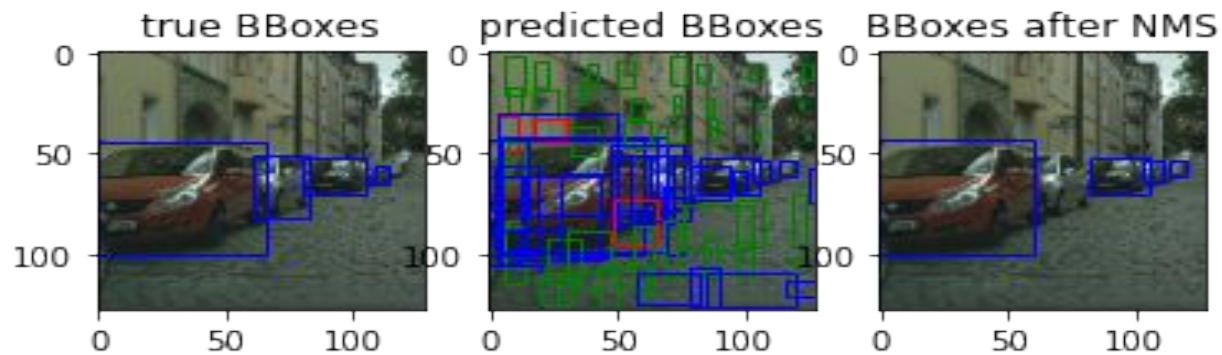
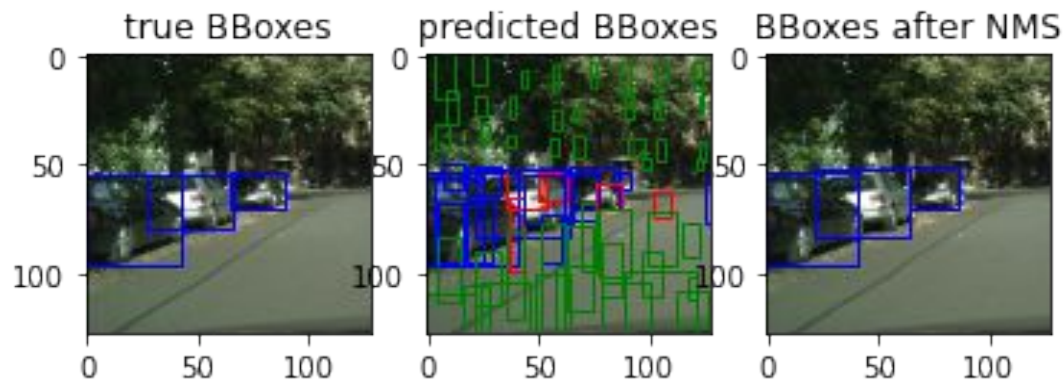
Non Max Suppression + Mean Average Precision

Implementation Walk through ~ 20 min

Non Max Suppression

```
for label in batch
    label[0,:,:]= 0 if label[0,:,:] < 0.6
    for each class
        grid_exists = find grid cells where max(label[5:8,gridx,gridy]) = class and label[0,gridx
,gridy]>0.6

    If len(grid_exists)>0:
        For each grid cell in grid_exists compute bounding box bbox and keep track of the corresponding
confidence score conf_score.
        For each bbox compute the iou with every other bbox. %% you can compute an iou matrix(iou_mat)
        bbox_group = iou_mat > 0.5
        max_conf_bbox_group = max(conf_score for each bbox_group).
        Now for each group keep the box with max_conf_bbox_group and make the confidence score of
every other box 0.
```



Mean Average Precision

- Mean average precision is computed over the entire dataset.
- You have define a yolo_evaluation function that computes the matches, score , total ground truths for each class.
- Example : Testdataset

```
for data in (test_loader):  
    input, target = data  
    predicted_out = yolo_net(input)  
  
    out_NMS=NMS(predicted_out)  
    match,scores,trues=yolo_evaluation(out_NMS,t  
target)  
    total_trues.append(trues)  
    for each class  
        match_values[class].append(match[class])  
        score_values[class].append(score[class])
```

```
AP=0  
cnt=0  
for each class:  
    if len(match_values[class])>0:  
        AP+=average_precision(match_values[class],scores  
_values[class],total_trues[class])  
        cnt++  
    if cnt>0:  
        mAp = AP/cnt
```

scores here are confidence scores

Average Precision

Compute average_precision for each class

```
average_precision(match_values, score_values, total_trues):  
    maximum_score= max(score)  
    ln= linspace(0.6, maximum_score, num=100)  
    precision_mat= zeros((101))  
    recall_mat= zeros((101))  
    for i, th in enumerate(ln):  
        matches= find match_values for which the score>th  
        TP= total(matches where matches = 1)
```

```
total_positive=total(matches)  
precision=1  
if total_positive>0:  
    precision=TP/total_positive  
recall=1  
if total_trues>0:  
    recall=TP/total_trues  
precision_mat[i]=precision  
recall_mat[i]=recall  
recall_mat[100]=0  
precision_mat[100]=1  
sorted_ind=np.argsort(recall_mat)  
sorted_recall=recall_mat[sorted_ind]  
sorted_precision=precision_mat[sorted_ind]  
  
area=auc(sorted_recall, sorted_precision)  
return area
```

Good tutorial for mAP

https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173

TA- assignment link -

https://docs.google.com/spreadsheets/d/1Ls9eOFbvaxlYnOS_TmROW1qIOulj9k2tO-zHFAoXryo/edit?usp=sharing

Go to the assigned TA's debug channel on discord.

WorkBook Link - <https://colab.research.google.com/drive/1X1HNYvhU1PXbgoaBbAkPTBgPVsTBFz1C?usp=sharing>

Settings



General



Video



Audio



Share Screen



Chat



Virtual Background



Recording



Profile



Statistics



Feedback



Keyboard Shortcuts



Accessibility

Speaker

Test Speaker

Same as System

Output Level:

Output Volume:



Microphone

Test Mic

Same as System

Input Level:

Input Volume:



☒ Automatically adjust microphone volume

☐ Use separate audio device to play ringtone simultaneously

☐ Join audio by computer when joining a meeting

☐ Mute microphone when joining a meeting

☐ Press and hold SPACE key to temporarily unmute yourself

Advanced