

Orbital_insight-NMS

June 6, 2019

```
In [1]: #import libraries
import pandas as pd
import numpy as np

def NMS(filename,Thre):
    #check all the anchors are +
    #check if the corner of start < corner end
    #remove all the box with less than 50% chance
    # have a processed list of index which shows which index has been processed-- check
    #pick the box with highest score-- call it pilote --check if it is not in {main_box_id}
    #find the box with high overlap with the box and add their index to suppress_box_idx
    #continue until no_process is empty

    #reading the bounding box data into dataframe
    df = pd.read_csv(filename ,header= None)

    df = df[df.iloc[:,-1] > 0.5] # only consider boxes with higher than chance prob
    df = df[df.iloc[:,2]>0]      # check if w is positive
    df = df[df.iloc[:,3]>0]      #check all h are positive

    df = np.asarray(df)

    main_box_id = []
    m,n = df.shape

    while m > 0 : # as long as there are boxes that have not been detected as max or ov

        overlap = []
        bestbox = df[:, -1].argmax() #find the index of the box whit highest probability

        overlap.append(bestbox) # createa list of box which is processed as max box or ov
        main_box_id.append(df[bestbox,:]) # create a list of max boxes as the final output

        #get the values of starting and ending point of the max box and clac it's area
        x1 = df[bestbox,0]
        y1 = df[bestbox,1]
        x2 = x1 + df[bestbox,2]
```

```

y2 = y1 + df[bestbox,3]
area_main_box = (x2-x1)*(y2-y1)

# for all other boxes in the list , go through one by one and calc their IOU with
for i in range(m):
    x3 = df[i,0]
    y3 = df[i,1]
    x4 = x3 + df[i,2]
    y4 = y3 + df[i,3]

    #get the coordinates of the intersection box
    Xs=max(x1,x3)
    Ys=max(y1,y3)
    Xe=min(x2,x4)
    Ye=min(y2,y4)

    #calc area of the box
    area_i_box = (x4-x3)*(y4-y3)

    #calc the intersection area
    intersection = (Xe-Xs)*(Ye-Ys)
    #calc IOU
    IOU = intersection /(area_main_box + area_i_box - intersection)

    #keep boxes which have higher than threshold IOU
    if ((IOU)) > Thre :
        overlap.append(i)
# after processing all the boxes and calc the IOU of them with the max box, suppress
# dataframe
df = np.delete(df,overlap,axis=0)
#update size of the new dataframe for the while loop check
m,n=df.shape

print(main_box_id)
main_box_id = pd.DataFrame(main_box_id)
#output can be written in any format
main_box_id.to_csv('output_boxes_csv.txt')
#     main_box_id.to_csv('output_boxes.txt', sep='\t')
return

```

```
In [2]: NMS('orbital',0.4)
```

```

# this is the most straight forward implimentation though it could be better if we sort
#the beginning and doing some tricks to speed up the algorithms

```

```
[array([10. , 11. , 20. , 20. , 0.8]), array([42. , 40. , 12. , 20. , 0.7])]
```