import matplotlib.image as mpimg

import cv2

import os

f1=os.listdir("ideal")

avgiou=0

avgprec=0

avgrec=0

i=0

print("image iou precision recall")

for imgname in f1:

path1="ideal/"+imgname

path2="output3/"+(imgname.replace("gtFine\_color.png","leftImg8bit.png"))

path3="Resources/"+(imgname.replace("gtFine\_color.png","leftImg8bit.png"))

img1=cv2.imread(path1,0)

img2=cv2.imread(path2,0)

img3 = cv2.imread(path3,0)

I = 0

U = 0

MO=IO=0

for j in range(0, img1.shape[0]):

for k in range(0, img1.shape[1]):

if img1[j, k] == 255 and img2[j, k] == 255:

I = I + 1

if img1[j, k] == 255 or img2[j, k] == 255:

U = U + 1

if img2[j,k]==255:

MO=MO+1

if img1[j,k]==255:

IO=IO+1

IoU=I/U

prec=I/MO #intersection(true positives)/my predictions(true positives+false positives)

rec=I/IO #intersection(true positives)/ideal predictions(true positives+true negatives)

print(imgname,IoU,prec,rec)

#print(IoU)

avgiou=avgiou+IoU

avgprec = avgprec + prec

avgrec = avgrec + rec

img0 = cv2.hconcat((img1, img2))

img5 = cv2.hconcat((img3, img3))

img4 = cv2.vconcat((img0, img5))

i=i+1

cv2.imshow("img" + str(i), img4)

avgiou=avgiou/23

avgprec=avgprec/23

avgrec=avgrec/23

print("average: ",avgiou,avgprec,avgrec)

cv2.waitKey(0)

image iou precision recall

dusseldorf\_000114\_000019\_gtFine\_color.png 0.7261142962825966 0.7831993456881247 0.9087773714179899

dusseldorf\_000194\_000019\_gtFine\_color.png 0.792578073527474 0.844444834424302 0.9280782286684154

tubingen\_000143\_000019\_gtFine\_color.png 0.7998711252797938 0.8796434432343727 0.8981682470771926

krefeld\_000000\_005252\_gtFine\_color.png 0.6406491724802241 0.7387660304958094 0.8282889274662238

erfurt\_000033\_000019\_gtFine\_color.png 0.6687636477071852 0.7361924566457447 0.8795414140493231

hamburg\_000000\_046078\_gtFine\_color.png 0.6988258750553833 0.8205990560778922 0.8248445116826357

hamburg\_000000\_069177\_gtFine\_color.png 0.6781235776951207 0.8123331453280692 0.8040941658137154

jena\_000117\_000019\_gtFine\_color.png 0.7527635136216025 0.8244590451160754 0.8964413624933323

strasbourg\_000001\_027097\_gtFine\_color.png 0.5769744123966404 0.7039140536568899 0.7618750806000946

hanover\_000000\_055124\_gtFine\_color.png 0.6572817281728173 0.9396505134210053 0.6862512921717884

dusseldorf\_000191\_000019\_gtFine\_color.png 0.7405987812417204 0.8410020860077022 0.8611772921459087

dusseldorf\_000093\_000019\_gtFine\_color.png 0.722131431684247 0.8313410995610118 0.8460854092526691

erfurt\_000098\_000019\_gtFine\_color.png 0.6790546603021929 0.736220549037919 0.8973866439005881

tubingen\_000138\_000019\_gtFine\_color.png 0.7693293226381461 0.8560006611160466 0.8836966258584652

erfurt\_000014\_000019\_gtFine\_color.png 0.8014383274197856 0.9130283603172878 0.8676786451879389

monchengladbach\_000000\_009615\_gtFine\_color.png 0.6917497556207234 0.8100689118340622 0.8256638820180147

dusseldorf\_000094\_000019\_gtFine\_color.png 0.6267704454880536 0.7747403964013969 0.7664447342530945

tubingen\_000014\_000019\_gtFine\_color.png 0.8656326781326781 0.9776698608160257 0.8830924685700857

tubingen\_000025\_000019\_gtFine\_color.png 0.8111376638424216 0.9354434608318022 0.8592359619515189

monchengladbach\_000000\_023856\_gtFine\_color.png 0.6729292030261401 0.8173870126207431 0.7919978138592216

zurich\_000034\_000019\_gtFine\_color.png 0.6472565991844911 0.7678598706655125 0.8047244654736917

aachen\_000064\_000019\_gtFine\_color.png 0.7382000949558782 0.8727209383346882 0.8272635538024035

bremen\_000156\_000019\_gtFine\_color.png 0.7740195297384654 0.8867014341590613 0.8589724158836011

Average IoU 0.7187910398040774

Average Precision 0.8305820245996323

Average Recall 0.8430339353738223