

```
[3] #           x,      y,      w,      h,      c
    gt = [[10,      0,     100,     100,      1], #A
           [200,    300,     95,     200,      1], #B
           [1000,   500,    250,     300,      1], #C
           [700,    500,     98,     112,      1], #D
           [690,   212,    102,     108,      1]] #E

#           x,      y,      w,      h,      c,      conf
pred = [[710,    530,     83,     92,      1,     0.4], #D
         [653,    250,     79,     94,      1,     0.6], #E (0.146177750155376)
         [205,    310,    100,    196,      1,     0.7], #B
         [1020,   490,    200,    312,      1,     0.5], #C
         [123,    463,    335,   1023,      1,     0.8], #B (0.055441268729665455)
         [463,    230,   1020,    350,      1,     0.3], #E(0.030857142857142857) C(0.0176678445229682)
         [5,       7,     101,    103,      1,     0.9]] #A
```

```
[4] #https://stackoverflow.com/a/57247833
from shapely.geometry import Polygon

def calculate_iou(box_1, box_2):
    poly_1 = Polygon(box_1)
    poly_2 = Polygon(box_2)
    iou = poly_1.intersection(poly_2).area / poly_1.union(poly_2).area
    return iou
```

```
#[pair(i,j), conf(j), iou(i,j)]
tp_fp = []
for i, tru in enumerate(gt):
    for j, pr in enumerate(pred):
        if i==0 and j==6 or i==1 and j==2 or i==2 and j==3 or i==3 and j==0 or i==4 and j==1:
            tp_fp.append([(i+1,j+1), pr[5], 'TP' if calculate_iou([[tru[0],tru[1]],[tru[0]+tru[2],tru[1]],[tru[0]+tru[2],tru[1]-tru[3]]],[tru[0],tru[1]-tru[3]]], [[pr[0],pr[1]],[pr[0]+pr[2],pr[3]]])])
tp_fp.append([(2,5),pred[4][5],'FP'])
tp_fp.append([(5,6),pred[5][5],'FP'])
tp_fp
```

```
[(1, 7), 0.9, 'TP'],
[(2, 3), 0.7, 'TP'],
[(3, 4), 0.5, 'TP'],
[(4, 1), 0.4, 'FP'],
[(5, 2), 0.6, 'FP'],
[(2, 5), 0.8, 'FP'],
[(5, 6), 0.3, 'FP']]
```

```
#sort reverse by conf
from operator import itemgetter
tp_fp_sorted = sorted(tp_fp, key=itemgetter(1), reverse=True)
tp_fp_sorted
```

```
[(1, 7), 0.9, 'TP'],
[(2, 5), 0.8, 'FP'],
[(2, 3), 0.7, 'TP'],
[(5, 2), 0.6, 'FP'],
[(3, 4), 0.5, 'TP'],
[(4, 1), 0.4, 'FP'],
[(5, 6), 0.3, 'FP']]
```

```
#add accTP and accFP
##[pair(i,j), conf(j), iou(i,j), accTP, accFP]
accTP=0
accFP=0
for row in tp_fp_sorted:
    if row[2]=='TP':
        accTP = accTP + 1
    elif row[2]=='FP':
        accFP = accFP + 1
    row.append(accTP)
    row.append(accFP)
```

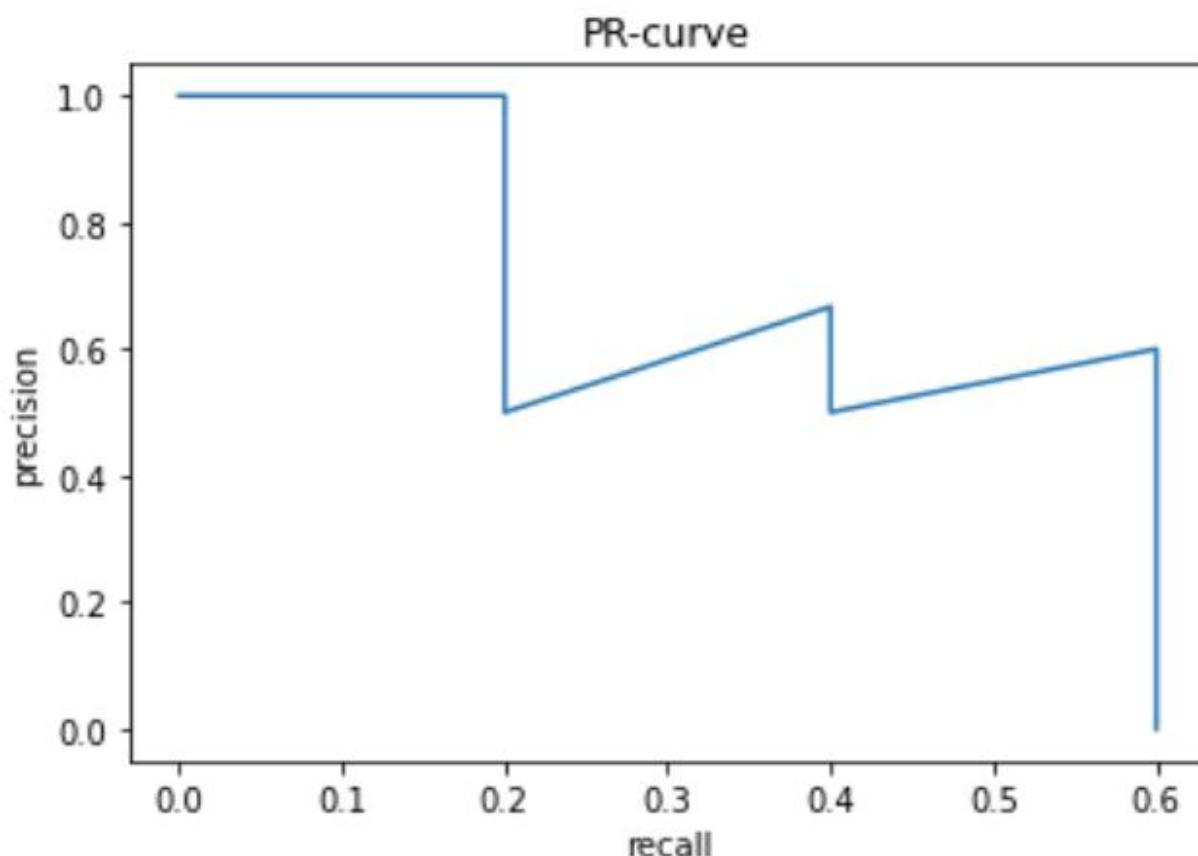
```
tp_fp_sorted
```

```
[(1, 7), 0.9, 'TP', 1, 0],
 [(2, 5), 0.8, 'FP', 1, 1],
 [(2, 3), 0.7, 'TP', 2, 1],
 [(5, 2), 0.6, 'FP', 2, 2],
 [(3, 4), 0.5, 'TP', 3, 2],
 [(4, 1), 0.4, 'FP', 3, 3],
 [(5, 6), 0.3, 'FP', 3, 4]]
```

```
#calc precision and recall
##[pair(i,j), conf(j), iou(i,j), accTP, accFP, precision, recall]
for row in tp_fp_sorted:
    row.append(row[3]/(row[3]+row[4]))
    row.append(row[3]/5)
tp_fp_sorted
```

```
[(1, 7), 0.9, 'TP', 1, 0, 1.0, 0.2],
 [(2, 5), 0.8, 'FP', 1, 1, 0.5, 0.2],
 [(2, 3), 0.7, 'TP', 2, 1, 0.6666666666666666, 0.4],
 [(5, 2), 0.6, 'FP', 2, 2, 0.5, 0.4],
 [(3, 4), 0.5, 'TP', 3, 2, 0.6, 0.6],
 [(4, 1), 0.4, 'FP', 3, 3, 0.5, 0.6],
 [(5, 6), 0.3, 'FP', 3, 4, 0.42857142857142855, 0.6]]
```

```
import matplotlib.pyplot as plt
x_ = [0]
y_ = [1]
for row in tp_fp_sorted:
    x_.append(row[6])
    y_.append(row[5])
x_.append(tp_fp_sorted[-1][6])
y_.append(0)
plt.plot(x_, y_)
plt.xlabel('recall')
plt.ylabel('precision')
plt.title('PR-curve')
plt.show()
```

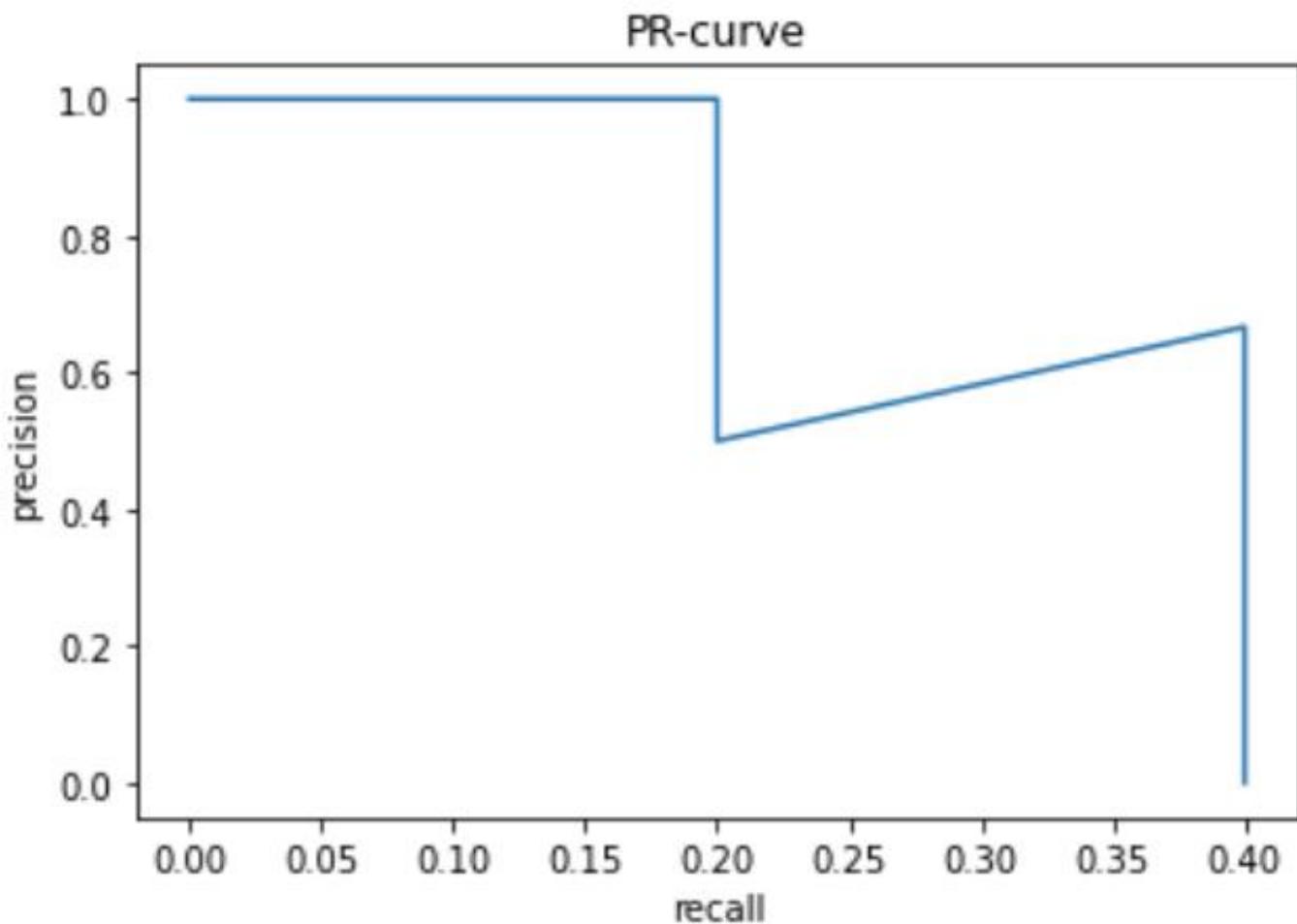


```
import numpy as np
ap05 = np.trapz(y_, x_)
print(ap05)
```

0.4266666666666664

```
x_ = [0]
y_ = [1]
for row in tp_fp_sorted:
    x_.append(row[6])
    y_.append(row[5])
x_.append(tp_fp_sorted[-1][6])
y_.append(0)
plt.plot(x_, y_)
plt.xlabel('recall')
plt.ylabel('precision')
plt.title('PR-curve')
plt.show()
```

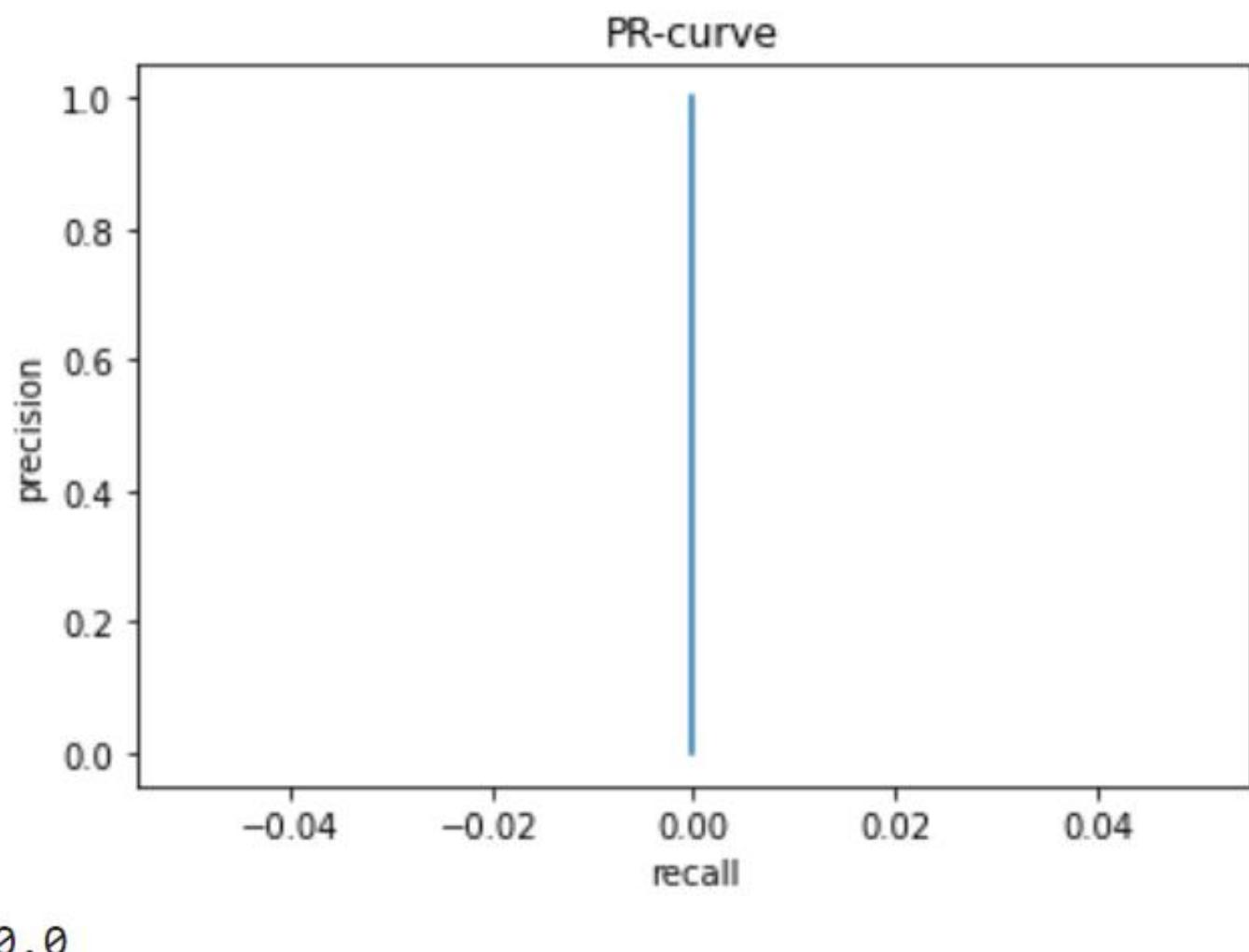
```
ap075 = np.trapz(y_, x_)
print(ap075)
```



0.31666666666666665

```
x_ = [0]
y_ = [1]
for row in tp_fp_sorted:
    x_.append(row[6])
    y_.append(row[5])
x_.append(tp_fp_sorted[-1][6])
y_.append(0)
plt.plot(x_, y_)
plt.xlabel('recall')
plt.ylabel('precision')
plt.title('PR-curve')
plt.show()
```

```
ap09 = np.trapz(y_, x_)
print(ap09)
```



```

def func(gt, pred, iou):
    tp_fp = []
    for i, tru in enumerate(gt):
        for j, pr in enumerate(pred):
            if i==0 and j==6 or i==1 and j==2 or i==2 and j==3 or i==3 and j==0 or i==4 and j==1:
                tp_fp.append([(i+1,j+1), pr[5], 'TP' if calculate_iou([[tru[0],tru[1]],[tru[0]+tru[2],tru[1]],[tru[0]+tru[2],tru[1]-tru[3]],[tru[0],tru[1]-tru[3]]], [[pr[0],pr[1]],[pr[0]+pr[2],pr[1]]]) >= iou else 'FP'])
            tp_fp.append([(2,5),pred[4][5], 'FP'])
    tp_fp.append([(5,6),pred[5][5], 'FP'])

    tp_fp_sorted = sorted(tp_fp, key=itemgetter(1), reverse=True)

    accTP=0
    accFP=0
    for row in tp_fp_sorted:
        if row[2]=='TP':
            accTP = accTP + 1
        elif row[2]=='FP':
            accFP = accFP + 1
        row.append(accTP)
        row.append(accFP)

    for row in tp_fp_sorted:
        row.append(row[3]/(row[3]+row[4]))
        row.append(row[3]/5)

    return max([row[6] for row in tp_fp_sorted])

```

```

IoU = [0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1]
R = []
for i in IoU:
    R.append(func(gt, pred, i))
R

```

```
[0.6, 0.6, 0.6, 0.6, 0.6, 0.4, 0.2, 0.0, 0.0, 0.0]
```

```
from sklearn import metrics  
AR = 2 * (metrics.auc(IoU, R))
```

AR

0.3299999999999996