

Introduction to Artificial Intelligence

Homework 1 Report

109550127 宋哲頤

1.Screenshot of your code and brief explanation

Part 1 screenshot:

```
# Begin your code (Part 1)
dataset=[]
pathc=dataPath+'/car'
pathnoc=dataPath+'/non-car'
for filename in os.listdir(pathc):
    img = cv2.imread(os.path.join(pathc,filename))
    img = cv2.resize(img, (36, 16), interpolation=cv2.INTER_AREA)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    if img is not None:
        dataset.append((img,1))
for filename in os.listdir(pathnoc):
    img = cv2.imread(os.path.join(pathnoc,filename))
    img = cv2.resize(img, (36, 16), interpolation=cv2.INTER_AREA)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    if img is not None:
        dataset.append((img,0))
#raise NotImplementedError("To be implemented")
# End your code (Part 1)
return dataset
```

在 part1 的部分我分別做兩次迴圈，一次輸入有車的圖，一次沒車的，並用 tuple 一起存入圖片跟標記 append 到 dataset 裡。

Part 2 screenshot:

```
# Begin your code (Part 2)
bestError=1.0;pos=0.0;nes=0.0;pot=0.0;net=0.0;tmp=0.0
featuremap=[]
for i in range(len(weights)):
    if labels[i]==1:
        pot+=weights[i]
    else:
        net+=weights[i]
sum=float(np.sum(weights))
for i in range(len(features)):
    featuremap.clear()
    pos=0.0
    nes=0.0
    for j in range(len(featureVals[i])):
        featuremap.append((featureVals[i][j],labels[j],weights[j]))
        featuremap.sort(key = lambda x: x[0])
    for j in range(len(featureVals[i])):
        threshold=featuremap[j][0]
        if j==0:
            continue

        if featuremap[j-1][1]==1:
            pos+=featuremap[j-1][2]
        else:
            nes+=featuremap[j-1][2]
        tmp=min(pos+(net-nes),nes+(pot-pos))
        if bestError > tmp/sum:
            bestError=tmp/sum
            if tmp==nes+(pot-pos):
                bestClf=WeakClassifier(features[i],threshold,1)
            else :
                bestClf=WeakClassifier(features[i],threshold,-1)
# raise NotImplementedError("To be implemented")
# End your code (Part 2)
# print(bestError)
return bestClf, bestError
```

在選擇 bestClf 跟 bestError 時我先 sort 所有的特徵值並且以前綴和在 featurevalues 的其中一個 row 中掃過每一個特徵值作為 threshold 尋找最小的 error。

Part 3 screenshot:

```
• ✓ for i in range (1,11):
    print('Start training your classifier')
    clf = adaboost.Adaboost(T=i)
    clf.train(trainData)
    print('\nEvaluate your classifier with training dataset')
    utils.evaluate(clf, trainData)
    print('\nEvaluate your classifier with test dataset')
    utils.evaluate(clf, testData)
```

將在 main.py 裡的 adaboost.Adaboost() 傳入 1~10 的 T 值

Part 4 screenshot:

```
# Begin your code (Part 4)
gif=cv2.VideoCapture('data/detect/video.gif')
f1=open('Adaboost_pred.txt','w')
while True:
    #ret,frame = gif.read()
    ret,frame = gif.read()
    if ret==False:
        break
    f=open(dataPath,'r')
    for line in f.readlines():
        s=line.split(' ')
        if int(s[0])!=76:
            continue
        parking_space_image=crop(int(s[0]),int(s[1]),int(s[2]),int(s[3]),int(s[4]),int(s[5]),int(s[6]),int(s[7]),frame)
        reframe=cv2.resize(parking_space_image,[36,16],interpolation=cv2.INTER_AREA)
        reframe = cv2.cvtColor(reframe, cv2.COLOR_BGR2GRAY)
        if clf.classify(reframe)==1:
            frame=draw_lines(frame,int(s[0]),int(s[1]),int(s[2]),int(s[3]))
            frame=draw_lines(frame,int(s[0]),int(s[1]),int(s[4]),int(s[5]))
            frame=draw_lines(frame,int(s[4]),int(s[5]),int(s[6]),int(s[7]))
            frame=draw_lines(frame,int(s[2]),int(s[3]),int(s[6]),int(s[7]))
            f1.write('1 ')
        else:
            f1.write('0 ')
    f1.write('\n')
    cv2.namedWindow('img', cv2.WINDOW_NORMAL)
    cv2.resizeWindow("img",1280 , 800)
    cv2.imshow("img",frame)
    cv2.waitKey(1)
    cv2.destroyAllWindows()
# raise NotImplementedError("To be implemented")
# End your code (Part 4)
```

```
def draw_lines(img,x1,y1,x2,y2):
    start_point = (x1, y1)
    end_point = (x2, y2)
    color = (0, 255, 0) # green
    thickness = 1 # 寬度
    cv2.line(img, start_point, end_point, color, thickness)
```

將 gif 的每一幀進行處理，先 crop 和 resize 之後用 classify 判斷有沒有車，有車就在圖上畫上綠框。

Part 5 screenshot:

```
# Begin your code (Part 5)
gif=cv2.VideoCapture('/content/yolov5/HW1_material/detect/video.gif')
fl=open('Yolov5_pred.txt','w')
while True:
    #ret,frame = gif.read()
    ret,frame = gif.read()
    if ret==False:
        break
    f=open('/content/yolov5/HW1_material/detect/detectData.txt','r')
    for line in f.readlines():
        s=line.split(' ')
        if int(s[0])==76:
            continue
        parking_space_image=crop(int(s[0]),int(s[1]),int(s[2]),int(s[3]),int(s[4]),int(s[5]),int(s[6]),int(s[7]),frame)
        reframe=cv2.resize(parking_space_image,(36,16),interpolation=cv2.INTER_AREA)
        #reframe=cv2.cvtColor(reframe,cv2.COLOR_BGR2GRAY)
        if yolov5_func.classify(reframe,weight_path,confidence_threshold,(36,16))==1:
            frame=draw_lines(frame,int(s[0]),int(s[1]),int(s[2]),int(s[3]))
            frame=draw_lines(frame,int(s[0]),int(s[1]),int(s[4]),int(s[5]))
            frame=draw_lines(frame,int(s[4]),int(s[5]),int(s[6]),int(s[7]))
            frame=draw_lines(frame,int(s[2]),int(s[3]),int(s[6]),int(s[7]))
            fl.write('1 ')
        else:
            fl.write('0 ')
    fl.write('\n')
    cv2.imwrite('frame.png', frame)
    break
```

做法和 part4 相似，crop 傳入值不同

2.Your implementation of the above requirements in detail

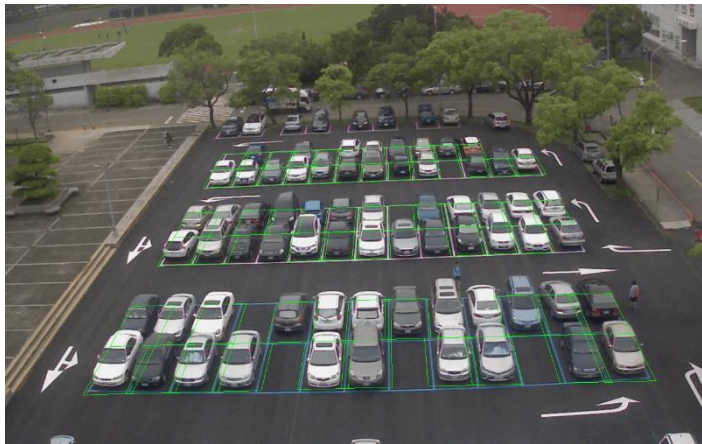
在這次作業主要的 Adaboost 的 selectBest 中我先計算所有有車樣本的權重 T+ 和所有沒有車的樣本 T-，然後使用兩個迴圈，外層的是 featurevalues 的每一個 row，每個 row 代表的是一組 dataset，在 featurevalues 每個 column 是每個不同的 feature 對同樣一組的 dataset 所計算出來的特質值，而我的作法是在每個 row 掃過一次，先 sort 後枚舉每個特徵值作為 threshold 並且逐一計算錯誤率，而我設每個 threshold 的前面有車樣本權重和為 S+，沒有的為 S-，然後因為 polarity 有分 1 和 -1，選大於 threshold 為有車或選小於 threshold 為有車，因此我運用

$$E=\min("S+" + ("T-" - "S-"), "S-" + ("T+" - "S+"))$$

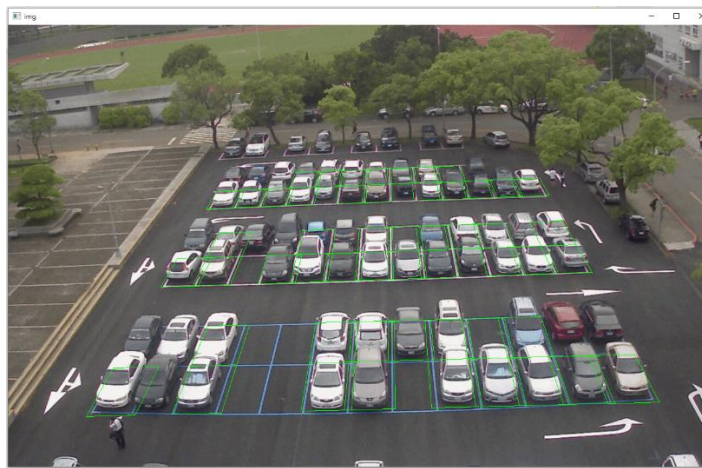
於是，通過把這格排序的表掃描從頭到尾一遍就可以為弱分類器選擇使分類誤差最小的 threshold，也就是選擇一個最佳的特徵值。

在 detection.py 中，我每次處理一偵，將 detectData.txt 一行一行讀進來，因 dataset 的停車位置順序跟 txt 檔一樣，所以我先 crop、resize、cv2.COLOR_BGR2GRAY 來處理圖片，再用已經 train 好的 clf 來偵測每一張 data 是否有車，用 clf.classfy 來判斷，是的話回傳 1 並且用 draw_lines() 傳入座標來畫綠色方框，同時 print "1" 或 "0" 進入 Adaboost_pred.txt 內

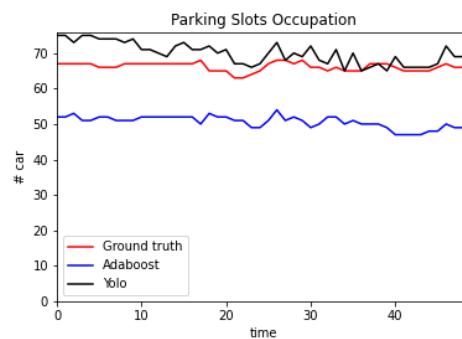
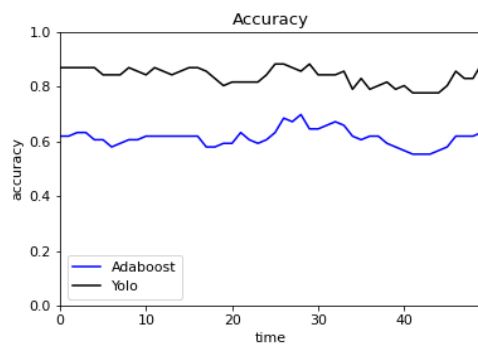
3. Discuss what you observed with accuracy, F1-score and parking slots occupation plot of different methods in the report.



Yolov5



Adaboost



首先，關於 part3 的 T=1~10 所推出的 accuracy:

T1: Accuracy: 532/600 (0.886667)

T2: Accuracy: 532/600 (0.886667)

T3: Accuracy: 573/600 (0.955000)

T4: Accuracy: 573/600 (0.955000)

T5: Accuracy: 579/600 (0.965000)

T6: Accuracy: 581/600 (0.968333)

T7: Accuracy: 578/600 (0.963333)

T8: Accuracy: 584/600 (0.973333)

T9: Accuracy: 585/600 (0.975000)

T10: Accuracy: 585/600 (0.975000)

由我在本機上跑的數據可以知道當 T 越大，隨著不同的 clf 使 weight 權重不斷的改變，能使準確率越高

在實做 adaboost 後比較 yolov5，從 Accuracy 圖可看出 yolov5 的準確率是大於 adaboost 的，從 Parking Slots Occupation 中也可看出 yolov5 的曲線較靠近 Ground truth，我認為因 yolov5 本身的幾個優點讓準確率提升：

(1) Mosaic 資料增強

(2) 自適應錨框計算：將計算初始描框的值遷入程式碼中，每次訓練時，自適應的計算不同訓練集中的最佳錨框值。

(3) 自適應圖片縮放：再縮放圖片時因圖片大小不同造成填充的黑邊不同，而黑邊填充過多會使要在資訊很多餘，影響推理速度，因此對原始影像自適應的新增最少的黑邊，影像高度上兩端的黑邊變少了，在推理時，計算量也會減少，即目標檢測速度會得到提升。

在 F1Score 中以 T=10 的 adaboost 為

precision:0.96721311、recall:0.983333=>F1-score:0.97493436

在 yolov5 為

precision:0.97993311、recall:0.976667=>F1-score:0.97829686

從 F1-score 可看出其兩個做法的分數相差不遠，因本身的 precision 跟 recall 的分數就相差無幾，因此 F1-score 的表現才如此相近。

4. Describe problems you meet and how you solve them

(1) 在實作 Adaboost 時，我一開始寫出了一個三個迴圈的版本，單純枚舉每個 row 的特徵值當 threshold 然後在枚舉一次判斷有車或沒車，其複雜度為 $O(n^3)$ ，當在跑 part3 時速度過慢，跑了三個小時還都跑不到 1/10，因此我決定試著用 $O(n^2)$ 的前綴和做法，三個小時就跑完了 T 從 1 到 10。

(2) 在 yolov5 的 detection 中

```
reframe = cv2.cvtColor(reframe, cv2.COLOR_BGR2GRAY)
```

如果將圖片變成灰階將會出現 bug，我查了一下資料發現是 yolov5_function.py 裡的 `img.transpose()` 出現問題，因為如果先轉換灰階會對圖像顏色的矩陣進行修改，這樣會造成 `img.transpose()` 無法執行

```
# Convert  
img = img.transpose((2, 0, 1))[:, :, -1] # HWC to CHW, BGR to RGB
```

後來我試著將 `cv2.cvtColor()` 給移除掉就可以順利跑了。