

Introduction to AI HW1 Report

109550025 謝翔丞

1. Screenshot of your code and brief explanation
2. Your implementation of the above requirements in detail

簡單概述，adaboost的概念是前一個弱分類器(weakclassifier)分錯的樣本會得到加強，而全部的樣本會在加權後再次被用來訓練下一個弱分類器，可以簡單分為以下步驟。

原先所有的樣本都會被賦予相等的權重比，接著開始迭代，如果某個樣本在訓練時被歸類錯誤，那麼下一次訓練時他的權重就會增加，反之若是訓練正確則會降低。

而adaboosty最重要的一步就是加大誤差率小的弱分類器並降低誤差率大的弱分類器的權重，最後結合成一個較為準確的強分類器。

Snapshot of Part1.

這個部份我是先建立空的list，用來存取讀完照片後的tuple;

接下來先用os.listdir取得datapath的子目錄，分別有car跟non-car，然後對於所有在子目錄裡面的子目錄再用os.listdir取得路徑，並再從裡面取得所有內容(for iptfile in iptfiles)，也就是照片。

下一步就是給定tuple的第二個參數，只要是從"car"資料夾內存取的圖片就給定label = 1，反之則是0，並且用cv2.resize將360*160的照片改成36*16再放入tuple1並append進一開始創立的dataset，就完成這個part。

```
def loadImages(dataPath):
    """
    Load all Images in the folder and transfer a list of tuples.
    The first element is the numpy array of shape (m, n) representing the image.
    (remember to resize and convert the parking space images to 36 x 16 grayscale images.)
    The second element is its classification (1 or 0)
    Parameters:
    | dataPath: The folder path.
    Returns:
    | dataset: The list of tuples.
    """
    # Begin your code (Part 1)
    dataset = []

    subfolders = os.listdir(dataPath)
    for subfolder in subfolders:
        subPath = dataPath + '/' + subfolder
        iptfiles = os.listdir(subPath)

        for iptfile in iptfiles:
            imgPath = subPath + '/' + iptfile
            img = cv2.imread(imgPath)

            if(subfolder == 'car'):
                label = 1
            elif(subfolder == 'non-car'):
                label = 0

            img_1 = cv2.resize(img, (36,16), interpolation = cv2.INTER_AREA)

            img_2 = cv2.cvtColor(img_1, cv2.COLOR_BGR2GRAY)
            tuple = (img_2, label)
            dataset.append(tuple)

    #raise NotImplementedError("To be implemented")
    # End your code (Part 1)

    return dataset
```

Snapshot of Part2

這裡的的第一步是先initialize bestError跟bestClf，接著對於每個feature j 訓練一個 classifier，而後者的結果受一個單一的Haarlike feature所限制。

下一段經過計算error之後，跟當前的bestError做比較，如果當前error小於bestError的話就更新bestError值，並選擇此classifier作為bestClf。

```
def selectBest(self, featureVals, iis, labels, features, weights):
    """
    Finds the appropriate weak classifier for each feature.
    Selects the best weak classifier for the given weights.
    Parameters:
        featureVals: A numpy array of shape (len(features), len(dataset)).
                     Each row represents the values of a single feature for each training sample.
        iis: A list of numpy array with shape (m, n) representing the integral images.
        labels: A list of integer.
               The ith element is the classification of the ith training sample.
        features: A numpy array of HaarFeature class.
        weights: A numpy array with shape(len(dataset)).
                 The ith element is the weight assigned to the ith training sample.
    Returns:
        bestClf: The best WeakClassifier Class
        bestError: The error of the best classifier
    """

    # Begin your code (Part 2)
    bestError = 1
    bestClf = WeakClassifier(features[0])

    for j in range(len(features)):

        tmperror = 0
        feature_classifier = WeakClassifier(features[j])

        for i in range(len(weights)):
            value = feature_classifier.classify(iis[i])
            tmperror += (weights[i]* abs(value - labels[i]))

        if(tmperror < bestError):
            bestError = tmperror
            bestClf = feature_classifier

    #raise NotImplementedError("To be implemented")
    # End your code (Part 2)
    return bestClf, bestError
```

Snapshot of Part4

這裡首先我先建立一個imglist目的是想說因為gif有50幀，所以想把每個frame都存進去，但後來發現只要秀第一張，所以這部應該是可省略。然後開啟一個adaboost_pred用來儲存每次判斷的結果以跟yolov5做比較。

接下來的步驟我將解釋寫在code註解上 方便比對。

```
# Begin your code (Part 4)
imglist=[]
ff=open("adaboost_pred.txt","w")
gifPath = 'data/detect/video.gif'

#from here i read the gif by PIL instead of cv2
#because according to the source i searched on the internet
#it refer that cv2 can't read gif(though i foun dsomebody successfully used it La
frame2 = Image.open(r"data/detect/video.gif")
numframes = 0
img = cv2.cvtColor(np.asarray(frame2),cv2.COLOR_RGB2BGR)
#i save the first frame as "img.png" in order to draw bounding_box on it
frame2.save("img.png")

numframes = 0
#A While loop to go through all the 50 frames
while frame2:
    new_img=cv2.imread("img.png")
    try:

        numframes += 1
        #frame2.seek(x) means the xth frame of frame2.gif
        frame2.seek(numframes)

        frame = frame2.copy()

    with open(dataPath, 'r') as f:
        #first step we read the number which means the num of parking spaces
        num = int(f.readline())
        #there are 76 plots means we will do th eloop 76 times
        for i in range(num):
            #read the four point pairs
            data = f.readline()
            #save the points in x1~x4 and y1~y4 with splitting them by a space before
            list2 = data.split(' ')
            x1,y1,x2,y2,x3,y3,x4,y4 = (int)(list2[0]),(int)(list2[1]),(int)(list2[2]),
            #convert the PIL type img to the cv2 type and convert it to the gray scale
            cvframe = cv2.cvtColor(np.asarray(frame),cv2.COLOR_RGB2GRAY)
            #crop the image by four point pairs by the function that TA gave
            cvframe_2 = crop(x1,y1,x2,y2,x3,y3,x4,y4,cvframe)
            #resize the ing to fit theclasify function
            new_cvframe = cv2.resize(cvframe_2,(36,16),interpolation=cv2.INTER_AREA)
```

```
        #use classifier to classify each plot image
        #if the return value is 1
        #then draw the green bounding_box on the border of the plot in the "new_img"
        #which is read from "img.png" we saved before
        #also write "1 " onto the adaboost_pred.txt
        if(clf.classify(new_cvframe)):
            ff.write('1 ')
            cv2.line(new_img,(x3,y3),(x4,y4),[0,255,0],2)
            cv2.line(new_img,(x3,y3),(x1,y1),[0,255,0],2)
            cv2.line(new_img,(x2,y2),(x1,y1),[0,255,0],2)
            cv2.line(new_img,(x2,y2),(x4,y4),[0,255,0],2)
        #if classifier return 0 ,then write "0" to adaboost_pred.txt
        else:
            ff.write('0 ')
        #refresh the list2 ,and append the new_img to the imglist
        list2=[]
        imglist.append(new_img)

        ff.write("\n")
        #if the frame has been the tail of the gif,then break
    except EOFError:
        print("end")
        break
#follow the instruction that the spec said
#so i show the first element in the imglist
#which is the first frame
cv2.imshow("0",imglist[0])
ff.close()
cv2.waitKey(0)
```

Snapshot of Part6.

這部分我查到很多其他的算法，但是因為不熟練因此未能來得及呈現出結果，所以我轉為對 adaboost 下手，我修改計算 error 的方式，從 mean absolute 改為 mean square 以提高它的準確率。

只不過很遺憾的，training set 跟 data set 的 accuracy 都下降了不少，看來應該有其他更好的調整方式。

```
# Begin your code (Part 2)
bestError = 1
bestClf = WeakClassifier(features[0])

for j in range(len(features)):

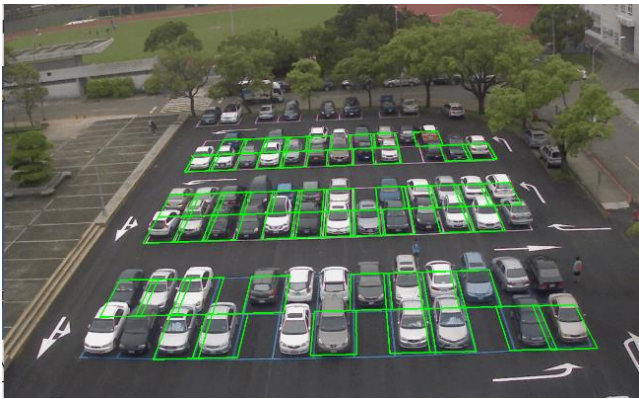
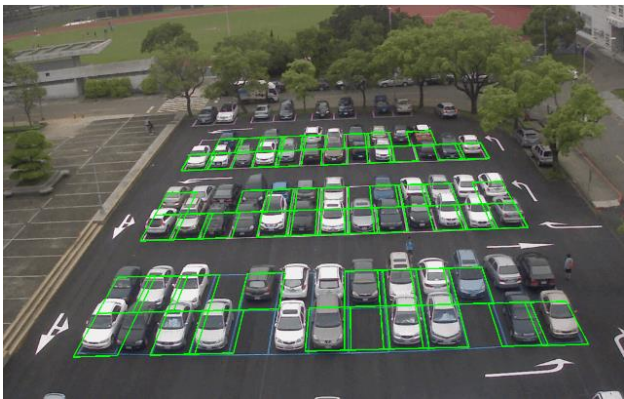
    tmperror = 0
    feature_classifier = WeakClassifier(features[j])

    for i in range(len(weights)):
        value = feature_classifier.classify(iis[i])
        # tmperror += (weights[i]* abs(value - labels[i]))

        #i modified the mean absolute error
        # to the mean square method
        tmperror += (weights[i]* pow((value - labels[i]),2))
    tmperror = pow(tmperror,0.5)

    if(tmperror < bestError):
        bestError = tmperror
        bestClf = feature_classifier

#raise NotImplementedError("To be implemented")
# End your code (Part 2)
return bestClf, bestError
```

Adaboost		Modified Adaboost	
			
Accuracy		Accuracy	
Train	0.905	Train	0.816
Test	0.868	Test	0.788

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

& Discuss the difference between Adaboost and Yolov5 (Part5.)

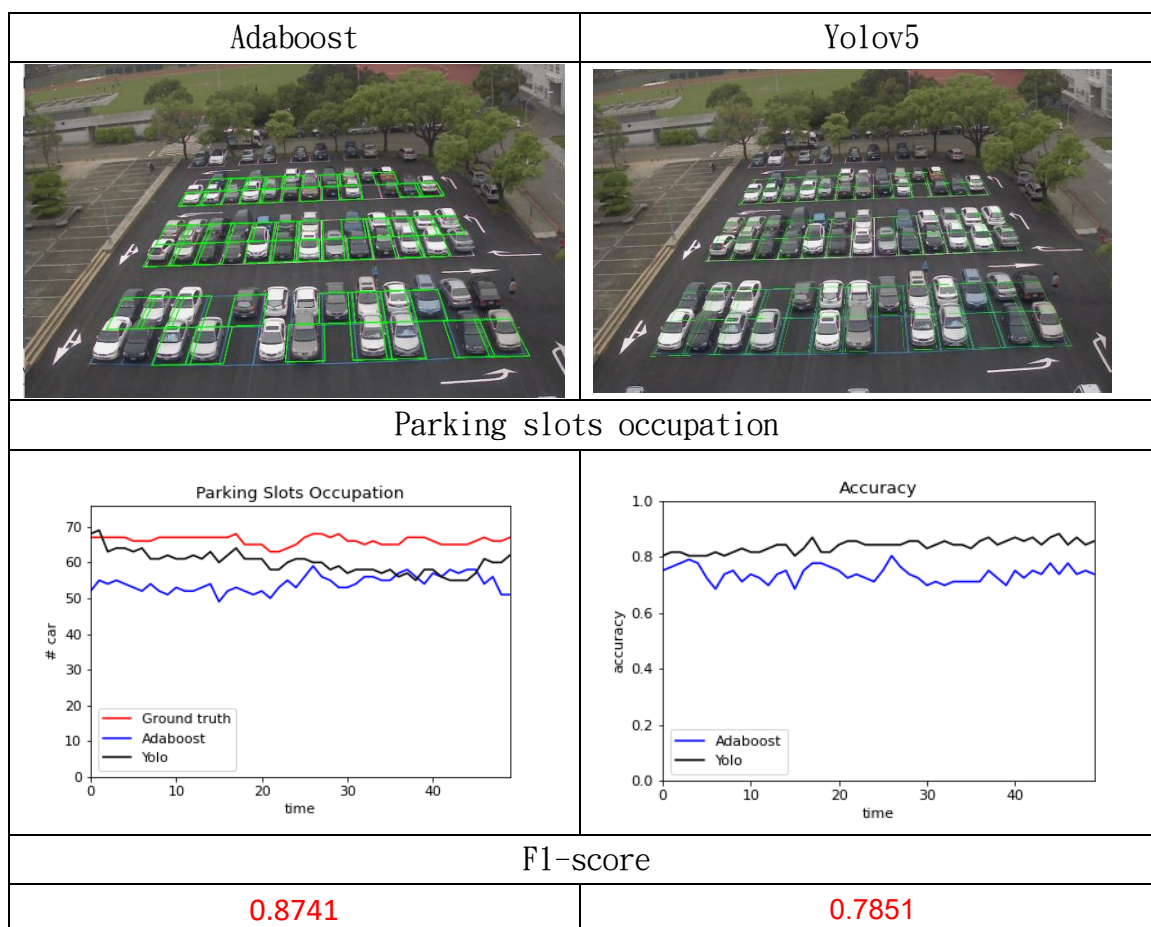
我們從下方兩張停車場的圖，以及plots occupation、accuracy的分布圖來討論。

我發現雖然我已經是拿取adaboost T=1~10中最為準確的一次結果來做比較，但是整體來說，Yolov5仍然較為精準，這可能有幾個原因。

1. adaboost 是將大量的弱分類器得出的結果做整合而得的推論，因此結果尚有滿大的進步空間。
2. adaboost在計算error時是採取”絕對值*權重”，這也有改良的空間，我有將我的嘗試結果放在上段尾部的部分。
3. 仔細看會發現，F1-score表現的部分，竟然有可能adaboost是高於yolov5的！我猜想這可能有幾個原因：

甲、觀察的樣本不夠。應該說，我只有觀察到單一次yolov5的結果，或許剛好是這筆資料的精準度較低，如果用更大量的資作比對可能有機會衝高f1-score。

乙、Yolov5的參數。如果我試著調整yolov5內部的參數，例如confidence_threshold等等之類的，或許能夠增加yolov5的精準度，進而提高f1-score。

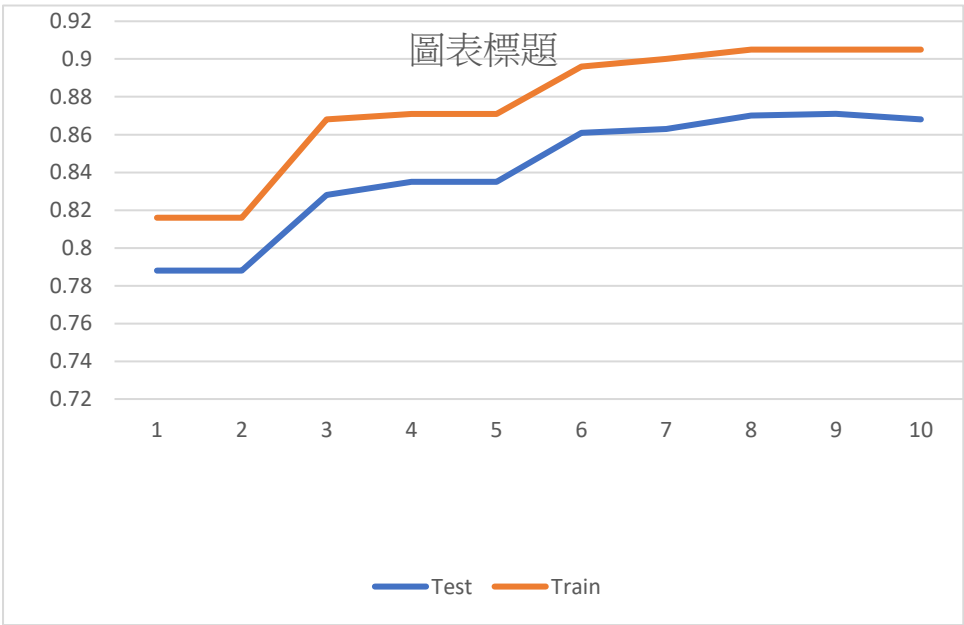


我將adaboost 10次iteration的結果繪製成表格與折線圖提供在下方。

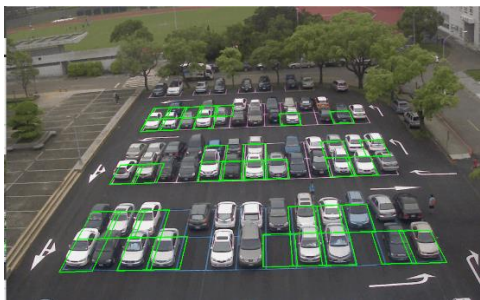
以下分享一些我觀察到除了上段，額外關於adaboost的結果：

1. 伴隨越高的iteration次數，accuracy有向上成長的趨勢，我認為這或許表示training的次數越多，對於電腦在判斷與分類的準確率有一定程度的幫助與增益。
2. 根據圖片也呈現肉眼可見的，T的數量越大，停車格判斷的正確率也明顯比前幾次都要更好。
3. 下次或許可以再試試看train更多的dataset，雖然這次已經是600/600了，不過如果訓練更多筆資料，應該會有助於提升準確率。

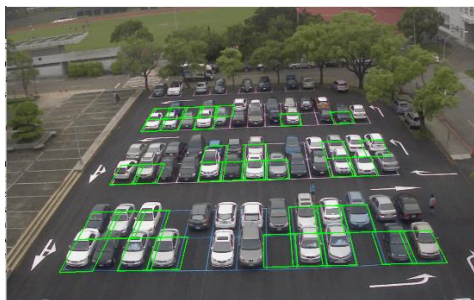
T	Test	Train
1	0.788	0.816
2	0.788	0.816
3	0.828	0.868
4	0.835	0.871
5	0.835	0.871
6	0.861	0.896
7	0.863	0.900
8	0.871	0.905
9	0.871	0.905
10	0.868	0.905



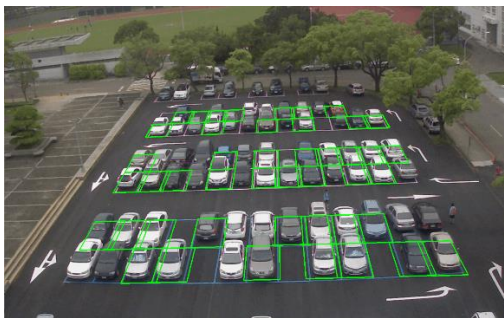
T=1



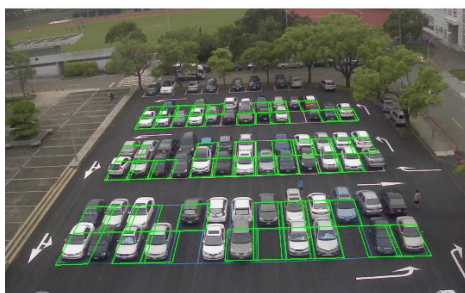
T=2



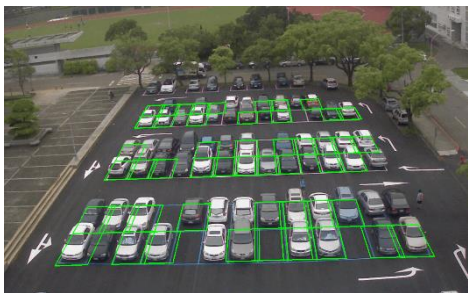
T=3



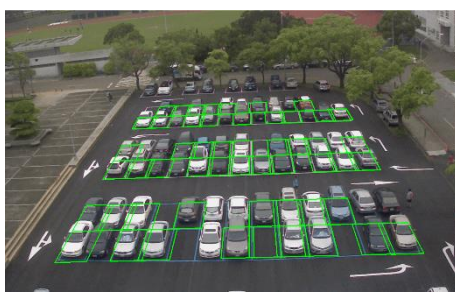
T=4



T=5



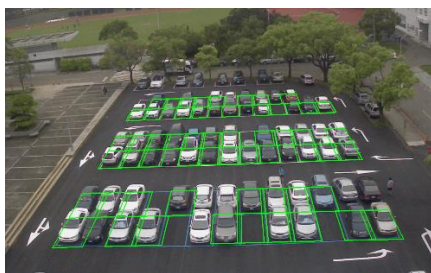
T=6



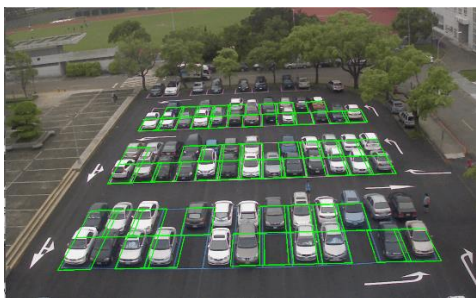
T=7



T=8



T=9



T=10



4. Describe problems you meet and how you solve them

這次有幾個比較有趣的點，第一個就是一開始我不知道同樣 T 的情況下其實只需要 train 一次就好，所以我很納悶這份作業未免也太耗費時間？每次都要將近十分鐘才能跑出一個答案，有時候要 debug 也要等好久，後來才知道 train 過一次之後就不用重複 save 跟 load。

問題是，在我知道這個好康之後，我就爽快地跑完 T=1 到 T=10，赫然發現每次結果都一樣???後來請教同學才知道原來更改 T 之後要重新 train 過，不然.pkl 檔根本長的一樣。

另外就是，這次不是直接讀取照片或是影片，而是 gif。雖然後來問到同學有利用 cv2 去處理，但是根據我原本查詢到的方法都是用 PIL 去存取 gif 再對每個 frame 去操作，所以多了一些步驟，再加上 cv2 是採取 BGR，然而 PIL 跟現今主流差不多，都是使用 RGB 的方式輸出影像的三通道，然後在過程中我因為對模組的不熟悉又犯一些蠢，所以在轉換上又花了一些時間在調整上。

然後還有在操作 yolov5 的時候，我一直碰到 img size 的問題，後來在討論區得到助教的救援，更改在我的 colab 之後發現，好像沒解決(?)，但是它變成可以執行下一步了，所以應該還是多少有點差異，非常感動。

這次的 hwl 其實滿有趣的，雖然我個人因為對 adaboost 不熟，所以一開始花了些時間在理解他的運算原理、甚至借了其他班同學的講義來看，但在過程中學習到非常多新知識，也趁機理解到要未來如果要跑更深入的 training 或是一些學習的話，需要多好的電腦才跑得動 😊

這次也很感謝助教在討論區上很即時的回覆，雖然我都不是提出問題的人(因為我遇到問題的時候都有同學已經問過了…)，但是基本上我的疑惑都能在討論區看到助教詳細的解說，謝謝助教~

109550025 謝翔丞