Artificial-Intelligence

homework1

0816023 張紳濡

1 Introduction

使用 Adaboost 與 Yolov5 對停車場的車輛進行偵測。

2 Data

- train: 600 car pictures and 600 non-car pictures.
- test: 600 car pictures and 600 non-car pictures.
- detect: A gif video with 50 pictures.

3 Adaboost Algorithm

使用許多 WeekClassifier 組合起來，每次做完調整 weight 使當前的 error rate 達到 0.5 並將新的 weight 傳遞給下個 WeekClassifier(這部分在 adaboost.py train function 內已寫好，詳細不在這贅述)

```
for t in range(self.T):
    print("Run No. of Iteration: %d" % (t+1))
    # Normalize weights
    weights = weights / np.linalg.norm(weights)
    # Compute error and select best classifiers
    clf, error = self.selectBest(featureVals, iis, labels, features, weights)
    #update weights
    accuracy = []
    for x, y in zip(iis, labels):
        correctness = abs(clf.classify(x) - y)
        accuracy.append(correctness)
    beta = error / (1.0 - error)
    for i in range(len(accuracy)):
        weights[i] = weights[i] * (beta ** (1 - accuracy[i]))
    alpha = math.log(1.0/beta)
    self.alphas.append(alpha)
    self.clfs.append(clf)
```

4 What I do

4.1 Part I Load Data

將 train 與 test 的 car, non-car 圖片讀進並轉灰階與 resize，將其轉成 nparray 並與 label 一起包成 tuple，存進一個 list。

```
dataset = []
for filename in os.listdir(dataPath+"/car"):
    image = cv2.cvtColor(cv2.resize(cv2.imread(dataPath+"/car/"+filename),(36,16)),cv2.COLOR_BGR2GRAY)
    dataset.append((np.asarray(image),1))
for filename in os.listdir(dataPath+"/non-car"):
    image = cv2.cvtColor(cv2.resize(cv2.imread(dataPath+"/non-car/"+filename),(36,16)),cv2.COLOR_BGR2GRAY)
    dataset.append((np.asarray(image),0))
```

4.2 Part II SelectBest

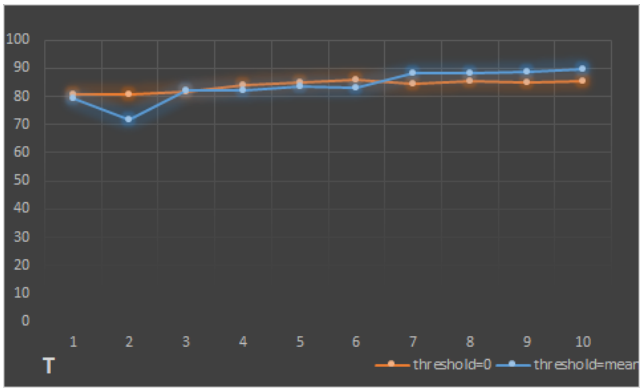
用 features 產生一堆 WeekClassifier(features 大約有 13W)

- feature:
  - line 181: 每次將 feature 傳進 classifier.py 內的 WeekClassifier，並將 threshold 設為 0
  - line 180: 每次將 feature 傳進 classifier.py 內的 WeekClassifier，並將 threshold 設為 mean(mean(car_featureval)+car_featureval))
- 將 ii 傳進去 (600 張) 計算，回傳一個 boolean 值。
- 對於每次的 ii 若回傳直不等於其 label，將其加入 errorsum(line 185)
- 對於每個 feature，若其 errorsum < mineerror，將 minerror 更新並將其記錄為 bestclf(line 187)。
- 將 minerror normalize，設為 bestError(line 191)。

```
168 bestClf = WeakClassifier(features[0])
169 weightsum = 0
170 for w in weights:
171     weightsum += w
172 errsum = 0
173 minerrnum = weightsum
174 for feature, val in zip(features, featureVals):
175 # for feature in features:
176     car_val = np.mean(val[np.array(labels) == 1])
177     not_car_val = np.mean(val[np.array(labels) == 0])
178 
179     # print(car_val, not_car_val)
180     clf = WeakClassifier(feature, np.mean(car_val + not_car_val), -1 if car_val > not_car_val else 1)
181     # clf = WeakClassifier(feature)
182     errsum = 0
183     for i in range(len(iis)):
184         re = clf.classify(iis[i])
185         if re != labels[i]:
186             errsum += weights[i]
187     if errsum < minerrnum:
188         minerrnum = errsum
189         bestClf = clf
190         # print(minerrnum)
191 bestError = minerrnum/weightsum
```

4.3 Part III Additional experiments

對於 Part II 兩種方法，T 從 0 到 10 進行測試得到以下圖表



T	threshold=0	threshold=mean
1	80.8	79.5
2	80.8	71.8
3	81.5	82.1
4	84.1	82.1
5	85.1	83.5
6	86.1	83.1
7	84.3	88.5
8	85.3	88.5
9	85	88.8
10	85.5	89.6

兩可以發現兩者在 T 變大時正確率都會提高，但 Threshold = mean 在更高的 T 有更好的表現，最後的比較我拿 Threshold =0,T=6 與 Threshold = mean,T=10 來比較

4.4 Part IV Detect car

- 將 video.gif 與 detectData.txt 讀進來 (line 59 62)
- 對於每張 frame 使用 crop fuction 切出 360*160 的圖片並 resize 成 36*16 的圖片轉成灰階 (line 72,73)
- 傳入 Part III 訓練好的 clf(line 74)
- 若回傳值為 1 就在其對應值畫出四邊形並輸出 1 或 0(line 77 80)
- 最後輸出這張畫好的 frame(line 88)(這邊我輸出了全部 50 張 frame 到 code 內的路徑)

```
59    vdo = cv2.VideoCapture("data/detect/video.gif")
60    fl = open(dataPath,'r')
61    pos = fl.read()
62    pos = pos.split()
63    file = open('Adaboost_pred.txt','w+')
64    k = 0
65    while vdo.isOpened():
66        output = []
67        ret,frame = vdo.read()
68        if not ret :
69            break
70        newimg = frame
71        for i in range(1,int(pos[0])*8,8):
72            img = crop(int(pos[i]),int(pos[i+1]),int(pos[i+2]),int(pos[i+3]),int(pos[i+4]),int(pos[i+5]),int(pos[i+6]),i
73            image = cv2.cvtColor(cv2.resize(img,(36,16)),cv2.COLOR_BGR2GRAY)
74            if clf.classify(np.asarray(image)):
75                # p1 = [int(pos[i+4]),int(pos[i+5])]
76                # p2 = [int(pos[i+2]),int(pos[i+3])]
77                cv2.line(newimg, (int(pos[i]),int(pos[i+1])), (int(pos[i+2]),int(pos[i+3])), (0,255,0), 1)
78                cv2.line(newimg, (int(pos[i+2]),int(pos[i+3])), (int(pos[i+6]),int(pos[i+7])) , (0,255,0), 1)
79                cv2.line(newimg, (int(pos[i+6]),int(pos[i+7])), (int(pos[i+4]),int(pos[i+5])) , (0,255,0), 1)
80                cv2.line(newimg, (int(pos[i+4]),int(pos[i+5])), (int(pos[i]),int(pos[i+1])) , (0,255,0), 1)
81                # cv2.rectangle(newimg,tuple(p1),tuple(p2),(0,255,0),1)
82            if clf.classify(np.asarray(image)):
83                output.append(1)
84            else:
85                output.append(0)
86        print(*output, file=file)
87        cv2.imshow("AAA", newimg)
88        cv2.imwrite("detectpic/adaboostpic"+str(k)+".png", newimg)
```

4.5 Part V compare with Yolov5

4.5.1 三種方法的第一張 frame:

Figure 1: Adaboost T = 6,Threshold = 0

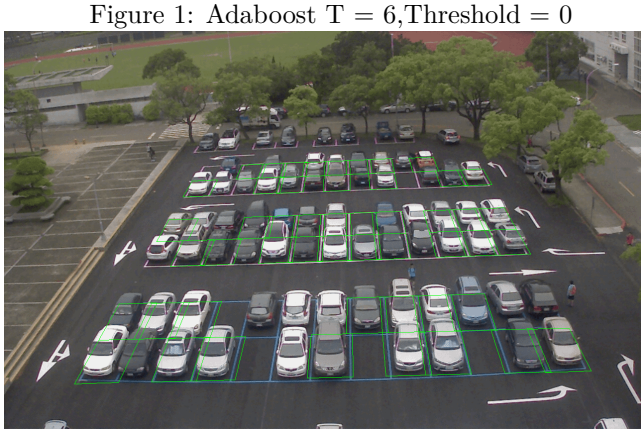


Figure 2: Adaboost T = 10,Threshold = 0

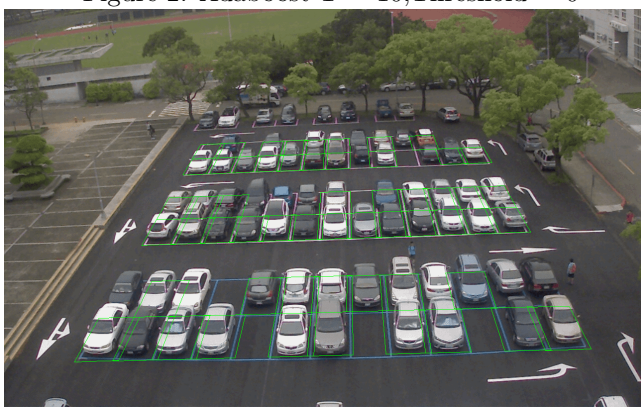
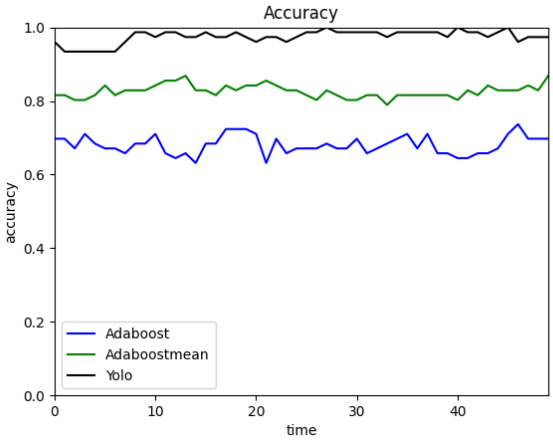


Figure 3: yolov5,confidence = 0.4

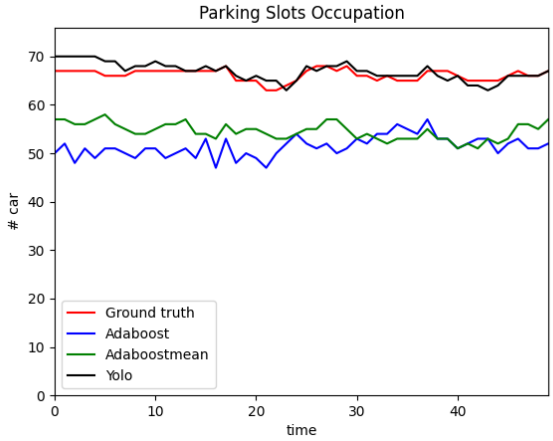
4.5.2 三種方法的 Accuracy:

Figure 4: Accuracy



4.5.3 三種方法的 Parking Slots Occupation:

Figure 5: Parking Slots Occupation



可以發現 Yolo 的 Accuracy 與 Parking Slots Occupation 表現都比 Adaboost 好，但使用 mean 當 threshold 表現上又比單純使用 0 當 threshold 的方法略好一些。

5 Problem I meet

5.1 SelectBest

當時在看完 adaboost algorithm 後卻不知道 selectbest 要我做甚麼，以及如何產出一堆的 weekclassifier，原本以為是要我們做調整 weight 參數的部分，但又看到期已經在 code 內。產生 weekclassifier 的方法遠本也以為是用調整 threshold，因為不知道 Haarfeature 在幹嘛。

5.2 Yolov5

Colab 的路徑部分遇到了一些問題，在 detect 部分要讀檔的路徑問題。以及 colab 不能用 waitkey，以及要我們 replace clf.classify() with yolov5_func.classify() 但其實不能直接使用要在前面加上 HW1_material.。另外在在 draw 的部分不知為何沒有 import pandas as pd。

5.3 other

Adaboost 跑得很久，又要從 T = 1 跑到 T = 10, 其 feature 的算法似乎很暴力，每張圖會有 13w 左右的 feature，總共又有 600 張圖

6 Problem I solve

6.1 SelectBest

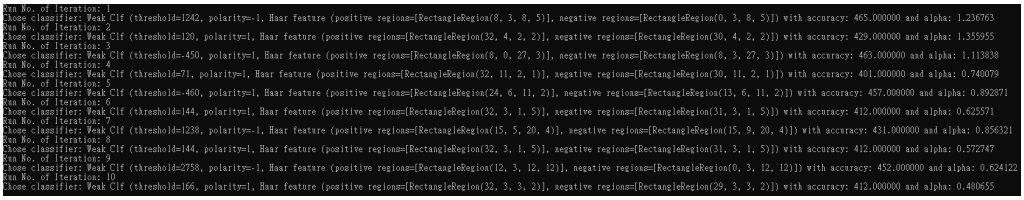
最後用輸入輸出推出要使用每個 feature 去產生 weekclassifier，並去大概看了一下 featureVal 在幹麻與其分布而有調整 Threshold = mean 的版本。

6.2 Yolov5

最後從上面的 code 與查了一下才找到路徑，draw 的部分我直接 copy 下來本機跑，並多繪製了 Threshold = mean 的摺線

6.3 other

後來有發現 adaboost 可以把 CLF 存下來，而且相同 T 取到的 feature 一樣 (SelectBest 回傳的一樣)，最後只跑一次 T = 10 並在每次存下當前 T 的 CLF，想知道每次 T 的 accuracy rate 只要跑 main.py 內的 utils.evaluate 部分就好了 (雖然發現時已經很晚了)



另外在執行時我把 adaboost.py 中 buildfeatures 內 w 與 h 調大產生較少的 feature 以加速測試，但其實調大一些雖然 feature 少了很多但正確沒下降多少

6.4 New Problem

- 是否有更好取 threshold 的方法可以增進 adaboost 的表現 (類似影像處理的 Otsu Threshold，找 featureval 分布兩分布曲線的交界處)
- yolov5 在本機執行，因為 colab 上路徑麻煩又要跑很久，不確定是否能直接弄下來在本機執行 (GPU 設定等)
- feature 的數量選擇，能否調整 w 與 h 找到執行時間與 accuracy 的平衡點。

7 Code

Adaboost code can be find in
https://github.com/Sakuya0229/Artificial-Intelligence-HW1