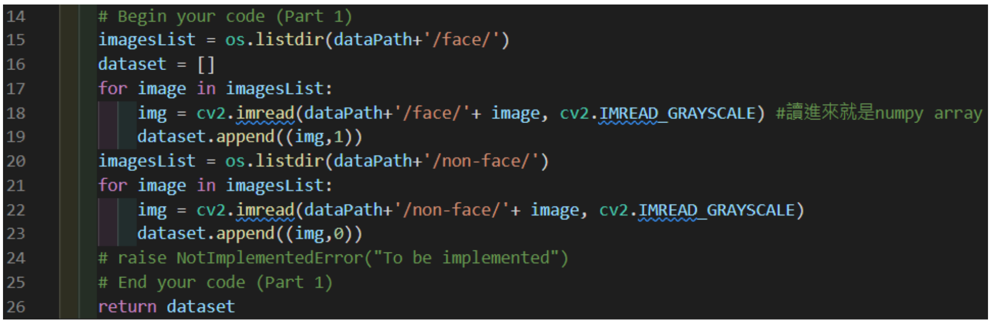
Homework 1: Face Detection Report (109550171)

Part I. Implementation (6%):

Please screenshot your code snippets of Part 1, Part 2, Part 4, and explain your implementation.

Part 1:



```
# Begin your code (Part 1)
imagesList = os.listdir(dataPath+'/face/')
dataset = []
for image in imagesList:
    img = cv2.imread(dataPath+'/face/'+ image, cv2.IMREAD_GRAYSCALE) #讀進來就是numpy array
    dataset.append((img,1))
imagesList = os.listdir(dataPath+'/non-face/')
for image in imagesList:
    img = cv2.imread(dataPath+'/non-face/'+ image, cv2.IMREAD_GRAYSCALE)
    dataset.append((img,0))
# raise NotImplementedError("To be implemented")
# End your code (Part 1)
return dataset
```

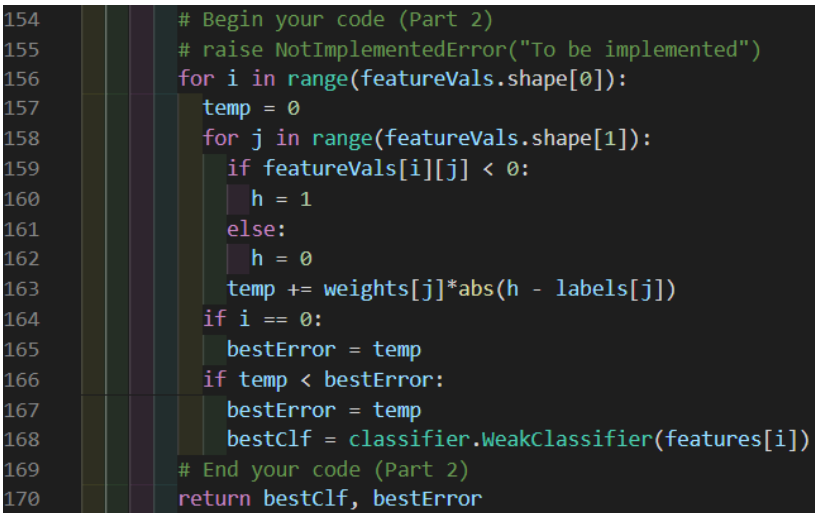
I dealt with ‘face’ and ‘non-face’ folders respectively, but I used the same way as following.

I used os module to load all images in the folders into a list called imagesList.

Then, I used cv2 to read images one by one, the result is a numpy array, and I combined the numpy array and classification into a tuple.

In the end, I appended these tuples into a list called dataset.

Part 2:

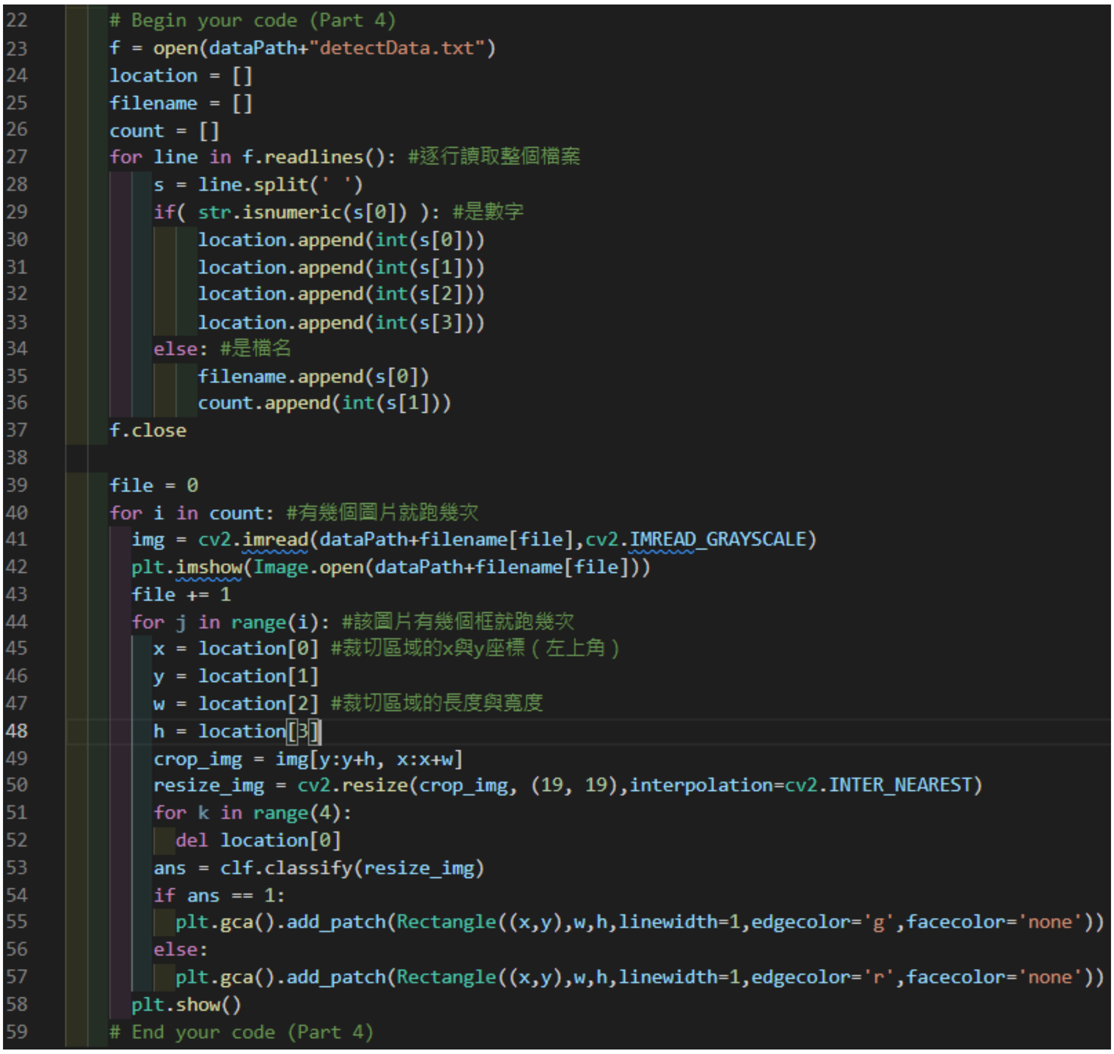


```
154 # Begin your code (Part 2)
155 # raise NotImplementedError("To be implemented")
156 for i in range(featureVals.shape[0]):
157     temp = 0
158     for j in range(featureVals.shape[1]):
159         if featureVals[i][j] < 0:
160             h = 1
161         else:
162             h = 0
163         temp += weights[j]*abs(h - labels[j])
164     if i == 0:
165         bestError = temp
166     if temp < bestError:
167         bestError = temp
168         bestClf = classifier.WeakClassifier(features[i])
169 # End your code (Part 2)
170 return bestClf, bestError
```

line 156~162: I used the classifier in the lecture slide to change every featureVals into result of classification.

line 163~168: I used the formula to evaluate the error, and the smallest is stored to bestError. I also recorded the best classifier.

Part 4:



```
22  # Begin your code (Part 4)
23  f = open(dataPath+"detectData.txt")
24  location = []
25  filename = []
26  count = []
27  for line in f.readlines(): #逐行讀取整個檔案
28      s = line.split(' ')
29      if( str.isnumeric(s[0]) ): #是數字
30          location.append(int(s[0]))
31          location.append(int(s[1]))
32          location.append(int(s[2]))
33          location.append(int(s[3]))
34      else: #是檔名
35          filename.append(s[0])
36          count.append(int(s[1]))
37  f.close
38  
39  file = 0
40  for i in count: #有幾個圖片就跑幾次
41    img = cv2.imread(dataPath+filename[file],cv2.IMREAD_GRAYSCALE)
42    plt.imshow(Image.open(dataPath+filename[file]))
43    file += 1
44    for j in range(i): #該圖片有幾個框就跑幾次
45      x = location[0] #裁切區域的x與y座標（左上角）
46      y = location[1]
47      w = location[2] #裁切區域的長度與寬度
48      h = location[3]
49      crop_img = img[y:y+h, x:x+w]
50      resize_img = cv2.resize(crop_img, (19, 19),interpolation=cv2.INTER_NEAREST)
51      for k in range(4):
52        del location[0]
53      ans = clf.classify(resize_img)
54      if ans == 1:
55        plt.gca().add_patch(Rectangle((x,y),w,h,linewidth=1,edgecolor='g',facecolor='none'))
56      else:
57        plt.gca().add_patch(Rectangle((x,y),w,h,linewidth=1,edgecolor='r',facecolor='none'))
58    plt.show()
59  # End your code (Part 4)
```

I opened the txt file, and read the whole file line by line.

line 24~36: Depending on the type of the data, I stored them into different lists.

Loop “for i in count:” : show the images.

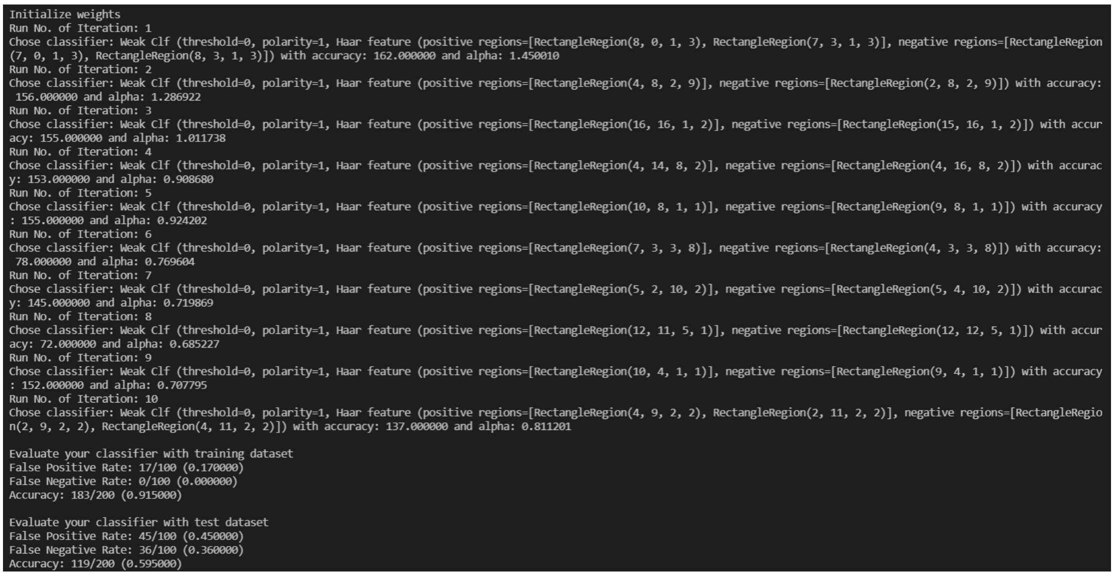
Loop “for j in range(i):” : for every box in the image, I extracted the information(coordinate, height, width) of the box and cropped the image.

line 53: put the cropped image into the classify function.

line 54~57: I used the result of the classify function to decide the color of the box.

Part II. Results & Analysis (12%):

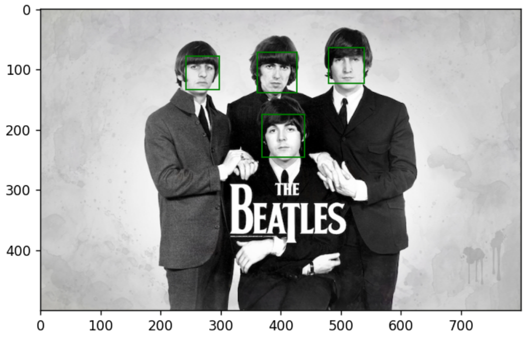
- Please screenshot the results.

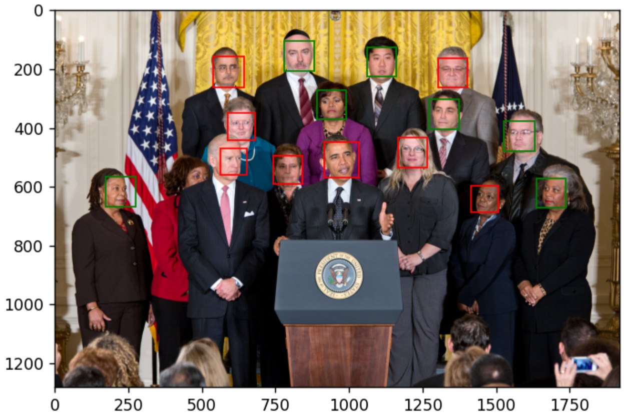


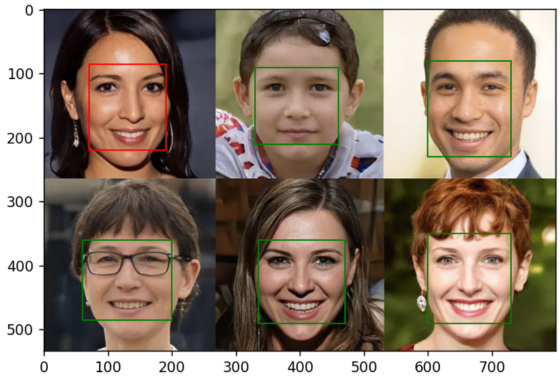
```
Initialize weights
Run No. of Iteration: 1
Chose classifier: Weak Clf (threshold=0, polarity=1, Haar feature (positive regions=[RectangleRegion(8, 0, 1, 3), RectangleRegion(7, 3, 1, 3)], negative regions=[RectangleRegion
(7, 0, 1, 3), RectangleRegion(8, 3, 1, 3)]) with accuracy: 162.000000 and alpha: 1.450010
Run No. of Iteration: 2
Chose classifier: Weak Clf (threshold=0, polarity=1, Haar feature (positive regions=[RectangleRegion(4, 8, 2, 9)], negative regions=[RectangleRegion(2, 8, 2, 9)]) with accuracy:
 156.000000 and alpha: 1.286922
Run No. of Iteration: 3
Chose classifier: Weak Clf (threshold=0, polarity=1, Haar feature (positive regions=[RectangleRegion(16, 16, 1, 2)], negative regions=[RectangleRegion(15, 16, 1, 2)]) with accur
acy: 155.000000 and alpha: 1.011738
Run No. of Iteration: 4
Chose classifier: Weak Clf (threshold=0, polarity=1, Haar feature (positive regions=[RectangleRegion(4, 14, 8, 2)], negative regions=[RectangleRegion(4, 16, 8, 2)]) with accurac
y: 153.000000 and alpha: 0.908680
Run No. of Iteration: 5
Chose classifier: Weak Clf (threshold=0, polarity=1, Haar feature (positive regions=[RectangleRegion(10, 8, 1, 1)], negative regions=[RectangleRegion(9, 8, 1, 1)]) with accuracy
: 155.000000 and alpha: 0.924202
Run No. of Iteration: 6
Chose classifier: Weak Clf (threshold=0, polarity=1, Haar feature (positive regions=[RectangleRegion(7, 3, 3, 8)], negative regions=[RectangleRegion(4, 3, 3, 8)]) with accuracy:
 78.000000 and alpha: 0.769604
Run No. of Iteration: 7
Chose classifier: Weak Clf (threshold=0, polarity=1, Haar feature (positive regions=[RectangleRegion(5, 2, 10, 2)], negative regions=[RectangleRegion(5, 4, 10, 2)]) with accurac
y: 145.000000 and alpha: 0.719869
Run No. of Iteration: 8
Chose classifier: Weak Clf (threshold=0, polarity=1, Haar feature (positive regions=[RectangleRegion(12, 11, 5, 1)], negative regions=[RectangleRegion(12, 12, 5, 1)]) with accur
acy: 72.000000 and alpha: 0.685227
Run No. of Iteration: 9
Chose classifier: Weak Clf (threshold=0, polarity=1, Haar feature (positive regions=[RectangleRegion(10, 4, 1, 1)], negative regions=[RectangleRegion(9, 4, 1, 1)]) with accuracy
: 152.000000 and alpha: 0.707795
Run No. of Iteration: 10
Chose classifier: Weak Clf (threshold=0, polarity=1, Haar feature (positive regions=[RectangleRegion(4, 9, 2, 2), RectangleRegion(2, 11, 2, 2)], negative regions=[RectangleRegio
n(2, 9, 2, 2), RectangleRegion(4, 11, 2, 2)]) with accuracy: 137.000000 and alpha: 0.811201

Evaluate your classifier with training dataset
False Positive Rate: 17/100 (0.170000)
False Negative Rate: 0/100 (0.000000)
Accuracy: 183/200 (0.915000)

Evaluate your classifier with test dataset
False Positive Rate: 45/100 (0.450000)
False Negative Rate: 36/100 (0.360000)
Accuracy: 119/200 (0.595000)
```

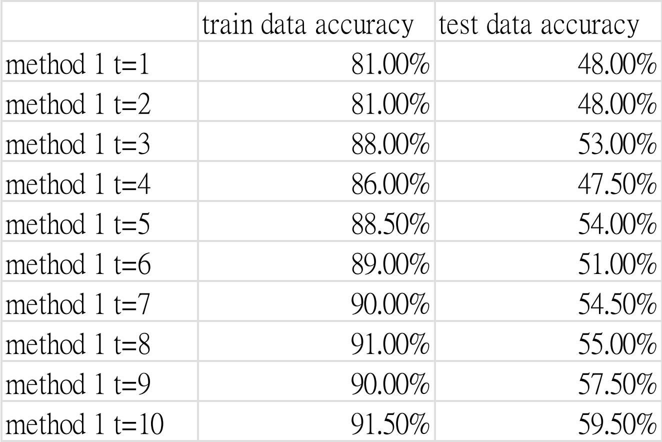






● Your analysis or observation.

Please discuss the performance difference between the training and testing dataset, and present the results using a table or chart as follows.



	train data accuracy	test data accuracy
method 1 t=1	81.00%	48.00%
method 1 t=2	81.00%	48.00%
method 1 t=3	88.00%	53.00%
method 1 t=4	86.00%	47.50%
method 1 t=5	88.50%	54.00%
method 1 t=6	89.00%	51.00%
method 1 t=7	90.00%	54.50%
method 1 t=8	91.00%	55.00%
method 1 t=9	90.00%	57.50%
method 1 t=10	91.50%	59.50%

When t becomes bigger, the train data and test data accuracy increase.
Because we use train data to train the module, the train data accuracy is higher than test data accuracy.

Part III. Answer the questions (12%):

1. Please describe a problem you encountered and how you solved it.

In the beginning, I had no idea about the meaning of the numbers in detectData.txt.
I opened the images in python, in that case, I can observe the coordinate of the images.
In the end, I found that the first two numbers in a row are coordinate of the upper left, the third and fourth number is the width and height.

2. What are the limitations of the Viola-Jones' algorithm?

1. It is too sensitive to lighting conditions.
2. It can't detect tilted faces well.
3. Restricted to binary classification.

3. Based on Viola-Jones' algorithm, how to improve the accuracy except increasing the training dataset and changing the parameter T?

Use high-qualified photo.

4. Please propose another possible face detection method (no matter how good or bad, please come up with an idea). Please discuss the pros and cons of the idea you proposed, compared to the Adaboost algorithm.

knowledge-based system:

Pros:

1. Always considers current data.
2. Analysis of a large amount of data in a short time.

Cons:

Doesn’t accept decisions which are different from the rules.