

Face-Recognize

- 使用方法與套件:

我們採用 `keras` 套件實作 CNN 模組，用自定義的陣列方式讀取圖片當作訓練資料。

- 步驟:

1. 讀檔:

使用 `image.open()` 讀檔，讀入的圖片資料會以陣列型態保存，然後因為每張圖片的大小不一樣所以再進行 `resize()`，使每張圖片的大小都一樣方便陣列儲存，也更好載入模組。

第 33 行:目的是將 RGB 的圖案轉成灰階影像。

第 34 行:這裡是將 array 的數值都統一處理成 $0 < \text{val} < 1$ 因為少了 05,09 所以我們只好將迴圈設計成 50 人 13 個每人。最後用 `face_data` 紀錄。(如下圖)

```
21 for row in range(50):
22     for col in range(13):
23         if col==4:
24             continue
25         if col==8:
26             continue
27         p=row+1
28         n=col+1
29         fs=fs[:1]+'{:0>2d}'.format(p)+fs[3]+'{:0>2d}'.format(n)+fs[6:]
30         print (fs)
31         img = Image.open(fs)
32         img = img.resize((dcol, drow), Image.BILINEAR)
33         img = img.convert('L')
34         img_ndarray = numpy.asarray(img, dtype='float64')/ 255
35         face_data[row*13+col] =img_ndarray
36
```

2. 製作數據組:

載入完的陣列可以直接拿來用，當作 **X** 作為數據

組，而 **y** 則是對印每個 **X** 數據的 **label**。

所以就簡單地用迴圈載入對印的陣列中。

第 45 行:這一行是使用 `train_test_split()` 函數從以建立的數據中隨機產生 **test** 的資料。

```
37 X = face_data
38 print (X.shape)
39 face_label = numpy.empty(650)
40 for i in range(650):
41     face_label[i]= int(i/13)
42 y = face_label
43 print (y.shape)
44
45 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state= 30)
46 print (X_train.shape)
47 print (X_test.shape)
48 print (y_train.shape)
49 print (y_test.shape)
50
51 print("Changing format.....")
52
53 X_train = X_train.reshape(-1, 1, dcol, drow)/255.
54 X_test = X_test.reshape(-1, 1, dcol, drow)/255.
55 y_train = np_utils.to_categorical(y_train, num_classes=50)
56 y_test = np_utils.to_categorical(y_test, num_classes=50)
57
```

3. 建立模組訓練

接下來就是利用 **keras** 套件，建立卷基層。

調整 **learning rate**、**loss function**、**optimizer function**。

最後用 `model.fit(X,y)` 載入訓練數據，在完成了。