

Specifications of the FaceRecognizer System for ML Term Project

組員:高英皓、陳彥儒、姜博瀚

Github

https://github.com/louis8ok1/Face_recongntion/

(a) 技術運用

Dense

是一個 fully connect layer，是最基本的連接層。

Dropout

隨機 Disable neuron，用來防止 Over-fitting。

Conv2D

卷積層；CNN 的基本層。

MaxPooling2D

最大化池化層

Flatten

把 MaxPooling2D 轉換降成一維，用來讓 MaxPooling2D 連接 Dense。

To_categorical

讓 Label1 one-hot encoding，之後可以用 Cross_entropy 計算 Loss。

Shuffle

隨機將輸入圖片洗牌，用來防止每次訓練結果都一樣。

(b) 程式描述

```
39 model = Sequential()
40 model.add(Conv2D(16, 3, activation="relu", input_shape=(240, 180, 3)))
41 model.add(MaxPooling2D(pool_size=(2, 2)))
42 model.add(Dropout(0.25))
43
44 model.add(Conv2D(32, 3, activation="relu"))
45 model.add(MaxPooling2D(pool_size=(2, 2)))
46 model.add(Dropout(0.25))
47
48 model.add(Conv2D(64, 3, activation="relu"))
49 model.add(MaxPooling2D(pool_size=(2, 2)))
50 model.add(Dropout(0.25))
51
52 model.add(Flatten())
53 model.add(Dense(128, activation='relu'))
54 #model.add(Dropout(0.5))
55 model.add(Dense(51, activation='softmax'))
56 print(" Model summary:")
57 model.summary()
58 model.compile(loss=keras.losses.categorical_crossentropy,
59               optimizer=keras.optimizers.Adadelta(),
60               metrics=['accuracy'])
```

在每層 Convolution 後面接上 MaxPooing2D 和 Dropout，MaxPooing2D 可以抽取每個圖片 2x2 的特徵(圖片變為一半大小)，並隨機 disable 25% 的 neuron。之後用 Flatten 將 Convolution 層降為一維，輸入 fully connect layer 使用 softmax 算出 50 個 Class 中，每一個 Class 的機率。最後將 Model compile，使用 Categorical Cross_entropy 做 Loss function，Optimizer 為 Adadelta。

```

15 for root, dirs, files in walk("Face Database/"):
16     for f in files:
17         fullpath = join(root, f)
18
19         Y.append(fullpath[15:17])
20
21         imageA = mpimg.imread(fullpath)
22
23         X.append(imageA)
24
25 X, Y = shuffle(X, Y)
26
27 Train_Data = np.array(X[:600]) / 255 * 2 - 1
28 Train_Label = np.array(Y[:600])
29
30 Test_Data = np.array(X[600:]) / 255 * 2 - 1
31 Test_Label = np.array(Y[600:])
32
33 Train_Label = to_categorical(Train_Label, num_classes=51)
34 Test_Label = to_categorical(Test_Label, num_classes=51)
35
36 print("Training Data Shape: ", Train_Data.shape)
37 print("Training Label Shape: ", Train_Label.shape)

```

將所有圖片載入後，將圖片編號 Label。接著把資料 Shuffle，再標準化成 1~-1 之間的數字，最後將 Label 做 One hot encoding。

```
41 model = load_model('train_model.h5')
42 model.summary()
43 result = model.predict(X)
44 print("Predict result: ")
45 print(result[0])
46 print("\nTrue Answer: ")
47 print(Y)
48
49 result_dict = {}
50 for i in range(len(result[0])):
51     result_dict[i] = float(result[0][i])
52 sorted_result_dict = sorted(result_dict.items(), key=operator.itemgetter(1), reverse=True)
53 cnt = 0
54 print("\nPredict Answer:")
55 print(sorted_result_dict[0][0])
56 print('\nTop 5 candidate: ')
57 for tp in sorted_result_dict:
58     cnt += 1
59     print(str(tp[0])+' Possibility: %0.6f' %(tp[1]))
60     if cnt >= 5:
61         break
```

讀入已經訓練好的 model 做預測，經過排序後，將前五名機率較高的 Class 和正確答案做比較。

(c) 程式測試過程

1. 輸入圖片庫中的所有圖片，圖片編號作為 Label
2. 把資料 Shuffle
3. 將圖片標準化成-1~1 之間的數字
4. 把 Label 做 One hot encoding
5. 將先前訓練好的 Model 讀入
6. 輸入要預測的圖片
7. 獲得每一個 Class 的機率
8. 將每一個 Class 的機率排序
9. 取出前五名的機率後印出

(d) 問題解決

1. 一開始 CNN 網路效果很差，不僅 Loss 大，Accuracy 也很低，test_accuracy 更永遠是 0.。首先做了資料標準化，將圖片每個像素值限制在 1~1 之間，讓 Loss 變小了很多。因為一開始沒有 Shuffle，導致 Classifier 取得沒有訓練過的 data，所以 test_accuracy 一直是 0。使用 Shuffle 之後隨機取得 test data，解決了這個問題。
2. 現在 Loss 降低了很多，accuracy 也有了 60%。但是無論怎麼修改 Model 架構，都無法進一步的獲得更好的準確率。之後在 Convolution 層連接上 MaxPooling 抽取每一個 2*2 像素中的特徵，此時 Accuracy 基本上有 90% 以上。
3. 一開始想不到如何選出前五名 Candidate，後來想到可以用 Python 的 Dictionary 做排序，就解決了這個問題。

(e) 組員分工

高英皓：Model 初始雛形、預測結果處理

陳彥儒：Model 架構及調教

姜博瀚：Mode 調教、報告製作

(f) 個人心得

陳彥儒：

這次的人臉辨識專案我認為非常有趣，從一開始不知道該如何下手，到後面跟組員討論慢慢把成品做出來。當我們看到預測圖片跟實際名字一樣時，那種喜悅感真的很棒！在這之中蠻麻煩的就是提升準確度，因為每一次在訓練前都不會知道出來的 loss 跟 acc 多少，所以只能一直去做嘗試，感謝組員！

高英皓：

一開始做的時候，Loss 飛天且 Accuracy 低到爆炸。這時候花了很多時間調整但依然沒有改進。多虧了和組員一起討論程式的方法，才順利的解決這個問題。如果之後有機會可以把它做成 APP 出來，讓他更有實際應用價值，將一定更有展望。

姜博瀚：

第一次嘗試製作人臉辨識，看到題目也是無從下手，後來跟組員討論之後，才理解一些技術要怎麼運用，以前都只是了解大概，但從未實做過。如今能看到成品且預測正確的感覺真好。