機器學習 final project

機械三
莊立楷

這次手指數字的辨識任務，我們使用自己設計的 CNN 架構進行訓練，然而這個任務的困難點就在於每張測試資料的背景都不同，這將造成機器辨識的難度大大提升，故我使用了一些解決方法進行改善，但結果都不是很好，在助教提供的測試集上，準確率在 30~40%中徘徊，但我真的很認真改方法訓練，希望能透過寫 report 的方式記錄下我整個努力的過程，助教給分不要太看重準確率，感謝！！

以下是我測試 accuracy 最好的一組 CNN 架構：

```
layers = [
    imageInputLayer([256 256 3])
    dropoutLayer(0.1)
    convolution2dLayer(10,40,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(16,40,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(10,40,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(10,40,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    dropoutLayer(0.2)
    convolution2dLayer(5,40,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(5,40,'Padding','same')
```

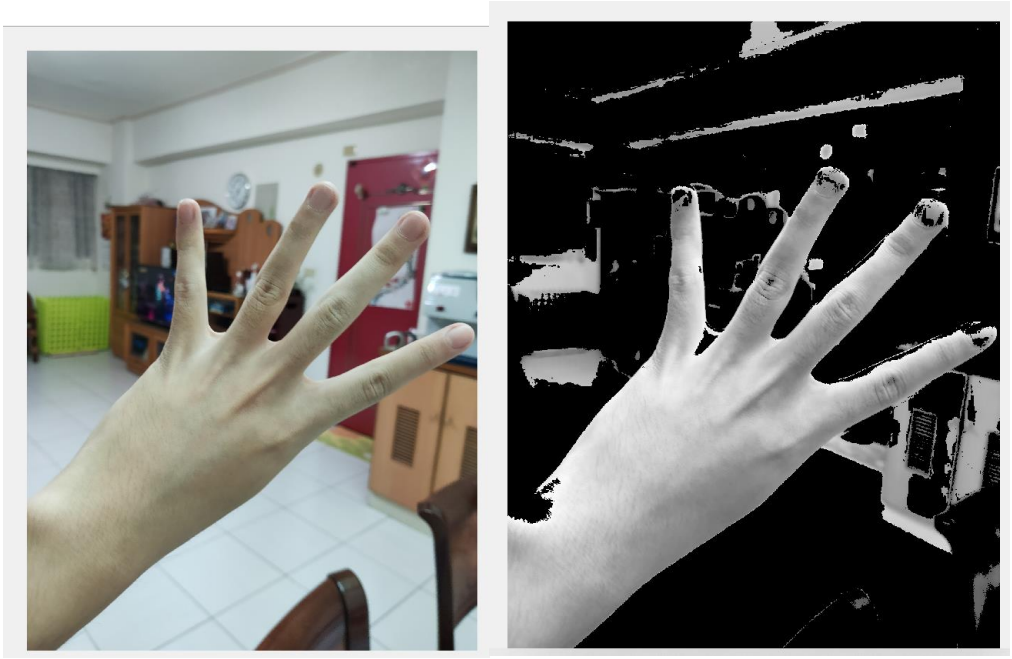
```
batchNormalizationLayer
reluLayer
maxPooling2dLayer(2,'Stride',2)
convolution2dLayer(5,40,'Padding','same')
batchNormalizationLayer
reluLayer
maxPooling2dLayer(2,'Stride',2)
dropoutLayer(0.2)
convolution2dLayer(5,40,'Padding','same')
batchNormalizationLayer
reluLayer
maxPooling2dLayer(2,'Stride',2)

fullyConnectedLayer(60)
reluLayer
fullyConnectedLayer(6)
softmaxLayer
classificationLayer];
```

我們發現 model 不夠深時，仍然可以 fit 訓練資料到 100%，但 validation 則接近 20%，當時以為是 overfitting，後來才發現需要把 model 加得更深，對 validation 的預測結果才會較好，不確定原因是什麼。(此時使用的 validation data 為我自己拍照切出來的，相較於助教給的照片，我的 validation 照片更接近 training 的照片，故 validation 準確度可以到 80%左右。)

然而，用這樣訓練好的模型去測試助教的照片時，準確率只有 37%，相當糟糕。值得一提的是，手勢數字為 0 和 1 的資料預測得較準確，但數字 2~5 則預測得亂七八糟。針對這個情況，我推論最大可能就是光線與背景問題造成辨識困難，故我執行了去背的動作：

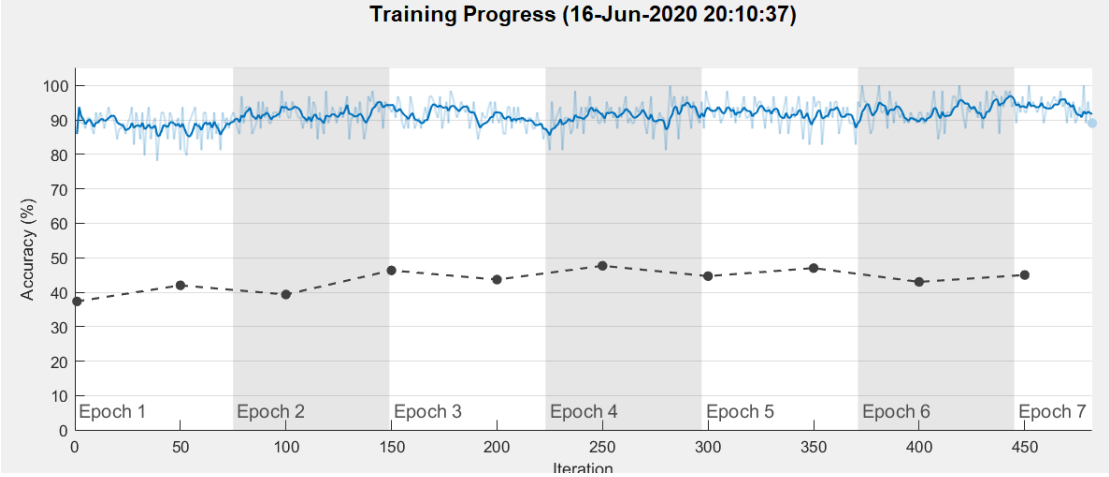
除了去背之外，我還執行了 PCA，其可將一張圖片的三維 RGB 資料降成一維變成灰階影像。使用 PCA 的原因維希望萃取出資料較好的特徵，同時降低運算量，將電腦運算配置給影像的 augmentation 的部分。



於是我的每張 input 資料則變為像上圖的部分，丟進 CNN 網路去訓練，結果….效果更差！原因是我的去背演算法僅為尋找 RGB 一個特定的閾值，藉此萃取出圖片中膚色的部分。然而，光線的影響還是很大，儘管我已經對整張相片做灰階值的 normalization，但光線的影響有時候是局部的，比如不是整張照片一起變亮，而是某些特定的手的部位變亮，這在我的演算法中是個大缺陷，造成手部可能會被侵蝕，關鍵的「手指」若減少了，資料則會整個錯掉。故最後放棄了去背的動作，直接將整張 RGB 餵給 CNN。

另外，為了增加準確度與防止 overfitting，我使用了 data 的 augmentation，將 input 的資料隨機旋轉、鏡射、平移，這方法讓我的資料量增加了 3~4 倍左右，但其缺點也十分明顯：儘管經過上述的線性轉換，資料雜亂的背景還是雜亂的背景！最主要的光源問題也沒有解決得很好，但此方法仍然帶給我們些微效果的提升，在助教給的測試集上準確率可以到 43%。

附上模型訓練後期的收斂情況(訓練前期成長幅度大，但忘記截圖了…)



其中藍線為 training accuracy，黑線為助教提供的測試集 accuracy，經過更久時間訓練後，最後準確率可以到 48%。

1. 以下為我使用的訓練習資料庫，共約 4000 張照片在雲端上。
https://drive.google.com/drive/folders/10fIP4BoFA0bqHfNoxcNz28jnLKFHNF2q?usp=sharing

2. 壓縮檔中有 training.m 檔，只要將裡面的訓練集、驗證集的檔案目錄位置改成助教電腦裡的檔案目錄位置即可，執行後即可開始重新訓練。(檔案目錄裡需要包含 6 個子資料夾(分別命名為 0,1,2,3,4,5)，各自放的是對應數字的圖片 。執行完成後會儲存一個 net.mat 檔，可用來 classify data。

3. 其中還有一個 trained_net.mat 檔，執行 testing.m 程式時會 load 進去。

4. testing.m 檔，只要將 rgbImage = imread('您的圖片')中的您的圖片改成助教電腦裡的測試圖片(圖片大小須為 256*256*3 維)位置即可，按執行後即可在 command window 中印出預測的數字為何。

5. 供同學測試的 10 張照片也在壓縮檔中。

註：我的 matlab 版本為 MATLAB R2019b。
若程式無法執行或會報錯，再麻煩助教跟我說！感謝！辛苦助教這學期了~