

Image Identification with Neural Network

Chest X-ray Image

Author: 吳岱錡

ID: 105071005

Supervisor: 陳素雲教授

壹、緒論

一、分析目的

在當今肺炎肆虐的背景下，肺炎的診斷含有三個層次：臨床診斷、放射線學診斷與病因診斷，即使一般而言只要患者們的胸部 X 光片上有陰影即可診斷肺炎，然而其檢測方式通常必須仰賴醫師們的專業才能作診斷，故希望藉此機會建構一個能自動檢查 X 光影像的分類器，以影像辨識的技術去判斷該檢驗者是否患有肺炎。

二、資料簡介

原資料：

- ◆ Train：5216 張肺部影像，1341 張正常、3875 張肺炎。
- ◆ Validation：16 張肺部影像，8 張正常、8 張肺炎。
- ◆ Test：624 張肺部影像，234 張正常、390 張肺炎。

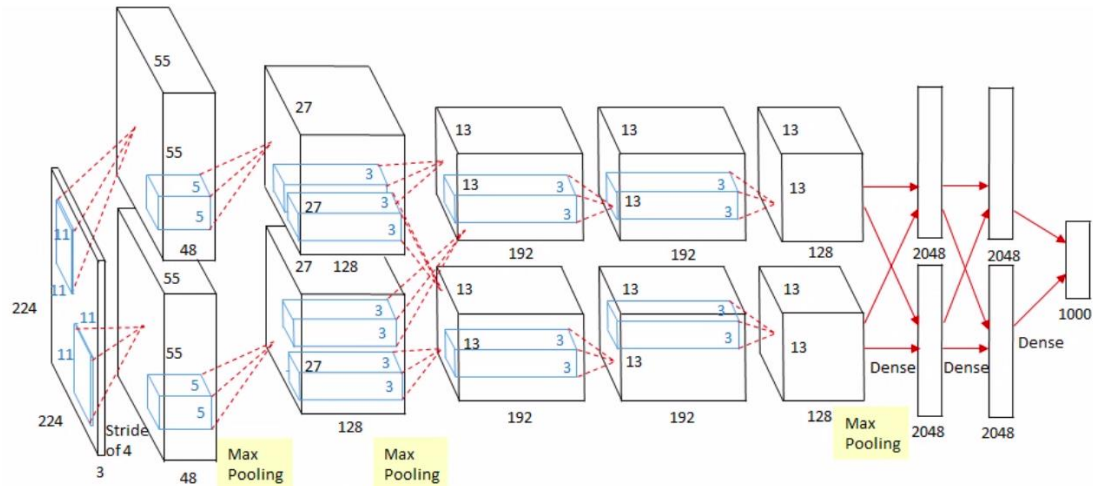
由於 Validation data 數量明顯不足，故在此次分析裡，訓練集共 5216 張影像中，85% 依舊當作訓練集(4434)，而 15% 用作模型配適時的驗證集(782)：

三、研究方法

在此報告中，我以 Alexnet 與 VGG_19 作為基底模型作修改，加入一些全連接層、BatchNormalizaion 層、dropout 層，更換 Optimizer 函數，抑或是改變批次大小等，以期能提升分類準確率，並將最後的結果作交叉比較分析。

貳、模型方法

一、 AlexNet



此模型為 Alex 於 2012 年提出的網路結構模型，在 ImageNet 大規模視覺辨識挑戰賽中以最低的 15.3% 的 Top-5 錯誤率獲得了冠軍，原論文也得出了兩個很重要的結論：

- I. CNN 模型深度對於提高辨識效能不容忽視。
- II. 圖形處理器 GPU 的活耀使得能訓練時間成本落在可接受的範圍內。

以下簡介各神經層特色：

- i. 卷積層 1：輸入圖像大小為 224x224x3；卷積核為 11x11；步長為(4, 4)；padding 為 valid，使圖根據卷積核大小和步長大小而變小；激活函數採用 ReLU；經過 pool size 為(3, 3)、步長(2, 2)的池化運算處理後；再進行局部響應歸一化(LRN)；最終輸出個數則為 96(卷積核數量)。※ $\text{ReLU}(x)=\max(x, 0)$
- ii. 卷積層 2：輸入數據大小為 27x27x96；卷積核為 5x5；步長為(1, 1)；padding 為 same，使得輸入圖像在卷積後寬高大小不變；激活函數採用 ReLU；經過 pool size 為(3, 3)、步長(2, 2)的池化運算處理後；再進行局部響應歸一化(LRN)；最終輸出個數則為 256(卷積核數量)。
- iii. 卷積層 3&4：輸出個數為 384(卷積核數量)；卷積核為 3x3；步長為(1, 1)；padding 為 same；激活函數採用 ReLU；且在這兩層中不使用 Max Pooling 與 LRN。
- iv. 卷積層 5：輸入數據大小為 13x13x384；卷積核為 3x3；步長為(1, 1)；

padding 為 same；激活函數採用 ReLU；然後經過 pool size 為(3, 3)、步長(2, 2)的池化運算處理；最終輸出個數為 256(卷積核數量)。

- v. 全連接層 6&7&8：第 6&7 層中使用 4096 個神經元，並以 ReLU 為激活函數；第 8 層中則使用 1000 個神經元，以 softmax 為激活函數；且在全連接層中加入 dropout 層以防止模型過擬合。

Alexnet 特點：

I. Local Response Normalization(局部響應歸一化 LRN)

模擬了神經生物學上一個名為「側抑制」的功能，意指被激活的神經元會抑制相鄰的神經元，以鄰近數據歸一化來達到局部抑制的作用，**使得響應較大的值相對更大**，藉此提高模型的泛化能力。

公式如下：

$$b_{x,y}^k = a_{x,y}^k / \left(k + \alpha \sum_{i=\max(0,x-n/2)}^{\min(W,x+n/2)} \sum_{j=\max(0,y-n/2)}^{\min(H,y+n/2)} (a_{i,j}^k)^2 \right)^\beta$$

重點：1. **只對數據相鄰區域執行歸一化，數據大小、維度不變**

2. **有助於減少訓練後的模型 inference 的錯誤率。**

II. 重疊池化(Overlapping Pooling)

正常池化中，pool size 大小與步長一致；而當 pool size **大於**步長大小時，便使得池化操作在部分像素上有重合，即為「重疊池化」。此舉在官方文檔中具**避免過擬合**的作用。

二、 VGG19

架構圖如下：

(16 個卷積層與 3 個全連接層)

VGG19		
序号	层结构	
1	conv1-1	1
2	relu1-1	
3	conv1-2	2
4	relu1-2	
5	pool1	
6	conv2-1	3
7	relu2-1	
8	conv2-2	4
9	relu2-2	
10	pool2	
11	conv3-1	5
12	relu3-1	
13	conv3-2	6
14	relu3-2	
15	conv3-3	7
16	relu3-3	
17	conv3-4	8
18	relu3-4	
19	pool3	
20	conv4-1	9
21	relu4-1	
22	conv4-2	10
23	relu4-2	
24	conv4-3	11
25	relu4-3	
26	conv4-4	12
27	relu4-4	
28	pool4	
29	conv5-1	13
30	relu5-1	
31	conv5-2	14
32	relu5-2	
33	conv5-3	15
34	relu5-3	
35	conv5-4	16
36	relu5-4	
37	pool5	
38	fc6(4096)	17
39	relu6	
40	fc7(4096)	18
41	relu7	
42	fc8(1000)	19
43	prob(softmax)	

VGG 是英國牛津大學 Visual Geometry Group 的縮寫，主要貢獻是使用更多的隱藏層，大量的圖片訓練，且使用相當多層的處理，以提高準確率至 90%。相比前者 Alexnet，Vgg19 以連續的幾個較小的 3x3 卷積核來代替 Alexnet 中較大的卷積核 11x11、7x7、5x5；具體而言，以三個 3x3 卷積核來代替 7x7 的卷積核、以兩個 3x3 卷積核來代替 5x5 的卷積核，目標為以多層非線性層來增加網路結構深度，藉此學習更複雜的模式，且不須使用過多參數。

其優點為：

1. 利用多個小卷積核代替大卷積核時，如：三個 3x3 卷積核以代替 7x7 卷積核時，由於網路深度加深，代表非線性函數(如 ReLU)的使用次數會跟著增加，更加提高了模型的辨識能力。
2. 減少參數的數量：多個小卷積核堆疊參數量比起單個大卷積核顯著降低。假設三個堆疊的 3x3 卷積層的輸入、輸出通道數皆為 k，參數數目即為 $3 \times (3k \times 3k) = 27k^2$ ；而對於一個 7x7 的卷積核，其參數數目則會增加到 $1 \times (7k \times 7k) = 49k^2$ 。
3. 驗證了不斷加深網路結構可以提升模型性能。

參、模型比較改動細節

Optimizer :

I. SGD(隨機梯度下降法) :

其梯度更新規則為：

一次跑單一樣本或小批次(Mini-Batch)的樣本，並算出一次或小批次梯度的平均後就更新一次，其中此處該樣本或小批次樣本為隨機抽取得來的，故稱之為「隨機」。

優點：對於大數據集而言，可能會有相似的樣本，由於 SGD 一次只進行一次更新，在計算梯度時並不會出現冗餘，故執行速度較 BGD(批量梯度下降法)快，且能夠增加新樣本。

缺點：並非每次迭代皆朝向整體最優解，訓練速度較快的代價就是準確度下降；且若學習率過大(預設 0.01)，易造成參數呈現鋸齒狀的更新。

II. RMSprop

為一種 Geoff Hinton 提出的自適應學習率發法，宗旨為解決另一算法(Adagrad)的學習率急遽下降的問題。

梯度更新規則：

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2$$

$$\Theta_{t+1} = \Theta_t - \frac{\alpha}{\sqrt{E[g^2]_t + \epsilon}} \cdot g_t$$

其中使用指數加權平均，旨在消除梯度下降中的擺動。當某一維度的導數較大時，指數加權平均就隨之變大；某一維度的導數較小時，指數加權平均就隨之變小；如此一來就保證了各維度導數都在一個量級，進而減少了擺動。

(Hinton 建議超參數設定中 $E[g^2]_{t-1}$ 權重為 0.9，而 η 則為 0.001(學習率預設參數)。)

III. Adam(Adaptive Moment Estimation)

此為另一種自適應學習率的算法。

除了如 RMSprop 一樣儲存了過去梯度二階矩陣估計 g_t 的指數衰減平均值外，亦保留了過去梯度一階矩陣估計 v_t 的指數衰減平均值：

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

並透過對期望值的不偏估計，取得校正過後的 \hat{m}_t 、 \hat{v}_t ：

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

梯度更新規則則為：

$$\Theta_{t+1} = \Theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

優點：經過偏置校正後，每次迭代學習率皆有明確範圍，故使得參數較為平穩。

(超參數設定中建議 $\beta_1 = 0.09, \beta_2 = 0.999, \epsilon = 10e - 8$ ，學習率預設為 0.001。)

肆、模型結果評估

如同一開始資料簡介所述，由於驗證集過小(16 筆資料)，故模型配適中將訓練集 85%的資料用來訓練模型(4434 筆)，而剩下的 15%(782 筆)作為模型驗證資料。

此外，在模型訓練中，固定 epoch 大小為 15，並設置表現最佳的權重組合作為 checkpoint，而非前一次 epoch 的權重組合，去進行下一次的分類預測。

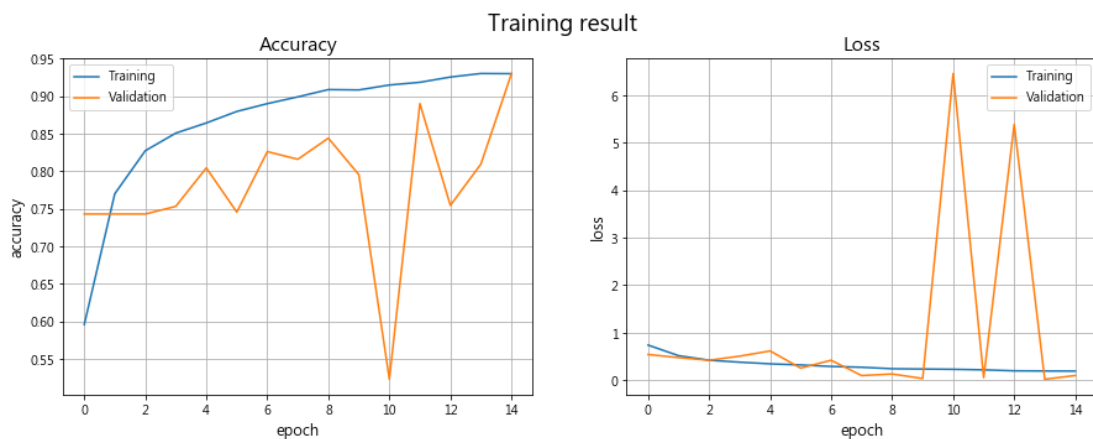
以下兩個模型各以三種方式依序去比較結果：

I. Alexnet：

除了原網路架構外，皆加入以下神經層：

1. 一層 Batchnormalization(批次正規化)。
2. 一層神經元個數為 128，激活函數為 ReLU 的全連接層。
3. 比率為 0.25 的 Dropout 層。
4. 一層 Batchnormalization。
5. 一層神經元個數為 64，激活函數為 sigmoid 的全連接層。
6. 比率為 0.2 的 Dropout 層。
7. 一層 Batchnormalization。
8. 最終輸出為神經元個數為 1，激活函數為 sigmoid 的全連接層。

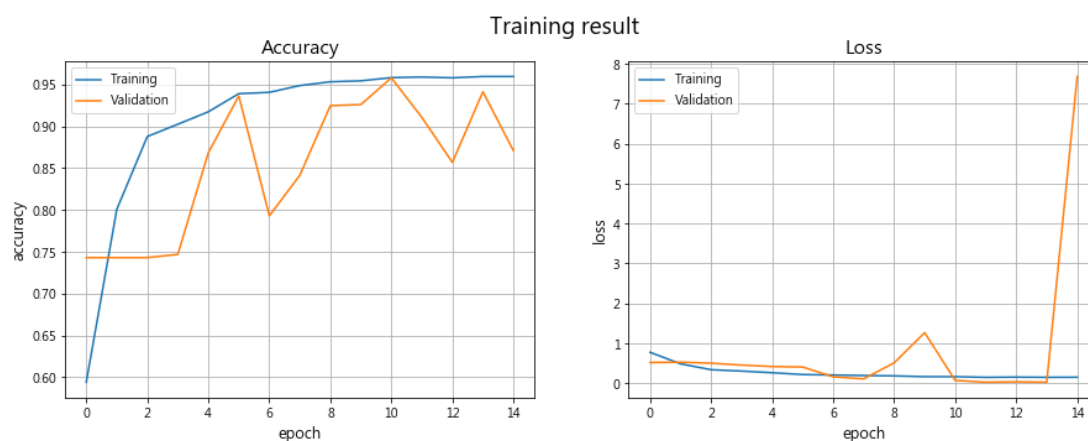
甲、Optimizer = SGD



驗證集準確率普遍低於 0.85，且到了第 9 個 epoch 後發生準確率陡降的狀況，平均震盪幅度大。

Average Training Accuracy 0.86744857
Average Validation Accuracy 0.7813299258550008

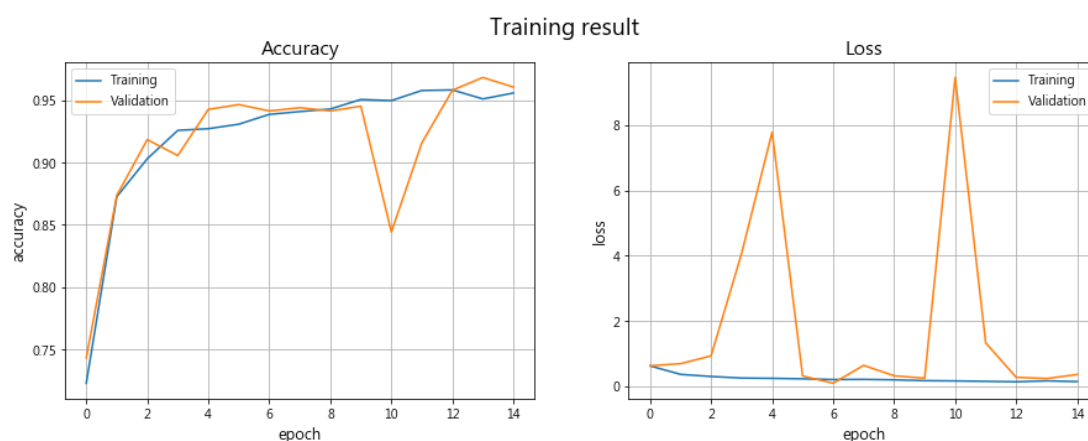
乙、Optimizer = Adam



將優化器更為 Adam 後，驗證集的準確度較前者快碰到高點(0.95 附近)，且震盪幅度也比前者要小，但仍呈現鋸齒狀。

Average Training Accuracy 0.90870553
Average Validation Accuracy 0.8534526864687602

丙、Batch-size 更新為 32(原為 64)



Average Training Accuracy 0.9216358
Average Validation Accuracy 0.916368289788564

驗證集準確率更早達到收斂，且大部分於 0.95 附近浮動，曲線亦較為平滑，唯有在 epoch 為 10 時走下坡，精準度更高。

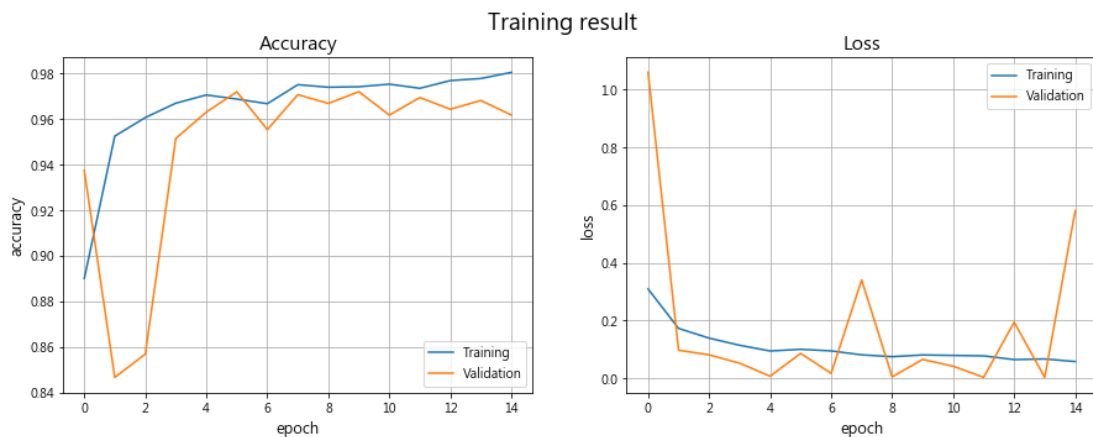
由平均訓練準確率與平均驗證準確率可以明顯觀察到，隨著優化器從 SGD 改成 Adam，再將 batch-size 從 64 更動為 32 時，兩者數值皆愈來愈高。

II. VGG19

除了原網路架構外，皆加入以下神經層：

1. 一層 Flatten 用來攤平，將多維輸入一維化。
2. 一層 Batchnormalization。
3. 一層神經元個數為 128，激活函數為 ReLU 的全連接層。
4. 比率為 0.25 的 Dropout 層。
5. 一層 Batchnormalization。
6. 一層神經元個數為 32，激活函數為 sigmoid 的全連接層。
7. 比率為 0.2 的 Dropout 層。
8. 一層 Batchnormalization。
9. 最終輸出為神經元個數為 1，激活函數為 sigmoid 的全連接層。

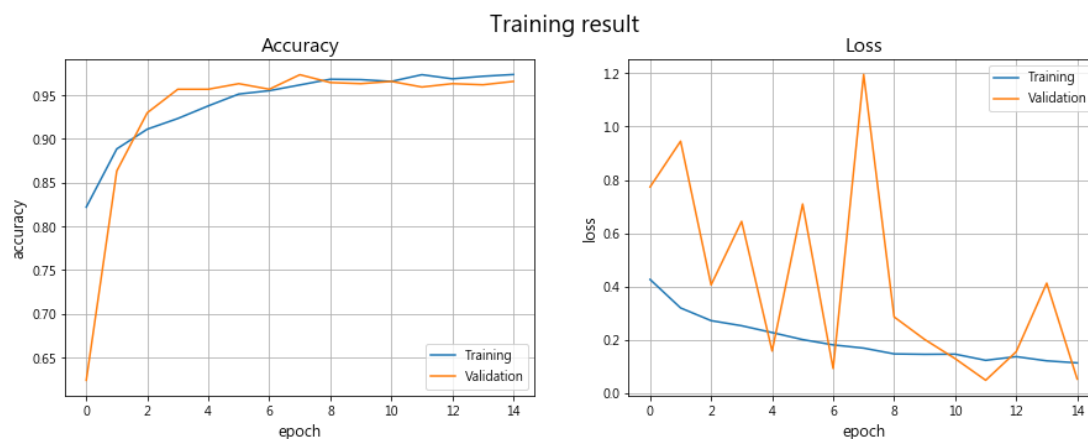
一、Optimizer = RMSprop



驗證準確率在第四個 epoch 即開始收斂在 0.97 附近，其震盪幅度小(約莫 0.01)。

Average Training Accuracy 0.9654489
Average Validation Accuracy 0.9477408369382222

二、Optimizer = Adam

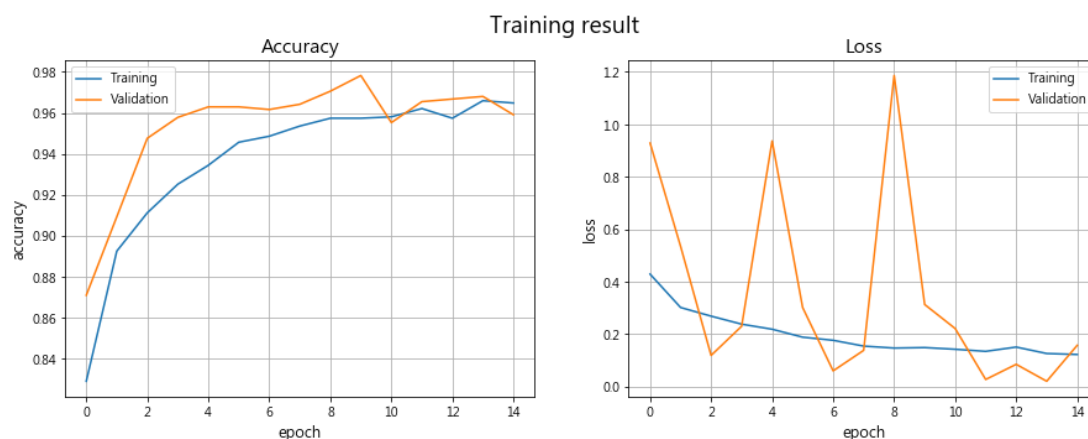


將優化器更改為 Adam 後，驗證集準確率更早(第二個 epoch)達成收斂(0.95 附近)，且震盪幅度比前者更小，曲線更加平滑；但驗證的 Binary-Entropy loss 震盪幅度變得更大。

Average Training Accuracy 0.94242966

Average Validation Accuracy 0.9309462944666544

三、Batch-size = 32(原為 64)



大部分驗證準確率比起訓練準確率還要高(除了 epoch 為 10 與 14)，且亦極早達成收斂(0.96 周遭)。

Average Training Accuracy 0.9375131

Average Validation Accuracy 0.9533674359321594

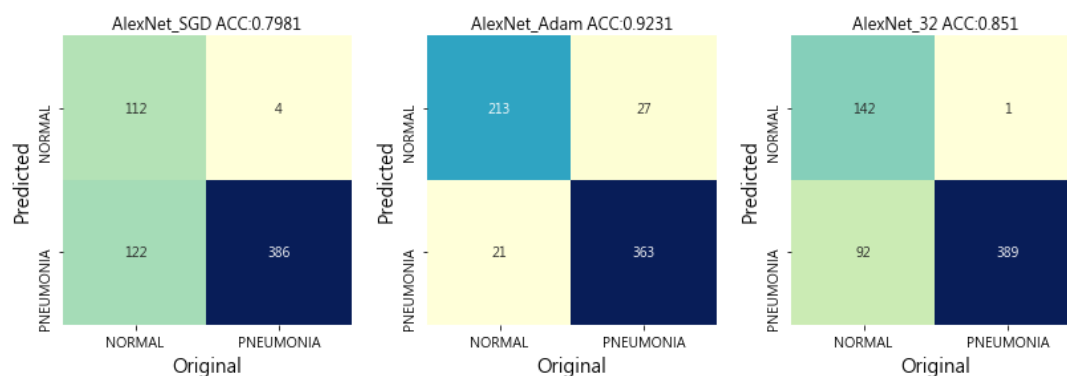
由平均訓練準確率與平均驗證準確率可以觀察到，隨著優化器從 SGD 改成 Adam，再將 batch-size 從 64 更動為 32，不同於 Alexnet，平均訓練準確率有

降低的趨勢，而 Vgg19 的驗證準確率則在 batch-size 為 32 時達到最高；兩者比起前一個模型 Alexnet 皆有長足的提升進步。

伍、結論

最後以測試集中的 624 筆資料進行模型預測比較：

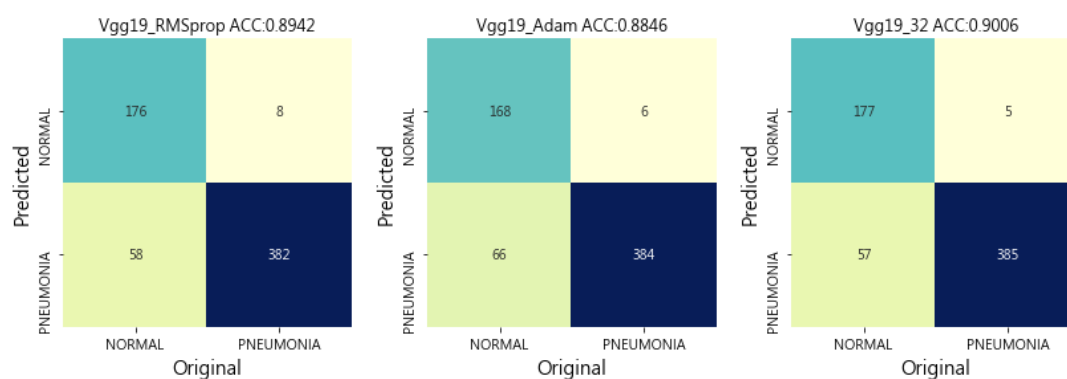
Alexnet：



	Alexnet-SGD	Alexnet-Adam	Alexnet_32
Accuracy	0.798077	0.923077	0.850962
Recall	0.989700	0.930800	0.997400
Precision	0.478600	0.910300	0.606800
F1-score	0.645200	0.920400	0.754500

綜合而言，Alexnet 將優化器設成 Adam 時能夠取得最佳的預測效果；而仔細比較的話能夠看出，Accuracy、Precision、F1-score 在優化器為 Adam，batch-size 為 64 時表現最佳，只有在批次大小改為 32 時，其召回率較其他模型來的高，代表對於肺炎患者的胸腔 X 光影像中預測出確實罹患肺炎的準確度很高。

VGG19：



	Vgg19-RSMprop	Vgg19-Adam	Vgg19_32
Accuracy	0.894231	0.884615	0.900641
Recall	0.979500	0.984600	0.987200
Precision	0.752100	0.717900	0.756400
F1-score	0.850900	0.830400	0.856500

整體而言，Vgg19 將優化器設成 Adam，Batch-size 設為 32 時能夠取得最佳的預測效果，四項精度皆最高；值得注意的是將優化器設為 Adam，然而批次大小仍為 64 時，其 Accuracy、Precision、F1-score 皆為模型當中表現最差的。

陸、References

<https://blog.csdn.net/u010089444/article/details/76725843>

<https://www.cnblogs.com/guoyaohua/p/8542554.html>

<https://keras.io/zh/optimizers/>

<https://runder.io/optimizing-gradient-descent/index.html>

<https://zhuanlan.zhihu.com/p/27066570>