

多重神經網路對COVID-19 CT影像處理反映結果的試驗

夏至賢¹ 謝奇容² 王采筑²

國立宜蘭大學資訊工程學系 教授¹ 國立宜蘭大學資訊工程學系 學士生²

摘要

嚴重特殊傳染性肺炎（COVID-19）在2019年末於中華人民共和國湖北省武漢市首次被發現，隨後在2020年初迅速擴散至全球多國，逐漸變成一場全球性大瘟疫。[1]截至2021/12/26，全球已有278,443,343名感染者，死亡病例數5,405,669人，全球致死率1.94%[2]。

疫情對於全球的經濟及民生具有重大影響，因此迅速而正確分辨患者是在疫情中的首要任務。Ai T. 等人 [3]發現一個事實，42% 的患者的胸部 CT 掃描可以幫助檢測患者在進行 RT-PCR 之前的病情改善[4]，是一個有效且低風險的手段。

近年來，醫療影像的大數據分析越趨流行，也有臨床應用的例子[5]，因此在這裡，我們想透過影像處理以及大數據分析進行監督式學習，分辨CT影像是否感染COVID-19。

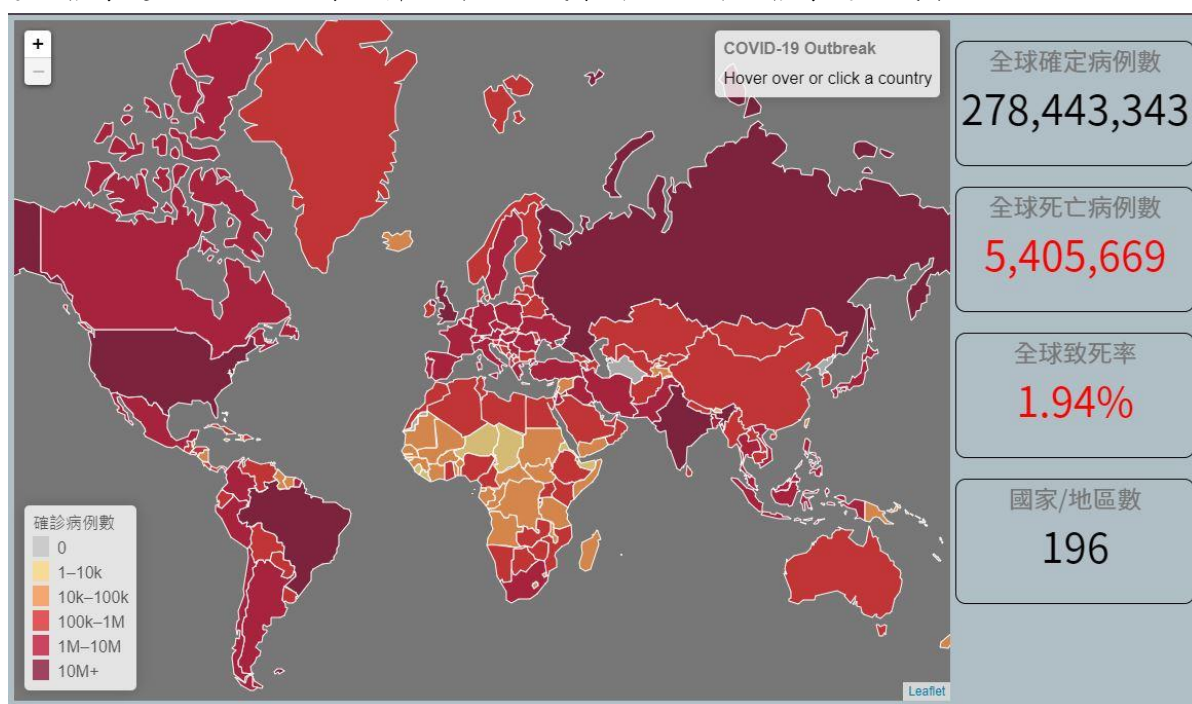


圖1、2021/12/26全球感染及死亡統計

關鍵字: 影像處理、醫療影像、監督式學習

A trial of multiple neural networks reflecting the results of COVID-19 CT image processing

C.-H. Hsia¹, Professor, Department of Information Engineering, National Yilan University

CHI-JUNG HSIEH², Bachelor student, Department of Information Engineering, National Yilan University

Tsai-Chu Wan³, Bachelor student, Department of Information Engineering, National Yilan University

ABSTRACT

Severe Special Infectious Pneumonia (COVID-19) was first detected in Wuhan City, Hubei Province, People's Republic of China at the end of 2019, and then quickly spread to many countries around the world in early 2020, gradually turning into a global pandemic. [1] As of 2021/12, there have been 278,443,343 infected people worldwide, with 5,405,669 deaths, and a global fatality rate of 1.94% (this statistics is as of 2021/12/26[2]).

The epidemic has a major impact on the global economy and people's livelihood, so quickly and correctly distinguishing patients is the first task in the epidemic. Ai T et al. [3] discovered the fact that 42% of patients' chest CT scan can help detect the patient's improvement before undergoing RT-PCR [4], which is an effective and low-risk method before undergoing RT-PCR. means.

In recent years, the big data analysis of medical imaging has become more and more popular, and there are also examples of clinical applications [5]. Therefore, here, we want to use image processing and big data analysis to conduct supervised learning through CT images to distinguish whether they are infected with COVID-19.

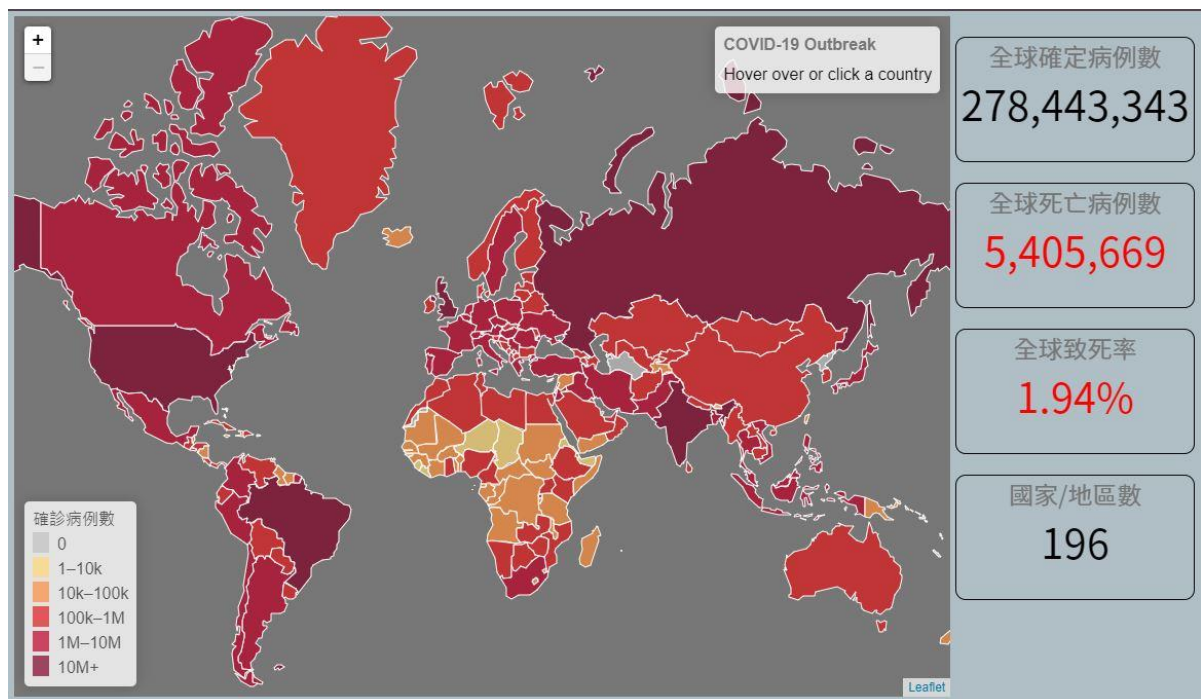


Figure 1, 2021/12/26 global infection and death statistics

Keywords: image processing, medical imaging, supervised learning

一、導論

自 1895 年 Wilhelm Conrad Roentgen 發現 X 射線輻射以來，醫學成像領域已經發展成為一門龐大的科學學科。透過當前圖像模式獲取患者的數據進行分析，例如電腦斷層掃描（CT）、磁共振斷層掃描（MRI）、正電子發射斷層掃描（PET）或超聲波檢查（US），為診斷提供了前所未有的機會、治療計劃和治療評估。醫學圖像處理對於利用這種不斷增加的數據量，並以適合特定醫學任務的方式探索和呈現所包含的信息是至關重要。[6]

如上述，我們所達成的特定醫學任務即是**透過肺部CT影像分析是否有感染COVID-19**，CT影像可用於判斷患者是否感染病毒性肺炎（COVID-19是一種由SARS-CoV-2病毒引起的病毒性肺炎）。但是CT 無法確定導致病毒性肺炎的病毒是 SARS-CoV-2 還是其他病毒。[7]

嚴格來說，CT 不能用於確認患者是否感染了 COVID-19。然而在疫情爆發期間，大多數病毒性肺炎是由 SARS-CoV-2 引起的。也就是說，如果患者根據CT結果確診為病毒性肺炎，這種病毒性肺炎極有可能是COVID-19。由於這一事實，CT 被認為可用於在 COVID-19 爆發期間進行診斷。

為此我們需要進行一系列手續對影像進行處理，再進行監督式學習。幸運的是，雖然COVID-19被發現於2019的12月，短短兩年不斷有人對這方面進行研究，前人對此進行許多試驗，包括不同的前處理方式及學習方式。得益於此，我們了解許多方式去進行CT影像的處理，先透過IEEE的論文了解實作的可能及方向去發想，再搜尋開源的資源去進行實驗。

最初我們的動機及主要概念來自於"Medical Image Processing and Deep Learning to Diagnose COVID-19 with CT Images"[8]一篇，其中詳細敘述他的作法以及來源。很遺憾的，關於模型以及部分處理方法並沒有開放程式碼，但不可否認其中仍然詳細的寫下方法的內容跟原理，我們就是從這點去思考有什麼方法可以達到目的及如何實現。

對於這項任務，經過教授的建議，我們最終考量醫學影像這個標籤，在各個步驟選擇相對適合醫學影像的處理方式，基於這個思路去構築我們處理影像的方法，步驟如圖2。

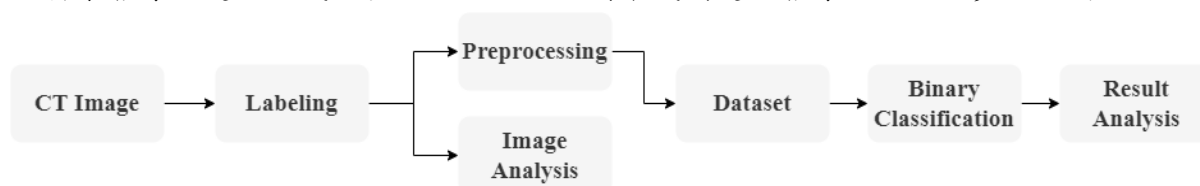


圖2、COVID-19 CT影像處理流程

這是一個很簡便的流程，實作上的難點在於前處理，經由算法增強影像的特徵，以方便我們去判斷及分類。事實上，濾波跟影像增強是非常重要的環，可以將它視為去蕪存菁，其中運用到的算法也是一門很深的學問。

在二元分類問題當中，監督式學習是常用的一種手段，如果特指CT影像的二元分類，那麼在許多問題當中，CNN及其變化型也十分常見，例如“Coronavirus disease 2019 deep learning models: Methodologic considerations” [9]中使用以CNN為底的VGG-19(91%)，和Inception_V2，又或是“A Novel Medical Diagnosis model for COVID-19 infection detection based on Deep Features and Bayesian Optimization” [10]中高達97.14%準確率的混合CNN模型。事實上，以CNN為主的COVID-19 CT影像二元分類甚至被認為可以進行臨床試驗。

結合教授的建議，我們根據影像的類型去選擇適合的神經網路進行試驗，最後將實作結果進行分析跟比較，即是我們在這篇投稿中的目的。在這份投稿中的試驗會結合處理過程進行分析，最後得出我們的結論。

二、資料集

關於COVID-19 CT資料集，網路上已有開源的資料，這邊我們使用來自Github上的開源資料集[11]。該數據集的實用性已得到中國武漢同濟醫院的一名高級放射科醫生的證實，他在 1 月至 4 月間該疾病爆發時期對大量 COVID-19 患者進行了診斷和治療。

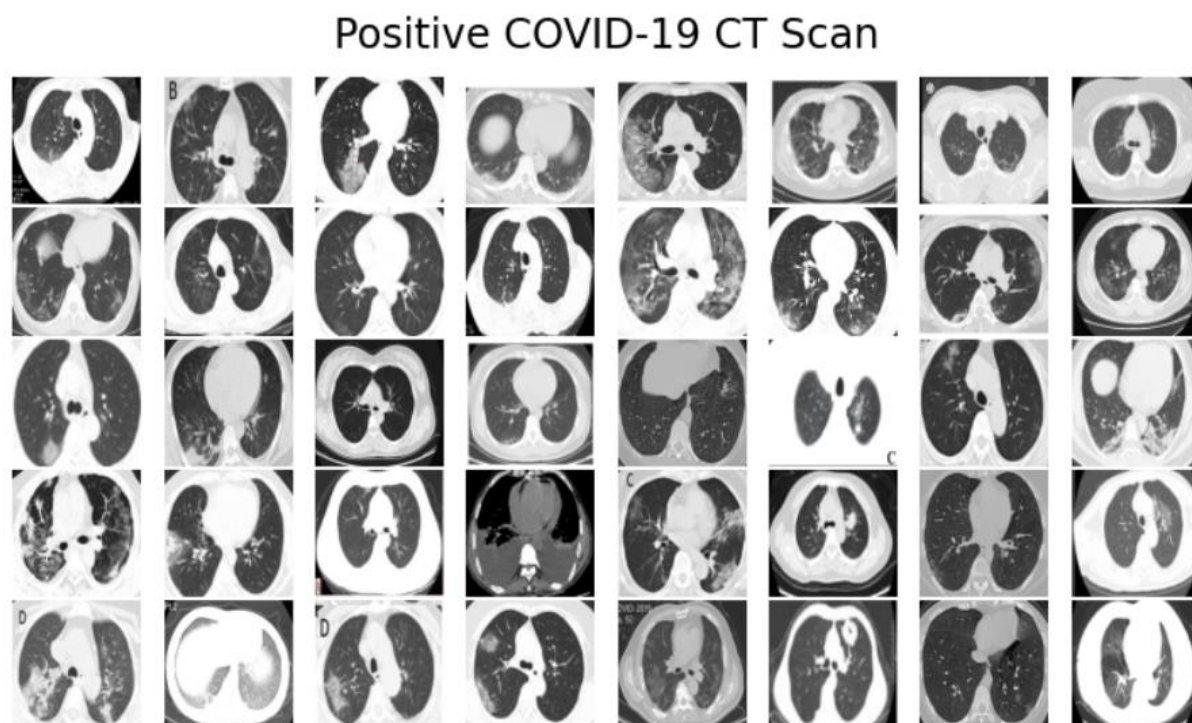


圖3、COVID-CT的影像(Positive COVID-19 CT Scan)

Negative COVID-19 CT Scan

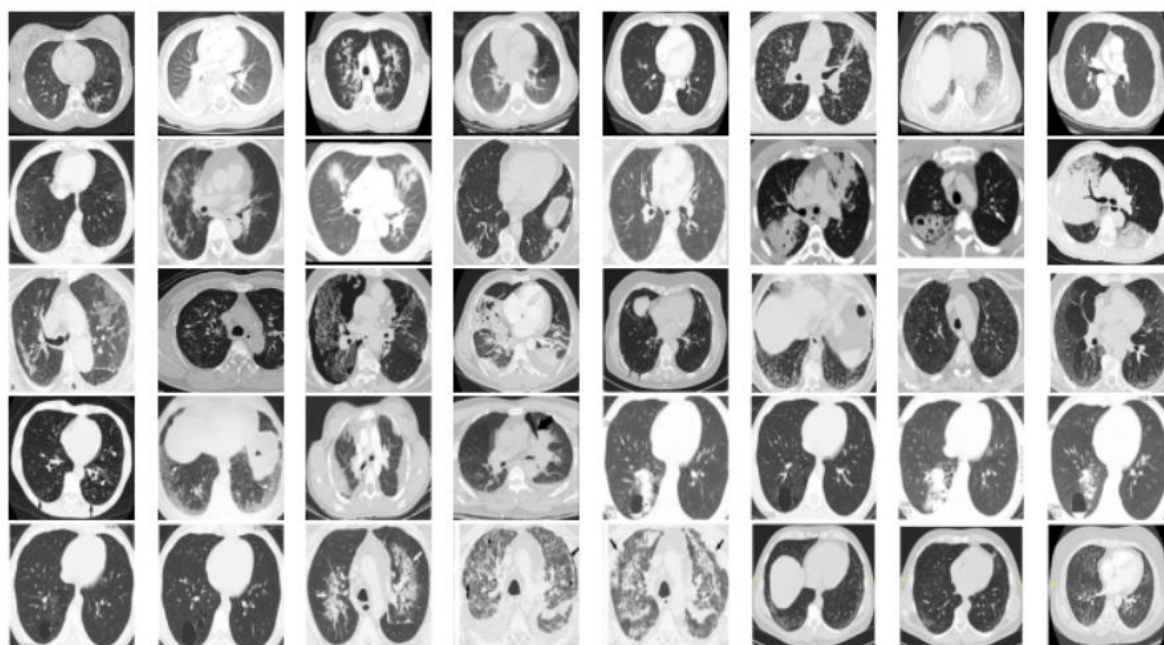


圖4、COVID-CT的影像(Negative COVID-19 CT Scan)

該資料集包含349個COVID-19 CT影像以及397個Non-COVID-19 CT影像，共746個，下圖是其中組成的圓餅圖。



圖5、資料集組成

三、前處理

在前處理中，最重要工作便是增強影像以及濾除雜訊，為此我們會進行一系列處理，讓CT影像更加清晰，以便在後續進行監督式學習的時候能夠更好的抓取特徵。

根據"Medical Image Processing and Deep Learning to Diagnose COVID-19 with CT Images"的方法，我們整理出如圖6的前處理流程，下面我們再根據各個流程進行說明。

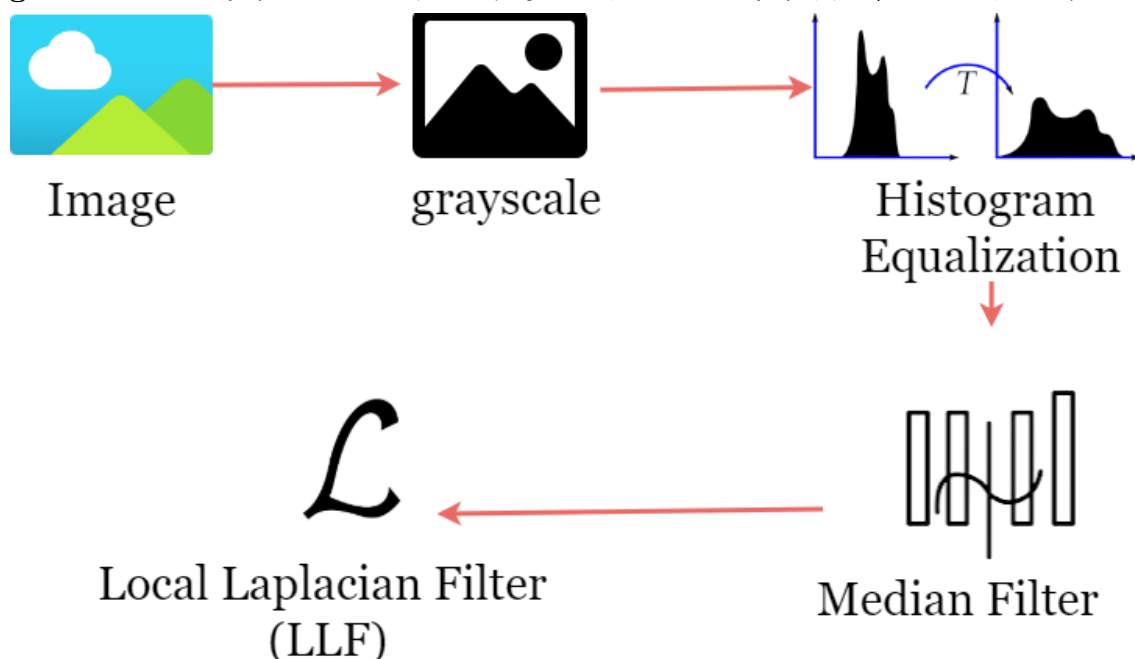


圖6、前處理流程

Positive COVID-19 CT Scan

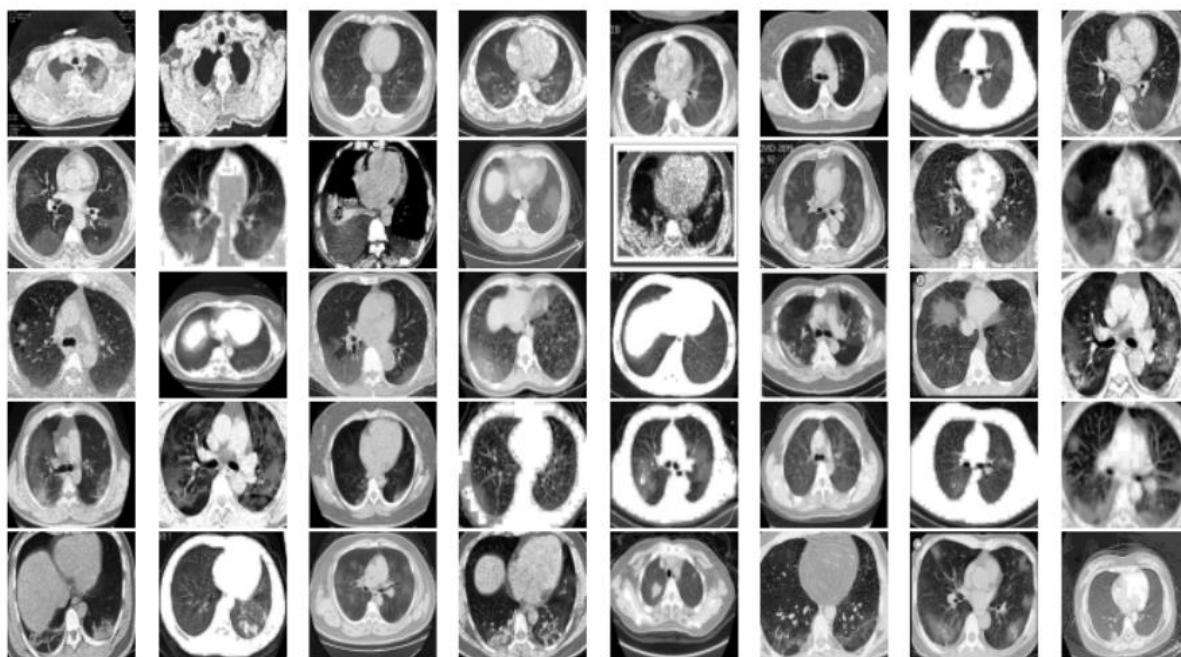


圖7、前處理後的COVID-CT的影像(Positive COVID-19 CT Scan)

Negative COVID-19 CT Scan

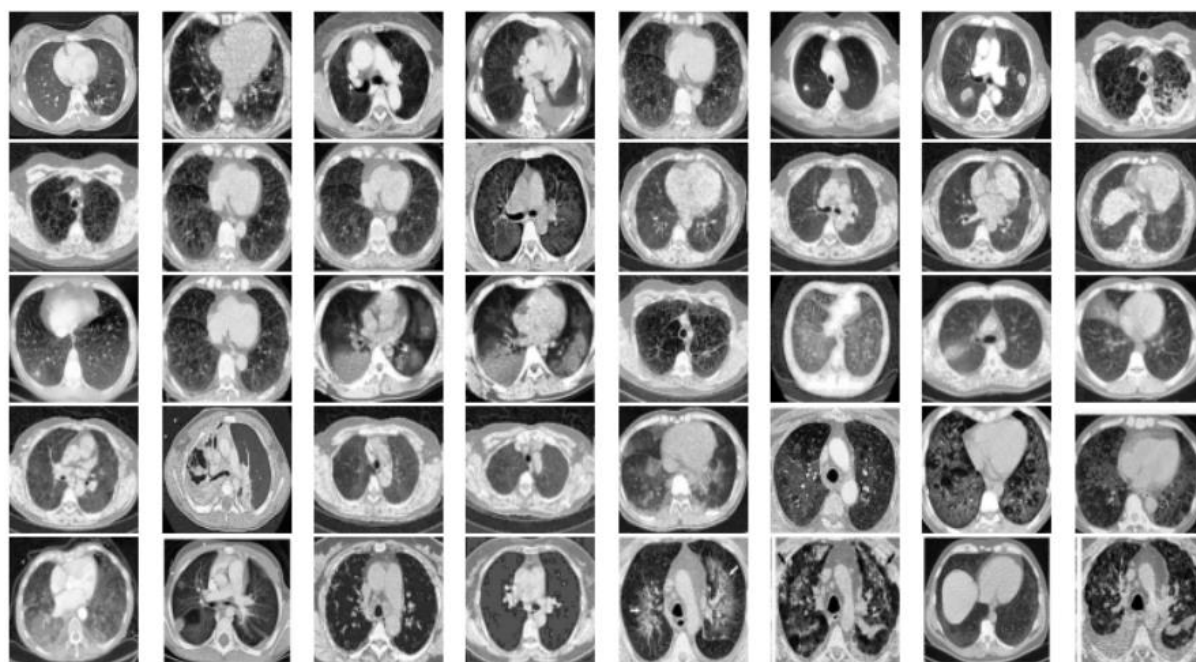


圖8、前處理後的COVID-CT的影像(Negative COVID-19 CT Scan)

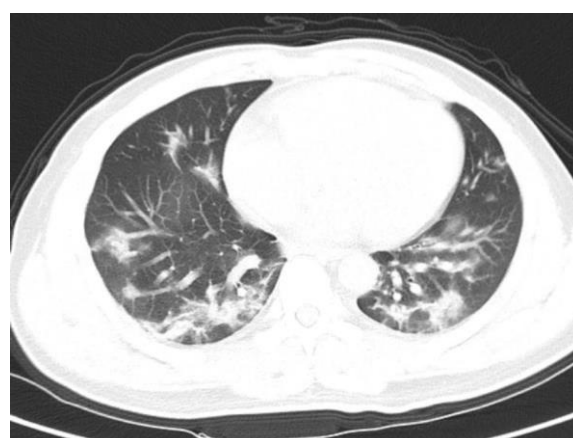
3-1. 二值化

基本的圖像分割方法，將各像素(pixel)的灰階值轉化，最後由灰度圖像傳為二值圖像。根據閾值選取的不同，二值化的算法分為固定閾值和自適應閾值。比較常用的二值化方法則有：雙峰法、P參數法、迭代法和OTSU法等。[12]

實作上，我們使用OpenCV的cvtColor()函數，在實際意義上，它是將圖像轉到不同色彩空間的函數，透過code函數決定轉換的目標色彩空間。這邊我們使用cv2.COLOR_BGR2GRAY參數進行轉換，效果如表1。



原圖



二值化影像

表1、二值化前後對比

3-2. 直方圖等化

直方圖均衡化是圖像處理領域中利用圖像直方圖對對比度進行調整的方法[13]。

考慮一個離散的灰階影像 $\{x\}$ ，讓 n_i 表示灰階 i 出現的次數，這樣圖像中灰階為 i 的像素的出現概率是

$$p_x(i) = p(x = i) = \frac{n_i}{n}, \quad 0 \leq i < L$$

式(1)、灰階為 i 的像素的出現概率

L 是圖像中所有的灰階數， n 是圖像中所有的像素數， $p_x(i)$ 實際上是像素值為 i 的圖像之直方圖，歸一化到 $[0,1]$ 。

把對應於 p_x 的累積分布函數，定義為：

$$cdf_x(i) = \sum_{j=0}^i p_x(j)$$

式(2)、圖像的累計歸一化直方圖

運用累積分布函數(CDF)對灰度值進行調整以實現對比度增強,把原始圖像的灰度直方圖從比較集中的某個灰度區間變成在全部灰度範圍內的均勻分佈[14]。

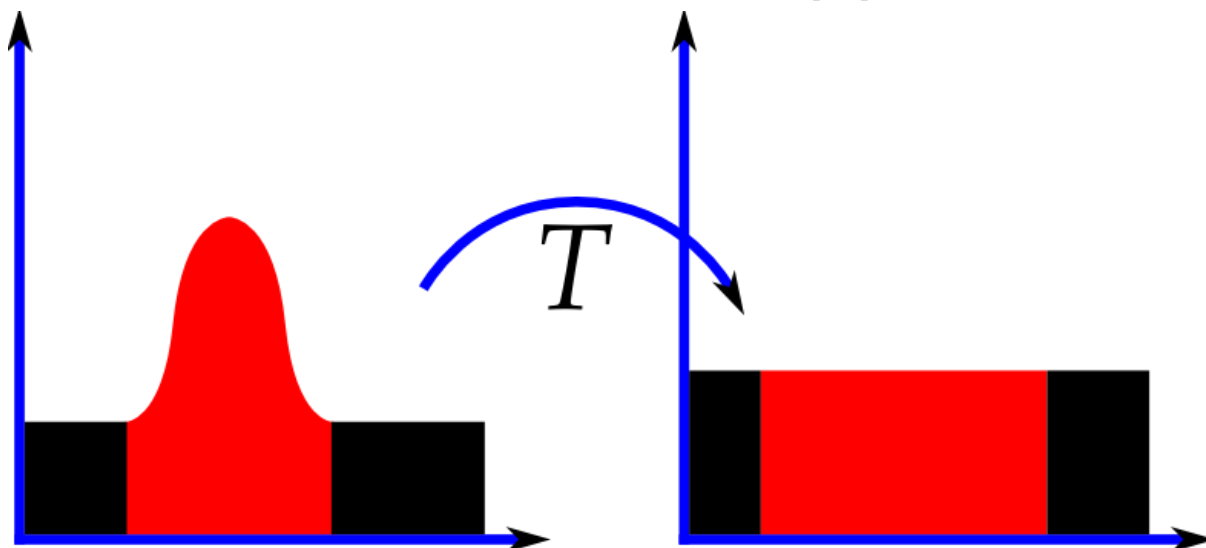


圖9、Histogrammeinebnung [15]

實作上我們使用OpenCV的`equalizeHist()`進行直方圖等化，結果如表2。



二值化影像



直方圖等化影像

表2、直方圖等化影像前後對比

3-3. 中值濾波

中值濾波實際上是做濾除雜訊用，雜訊會干擾影像特徵的擷取，因此濾除雜訊是很重要的工作。

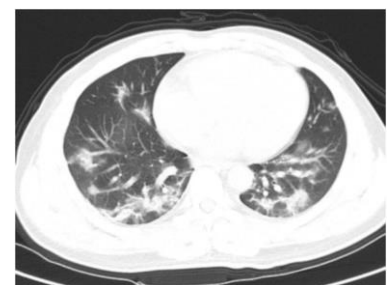
事實上，影像降噪的方式有很多種，與中值濾波相對的，還有均值濾波(Average filter)以及Laplace and Gaussian 濾波等，之所以選用中值濾波是因為其保留邊緣的特性，這也是在後續分類問題一個重要的因素，其比較如表3。



Medium_filter



Laplacian of Gaussian



Average_filter

表3、濾波對比圖

這邊我們實作使用OpenCV的medianBlur()[16]，其中有一個參數ksize，這決定內核大小，一般式3或是5，此時圖像深度應為CV_8U、CV_16U或CV_32F，這邊我們使用ksize=3。

3-4. 局部拉普拉斯濾波

事實上，它是邊緣感知圖像處理的一種方法[17]。簡單的說，就是需要遍歷所有高斯金字塔圖像中的所有像素，根據每個像素的像素值，都由原圖和某個映射函數重新計算出一個和原圖一樣大小的圖像，然後計算該圖像的拉普拉斯金字塔[18]。

實作上我們參考Github上面的Fast_LLF_python[19]，以其中的llf.py為主進行實作，效果如圖10。

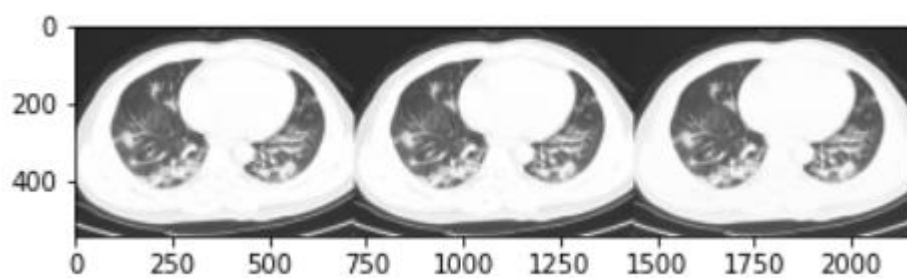


圖10、局部拉普拉斯濾波處理圖像

四、影像分析

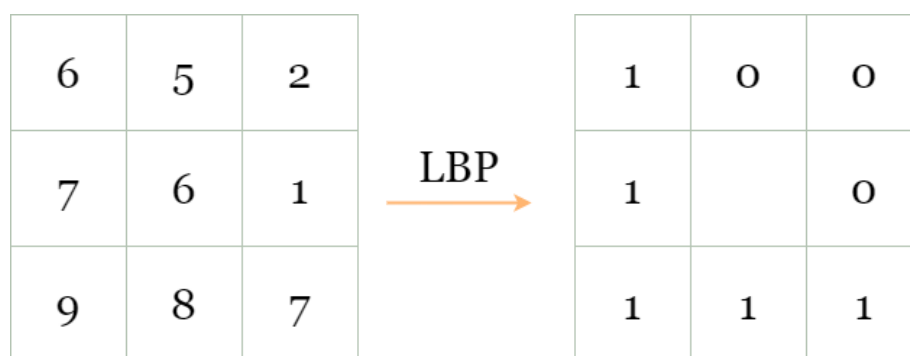
4-1.局部二值模式

局部二進制模式，或簡稱 LBP，是一種紋理描述符，由 Ojala 等人2002 年的論文"Multiresolution Grayscale and Rotation Invariant Texture Classification with Local Binary Patterns"開始流行。

局部二進制模式（LBP）是一種簡單但非常有效的紋理運算符，它通過對每個像素的鄰域進行閾值處理來標記圖像的像素，並將結果視為二進制數。由於其判別能力和計算簡單性，LBP 紋理算子已成為各種應用中的流行方法。它可以被看作是對紋理分析的傳統發散統計和結構模型的統一方法。[20]

為了讓特徵更加明顯，我們選用這種方式。在當下，我們使用紋理分類的 LBP 方法，圖像中 LBP 代碼的出現被收集到一個直方圖中，然後通過計算簡單的直方圖相似度來執行分類。

傳統的LBP編碼易於理解：（見圖9），以 3×3 為單位，進行類似二值化的標記，部分人稱為方形，這是相對於"圓形"而言。



Pattern=11110001

圖11、傳統LBP算法的變換

近年來學者們針對LBP作了一些改良，不使用方形而採用「圓形」的LBP Mask。所謂的「圓形」是以中心像素點為圓心，半徑為R pixels畫圓，於該圓範圍內總共選取P個點作為採樣點，再利用前述的處理方式來得到LBP Mask值。[21]

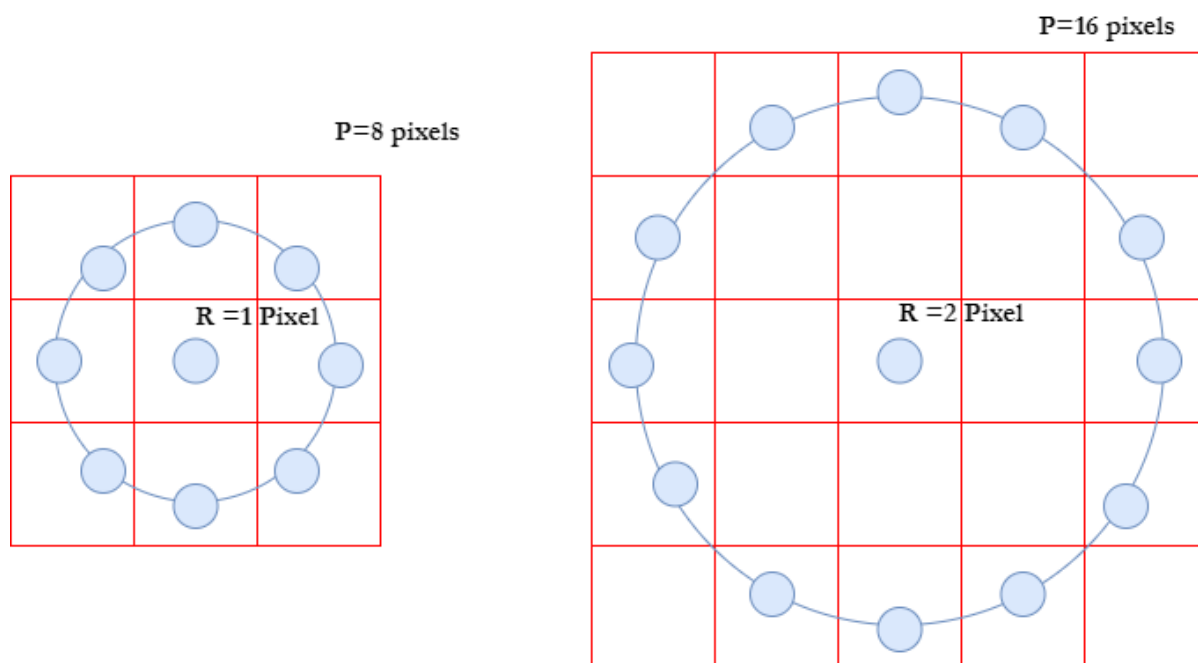


圖12、圓形LBP算法

這邊我們參考Github上面的開源程式進行實驗[22]，結果如表4。

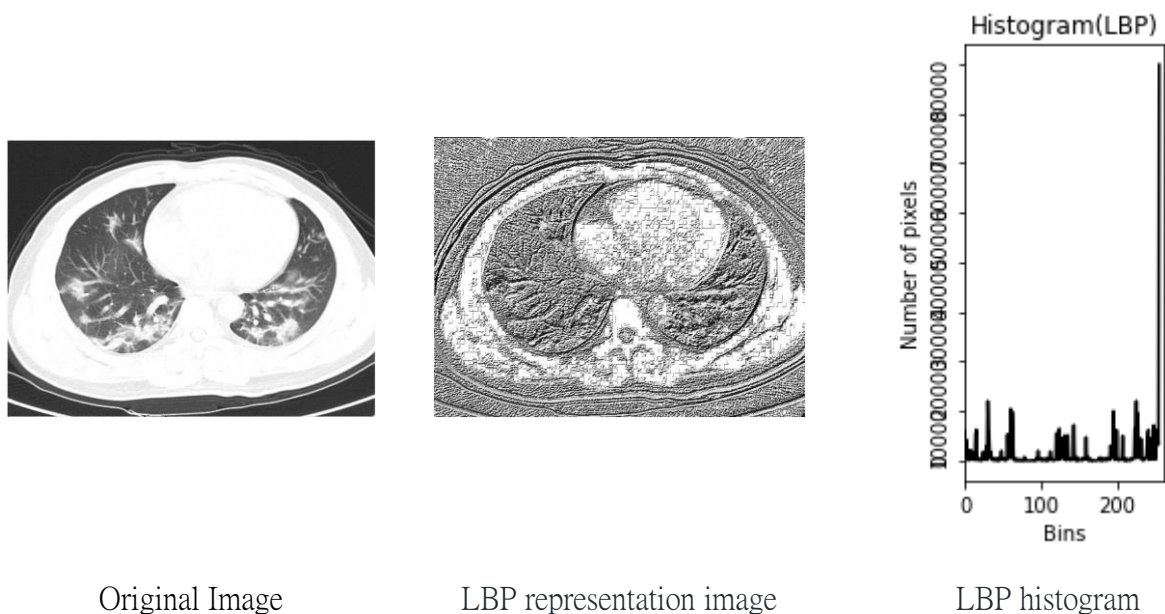


表4、LBP實驗結果

4-2. 灰度共生矩陣

[23]灰度共生矩陣(GLCM)的統計方法是20世紀70年代初由R. Haralick等人提出的，它是在假定圖像中各像素間的空間分佈關係包含了圖像紋理信息的前提下，提出的具有廣泛性的紋理分析方法。

灰度共生矩陣被定義為從灰度為 i 的像素點出發，離開某個固定位置的點上灰度值為 j 的概率，即，所有估計的值可以表示成一個矩陣的形式，因此被稱為灰度共生矩陣。對於紋理變化緩慢的圖像。

由於灰度共生矩陣的數據量較大，一般不直接作為區分紋理的特徵，而是基於它構建的一些統計量作為紋理分類特徵。Haralick曾提出了14種基於灰度共生矩陣計算出來的統計量，即：能量、熵、對比度、均勻性、相關性、方差、和平均、和方差、和熵、差方差、差平均、差熵、相關信息測度以及最大相關係數。

這邊我們參照"Medical Image Processing and Deep Learning to Diagnose COVID-19 with CT Images"[8]，節選四個統計量作為我們分析的依據：相關性、能量、對比度以及熵。

我們參考CSDN博客上分享的<https://blog.csdn.net/u013066730/article/details/109776522>程式去實作出灰度共生矩陣的程式，程式中先找出整張圖像中最大的灰階值，接著產生灰度共生矩陣，再根據矩陣計算出相關性、能量、對比度及熵。



圖13、試驗圖像

4-2-1. 產生灰度共生矩陣

產生灰度共生矩陣的流程，先決定一個最大灰度級數，並且找出最大灰階，之後遍歷每一個像素統計灰階值產生灰階矩陣，這邊我們分別用16跟32當作最大灰度級數，產生灰階矩陣的熱力圖進行比較(圖14、圖15)。

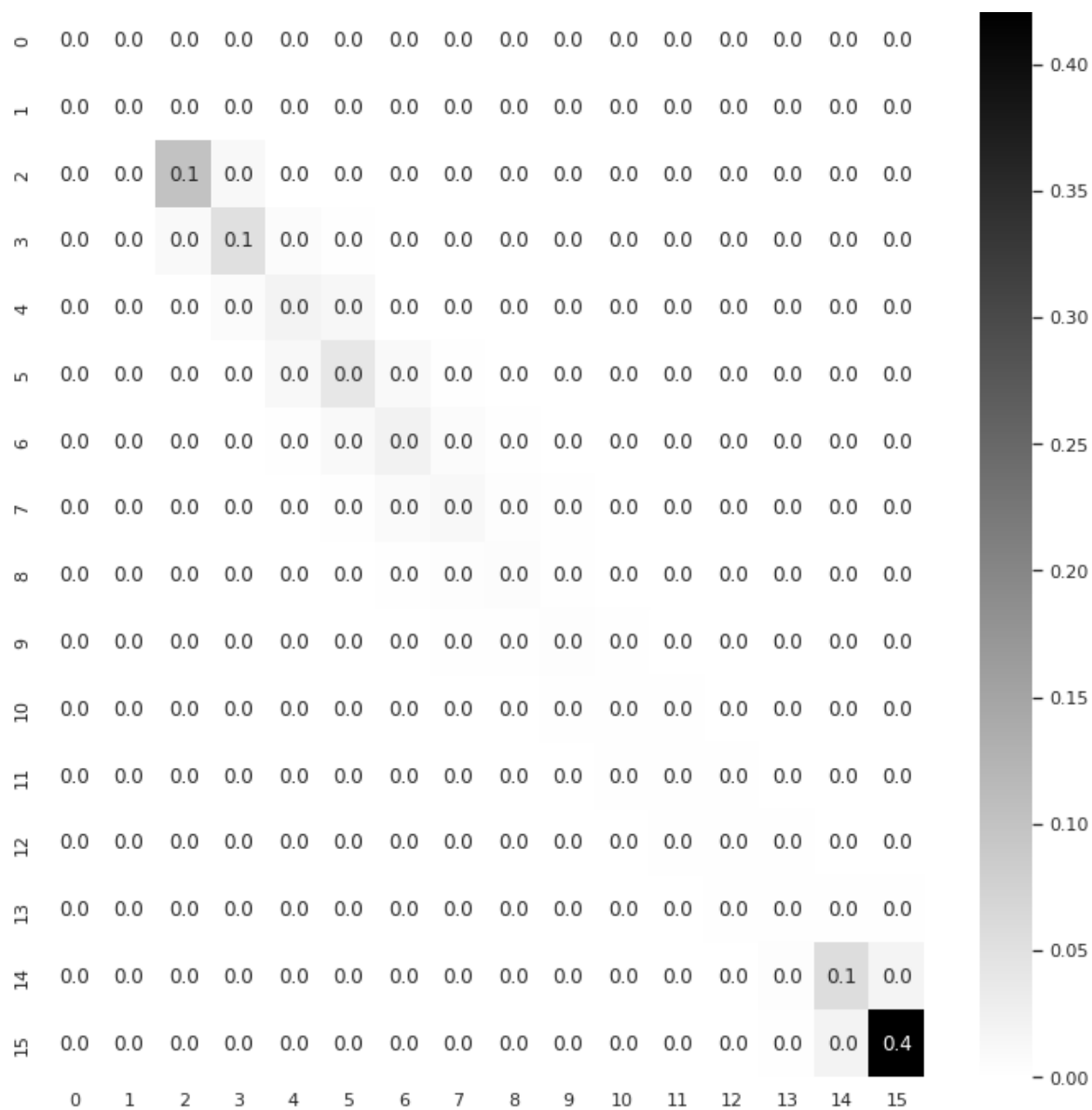


圖14、最大灰度級數為16的灰度共生矩陣熱力圖

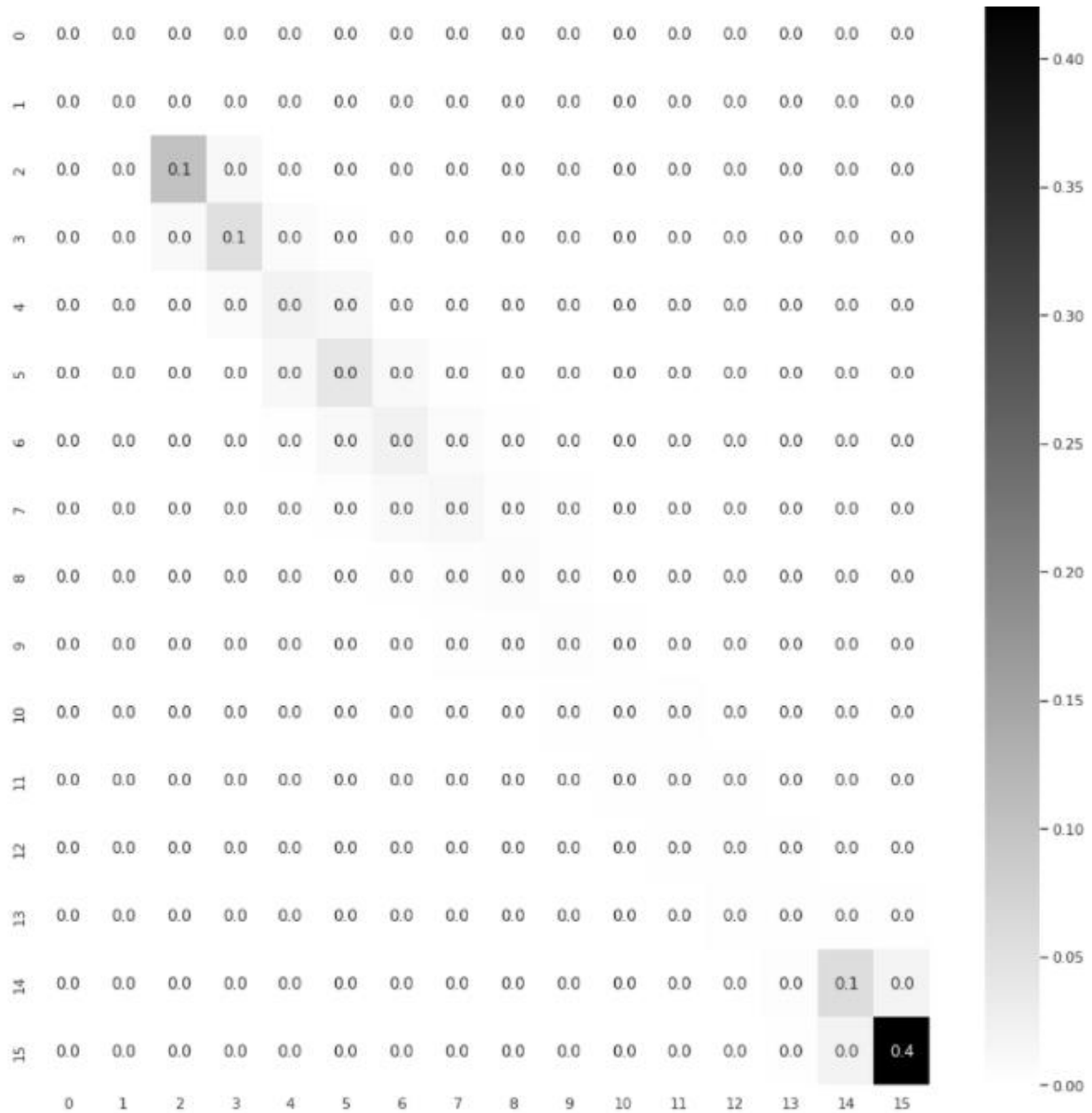


圖15、最大灰度級數為32的灰度共生矩陣熱力圖

4-2-2. 灰度共生矩陣的相關性

灰度共生矩陣用來度量圖像的灰度級在行或列方向上的相似程度，因此值的大小反應了局部灰度相關性，值越大，相關性也越大。

$$\text{Correlation} = \frac{\sum_{i=1}^N \sum_{j=1}^N (i - \bar{x})(j - \bar{y})P(i, j)}{\sigma_x \sigma_y}$$

式(3)、灰度共生矩陣的相關性[23]

4-2-3. 灰度共生矩陣的能量

能量變換反映了圖像灰度分佈均勻程度和紋理粗細度。若灰度共生矩陣的元素值相近，則能量較小，表示紋理細緻；若其中一些值大，而其它值小，則能量值較大。能量值大表明一種較均一和規則變化的紋理模式。

$$\text{Energy} \quad \sum_{i=1}^N \sum_{j=1}^N P(i, j)^2$$

式(4)、灰度共生矩陣的能量[23]

4-2-4. 灰度共生矩陣的對比度

度量矩陣的值是如何分佈和圖像中局部變化的多少，反應了圖像的清晰度和紋理的溝紋深淺。紋理的溝紋越深，反差越大，效果越清晰；反之，對比值小，則溝紋淺，效果模糊。

$$\text{Contrast} \quad \sum_{i=1}^N \sum_{j=1}^N (i - j)^2 P(i, j)$$

式(5)、灰度共生矩陣的對比度[23]

在我們實作中，對比度為0.9181790123456792，這意味圖像具有明顯的紋理。

4-2-5. 灰度共生矩陣的熵

圖像包含信息量的隨機性度量。當共生矩陣中所有值均相等或者像素值表現出最大的隨機性時，熵最大；因此熵值表明了圖像灰度分佈的複雜程度，熵值越大，圖像越複雜。

$$\text{Entropy} \quad - \sum_{i=1}^N \sum_{j=1}^N P(i, j) \lg P(i, j)$$

式(5)、灰度共生矩陣的熵[23]

在我們實作中，熵為2.704669027671504。

五、模型架構

這邊使用五種網路去進行分析跟比較，分別是InceptionV3, VGG, ResNet50, Xception[24]以及DenseNet169。

之所以使用這五種網路，是基於影像種類以及問題去思考什麼樣的網路較適合，進行實驗後得出結果並分析。關於這個問題，有許多作者給出了他們的解答，然而根據處理的方式不同，也會有不一樣的結果。意即，影像處理對於細節影響重大，不同的處理過程也會有影響，不單是只有模型而已。

這邊我們以"The Multimodal Deep Learning for Diagnosing COVID-19 Pneumonia from Chest CTScan and X-Ray Images"[25]為舉例，這篇提到我們在其他論文(例:[26])中也經常針對該問題使用的方法，所以我們節選這五個網路進行實驗。

在這五個網路當中，VGGNet、Inception（當時是V1）以及ResNets 都曾在ImageNet 競賽獲得冠軍過，這也是選他們的其中一個原因。Keras把它們都收錄進框架內，稱為Keras Applications，當然也不只這些，我們可以在keras官方文檔中找到他們的內容。

程式碼的部分，InceptionV3, VGG, ResNet50, Xception會在載入資料後輸出下列幾項結果供我們參考：

1. 混淆矩陣
2. Classifier Report
3. RUC Diagram
4. .h5 file and .h5fd file
5. Accuracy Diagram
6. Loss Diagram

透過這些資料去分析，我們將利用輸出的.h5檔案繪製出架構圖。

5-1. InceptionV3

InceptionV3是一個48層深的卷積神經網絡。

[27]InceptionV3 跟 InceptionV2 出自於同一篇論文，發表於2015年12月，但是相較Inception V2有兩處最大的不同：一是引入Factoriation into small convolutions 的思想：將 $n \times n$ 拆成 $n \times 1$ 和 $1 \times n$ (圖14)，這種做法可以節約大量參數、加速運算並減輕過擬合，這種非對稱型捲積結構拆分相較於對稱的小卷積合，特徵更加明顯，也有處理更多特徵的能力。

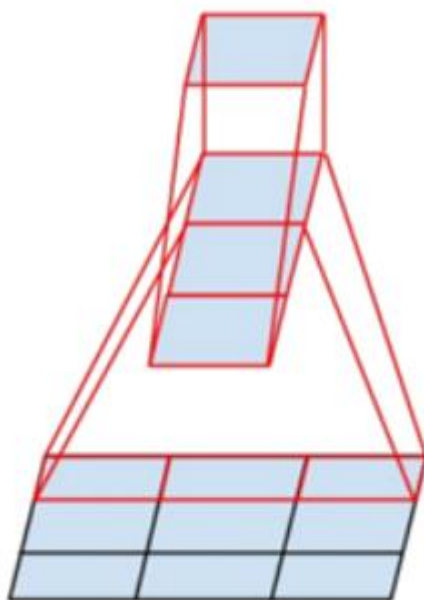


圖16、非對稱型捲積結構拆分[27]

另外一項重要的改動是分支，在Inception V3，前面的部分是普通的卷積層，後面則是Inception Module。在Inception Module中，具有分支，並且在這些分支之中也還有分支(圖17)。

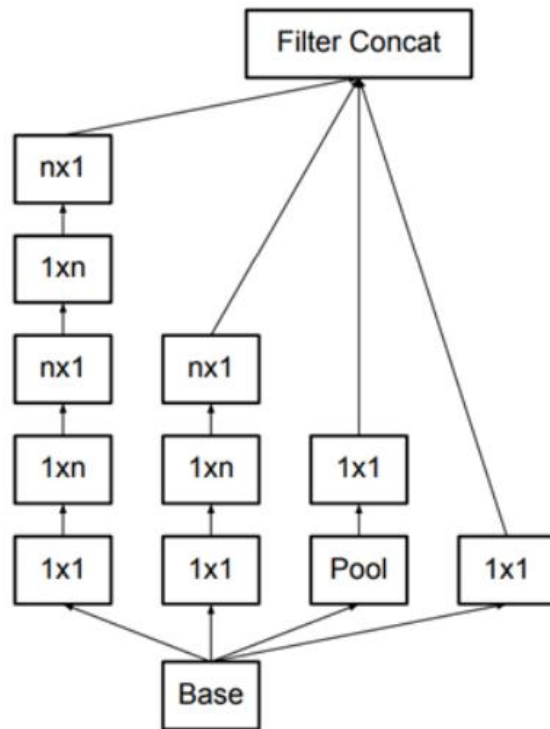
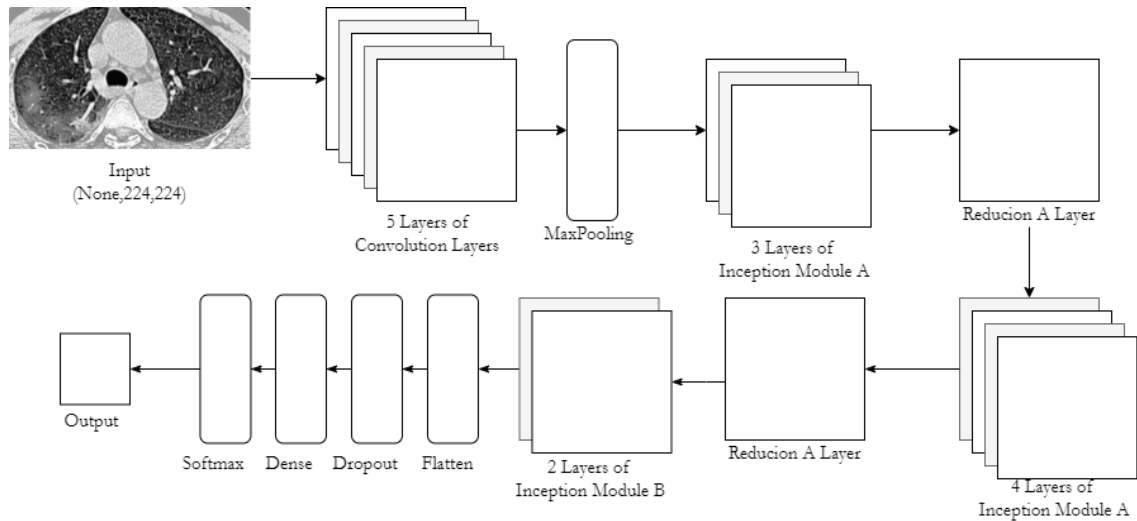


圖17、Inception Module[27]

實作上，我們採用的模型是google api[28]

的inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5模型，透過載入該模型其架構如圖18[29]。



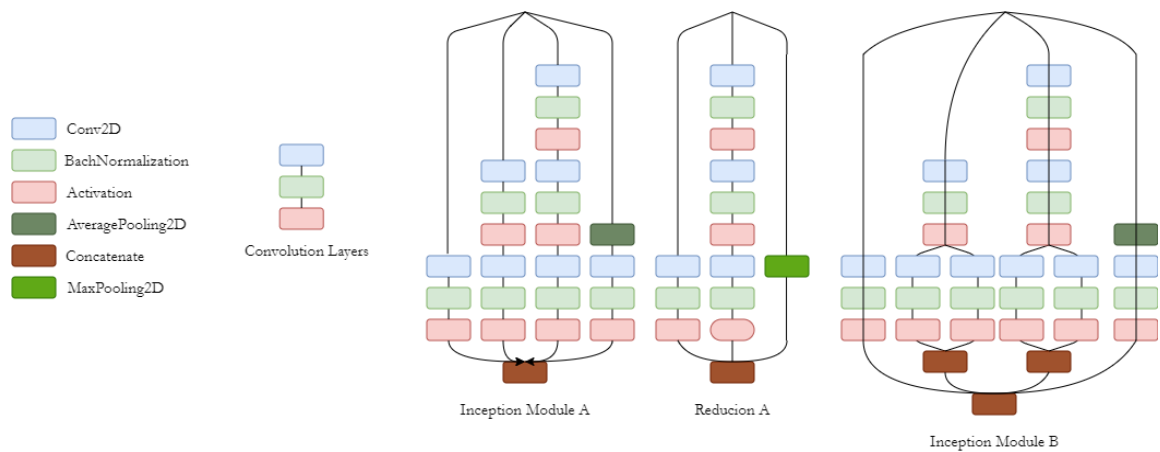


圖18、A high-level diagram of the model

如上圖，以卷積層為主體的網路，並有許多分支。這些分支是Inception的一大特色，可以看到在兩個Inception Module都有四個分支，在Inception Module B的分支當中還有分支。

5-2.VGG19

Vgg Network: Very Deep Convolutional Networks for Large-Scale Image Recognition,

VGG19包含了19個隱藏層（16個卷積層和3個全連接層），VGG網路的結構非常一致，從頭到尾全部使用的是3x3的捲積和2x2的max pooling。它的優點在於結構簡潔，但伴隨著它極深的特性，會造成大量參數，空間占用大的缺點。

圖19[30]VGG論文中的六種結構，根據不同的卷積核大小與神經網路層數作為區分，較為出名的為紅標的VGG16與黃標的VGG19，而他們的差異在於隱藏層的數量。

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

圖19、各種VGG的架構

實作上，我們採用的模型是github[31]上的https://github.com/fchollet/deep-learning-models/releases/download/v0.1/vgg19_weights_tf_dim_ordering_tf_kernels_notop.h5模型，架構如圖20。

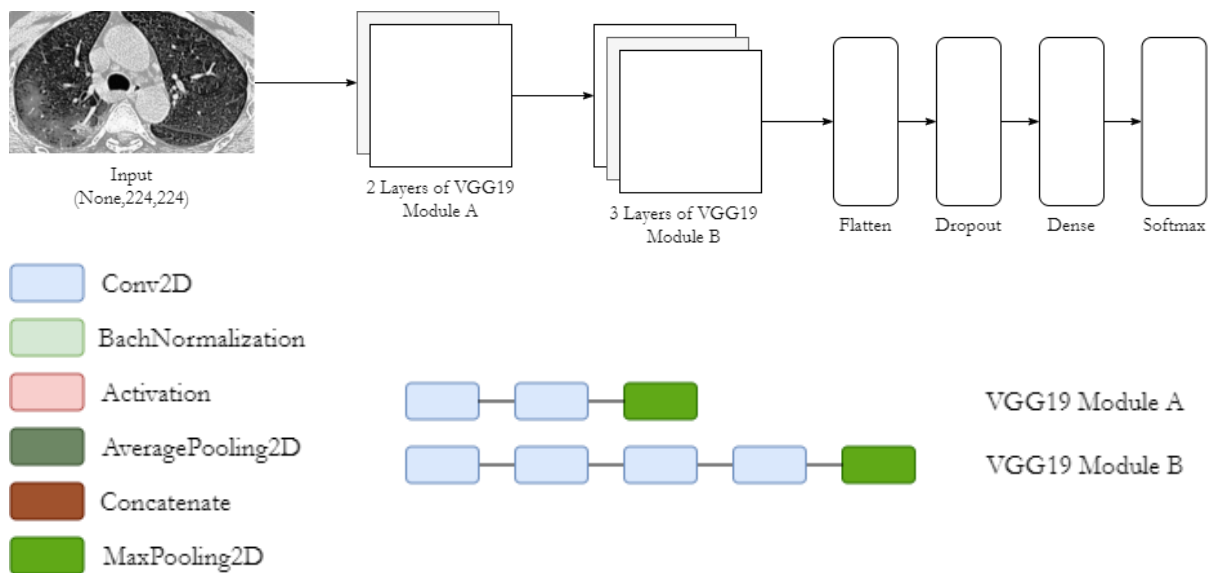


圖20、VGG19 模型架構

5-3.ResNet50

ResNet論文[32]出現的背景是流行淺層網路的時代，當時深一點的網路效果甚至更差。ResNet的存在簡單地使得深層網路更容易訓練，也開啟了各種超深網路的時代。

ResNet的基礎是一個建構塊，簡單地增加一條路線做單純的加法（圖21）組合的convolution layer。使深層網路訓練變得容易許多。而在疊更深的網路時，ResNet設計了瓶頸建構塊，降低3x3 convolution的寬度，大幅減少了所需的運算量。

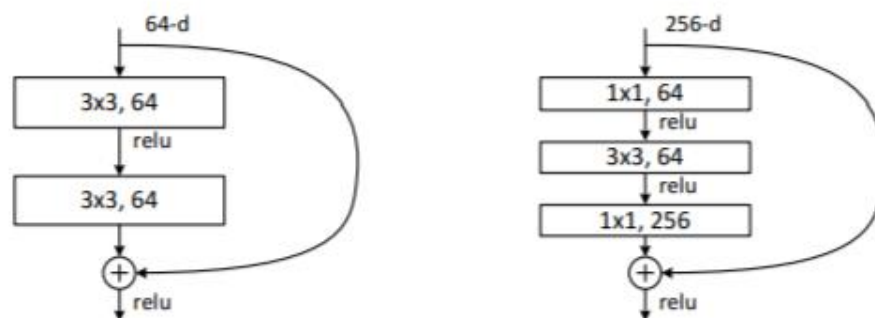


圖21、左邊是ResNet34的建構塊，右邊是ResNet-50/101/152的瓶頸建構塊

實作上，我們採用的模型是google api[33]上的https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5模型。

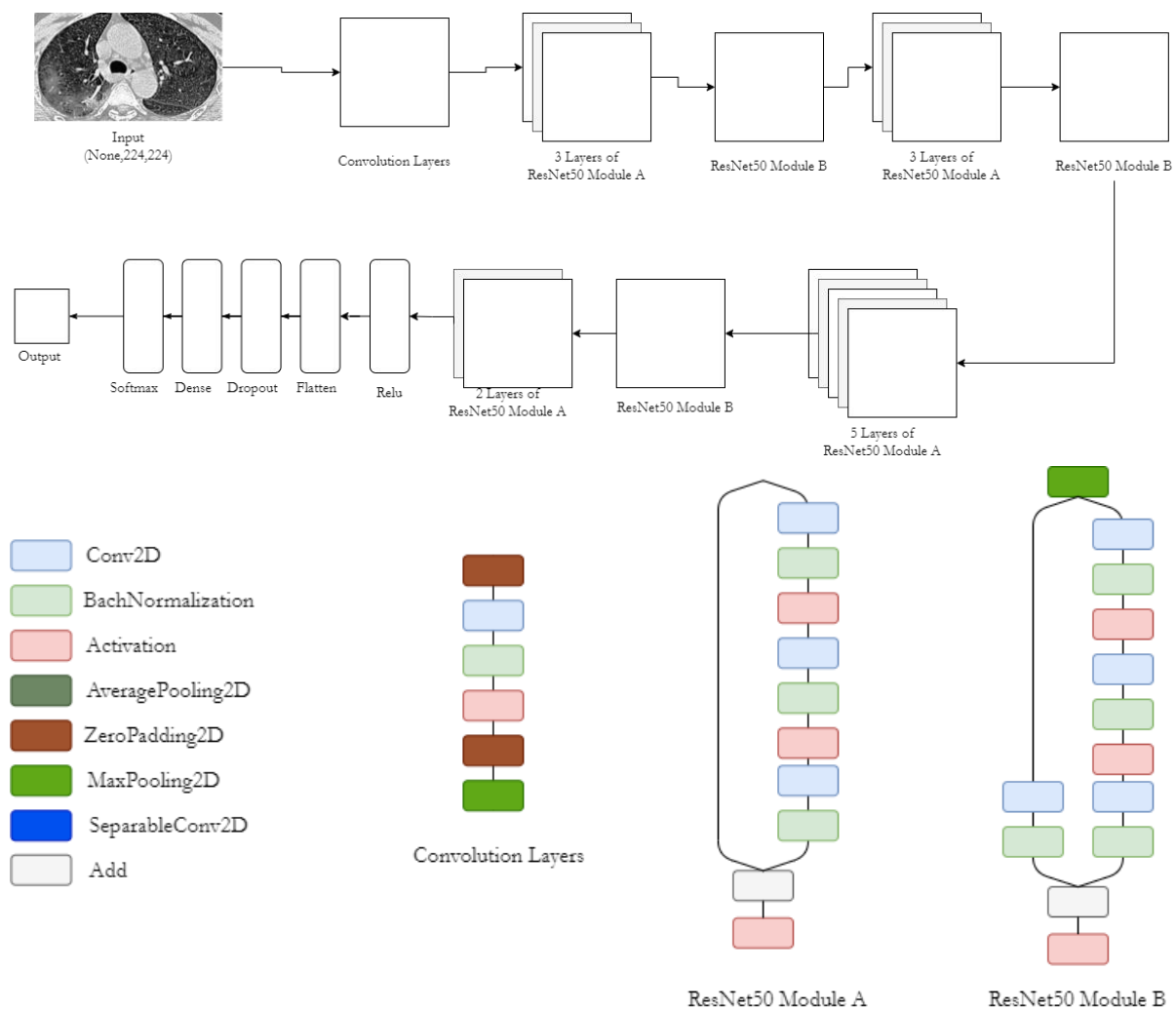


圖22、ResNet50 模型架構

5-4.Xception

Xception 取自 Extreme Inception，即表示 Xception 是一種極端的 Inception，於2016年提出[34]。

"Xception: Deep Learning with Depthwise Separable Convolutions"[34]有詳細描述，以 InceptionV3 為基礎去做改進，採用 depthwise separable convolution 來替代原本的 Inception module，並且加入了 ResNet 的 Residual Learning 方法。

下圖(圖23)可以看出Inception module跟Xception的差異，圖源來自"Xception: Deep Learning with Depthwise Separable Convolutions"，將每個維度都各自使用一個相應的 3x3 卷積層，把每個維度都各自分開來，這也就是Xception 的核心思想 — 將空間與通道分離。

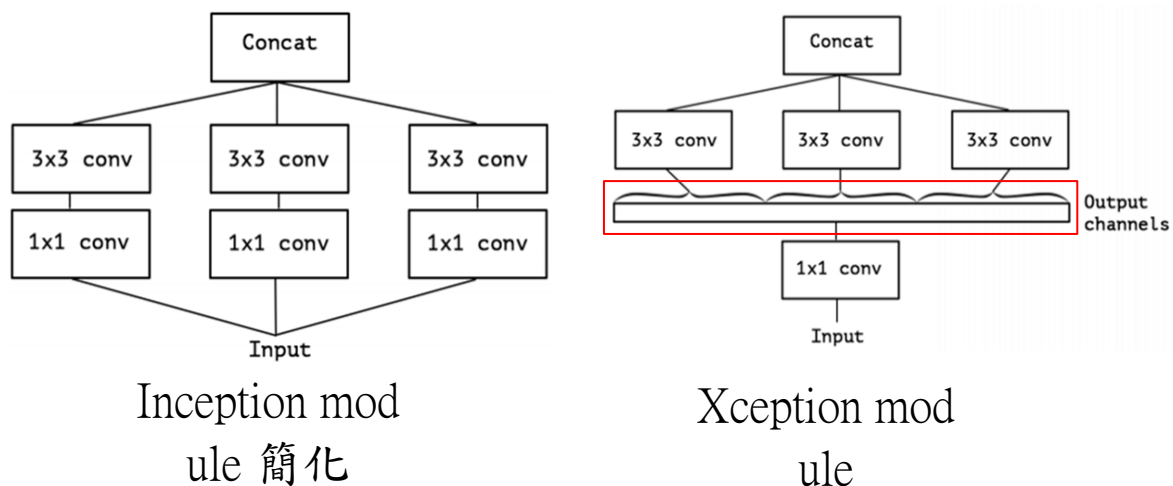


圖23、Inception module跟Xception的差異

Xception 的架構總共分為三個 flow — Entry/Middle/Exit，其中 Middle flow 會重複使用八次。

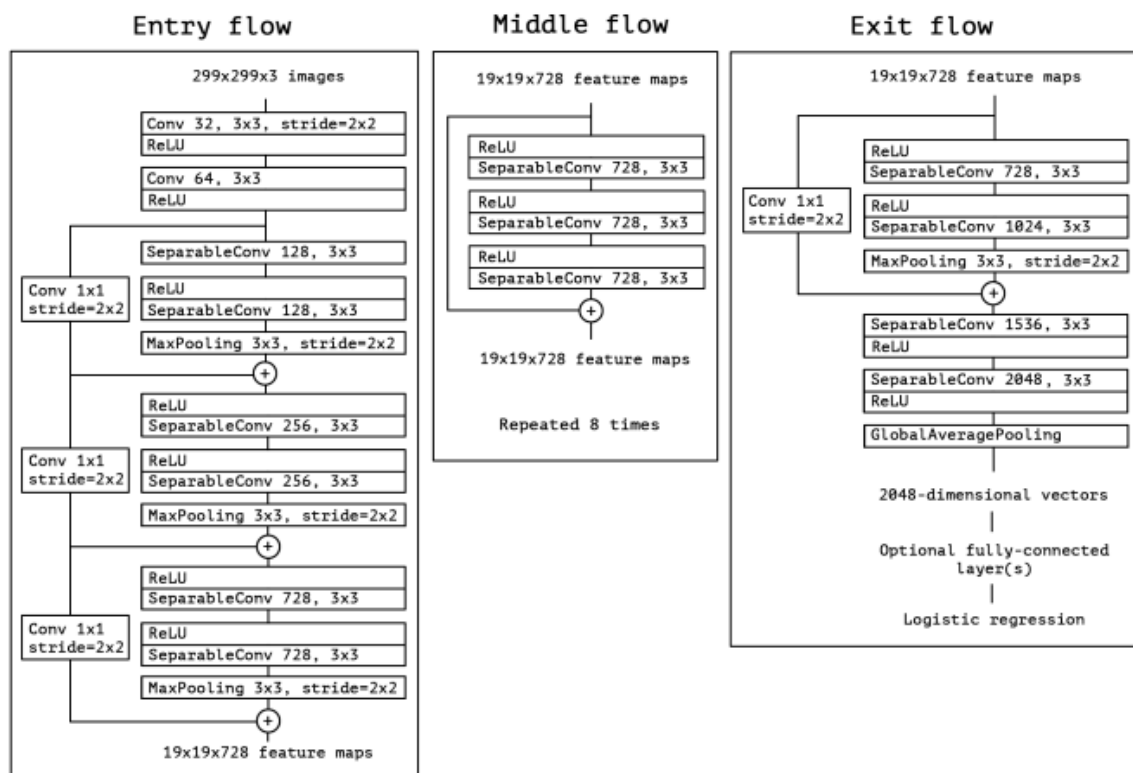


圖24、Xception 的整體架構

實作上，我們採用的模型是github[31]上的https://github.com/fchollet/deep-learning-models/releases/download/v0.4/xception_weights_tf_dim_ordering_tf_kernels_notop.h5模型。

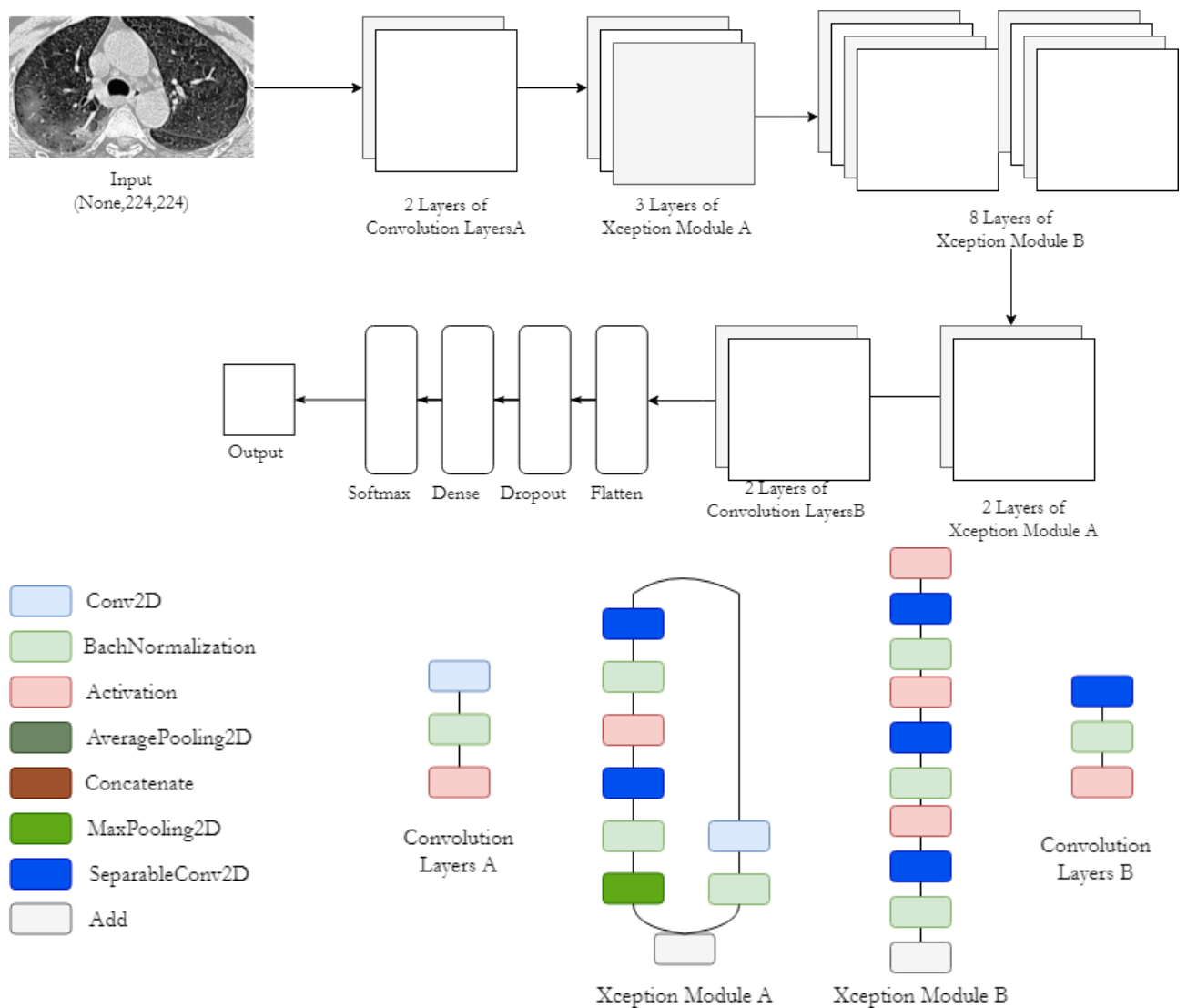


圖25、Xception 模型架構

5-5.DenseNet169

DenseNet的特點在於feature上，加強了feature的傳遞且更有效地利用了feature，另外，對於梯度消失問題，DenseNet提出了不一樣的方式去解決(圖26)。

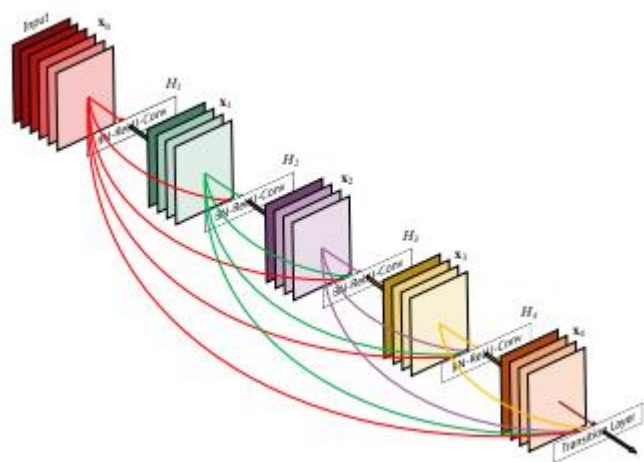


圖26、一個五層的dense block的結構圖[35]

根據"Densely Connected Convolutional Networks"中的說法，每一層都將所有前面的特徵圖作為輸入，這樣一來可以減輕梯度消失的問題。

關於DenseNet的架構，如圖x2為例，具有三個Dense Block的深度DenseNet。兩個相鄰塊之間的層稱為過渡層，通過卷積和池化改變特徵圖的大小。

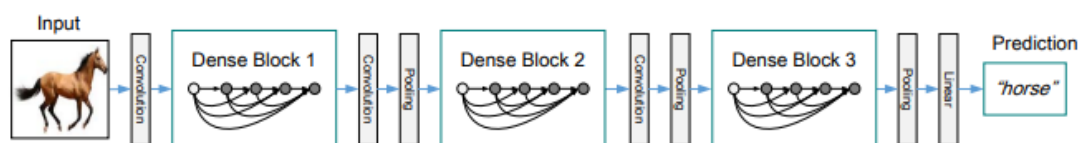


圖27、一個具有三個Dense Block的深度DenseNet[35]

實作上我們使用資料集裡面自帶的DenseNet程式進行，事實上令人意外的，它非常契合該資料集，結果會在第六章詳述。

六、結果

6-1. InceptionV3

6-1-1. Confusion Matrix

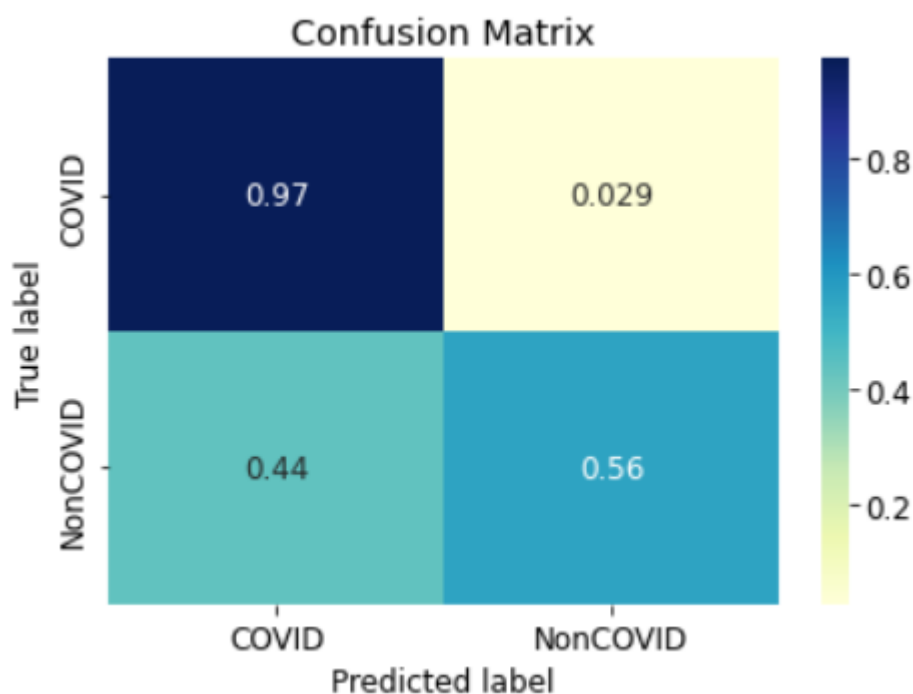


圖28、Confusion Matrix of InceptionV3

6-1-2. Classification Report

	precision	recall	f1-score	support
0	0.80	0.97	0.87	140
1	0.92	0.56	0.70	80
accuracy			0.82	220
macro avg	0.86	0.77	0.79	220
weighted avg	0.84	0.82	0.81	220

圖29、Classification Report of InceptionV3

6-2. VGG19

6-2-1. Confusion Matrix

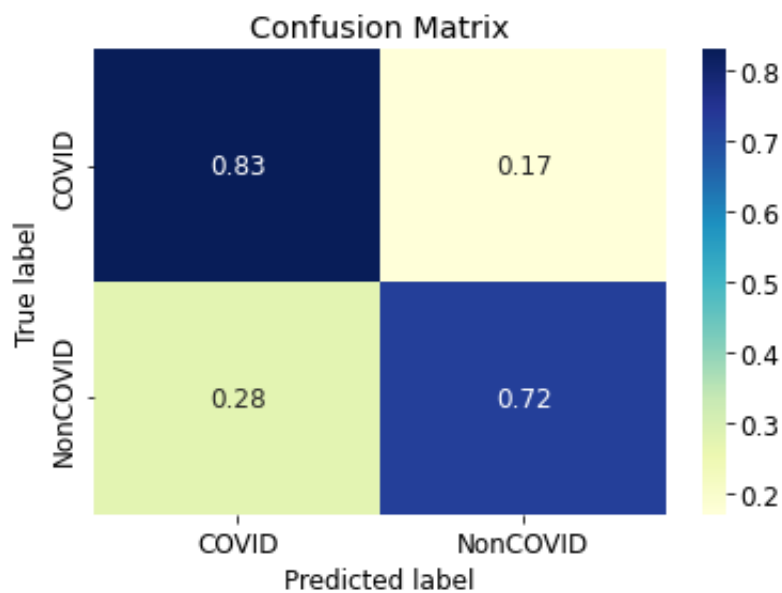


圖30、Confusion Matrix of VGG19

6-2-2. Classification Report

	precision	recall	f1-score	support
0	0.72	0.83	0.77	70
1	0.83	0.72	0.77	80
accuracy			0.77	150
macro avg	0.78	0.78	0.77	150
weighted avg	0.78	0.77	0.77	150

圖31、Classification Report of VGG19

6-3. ResNet50

6-3-1. Confusion Matrix

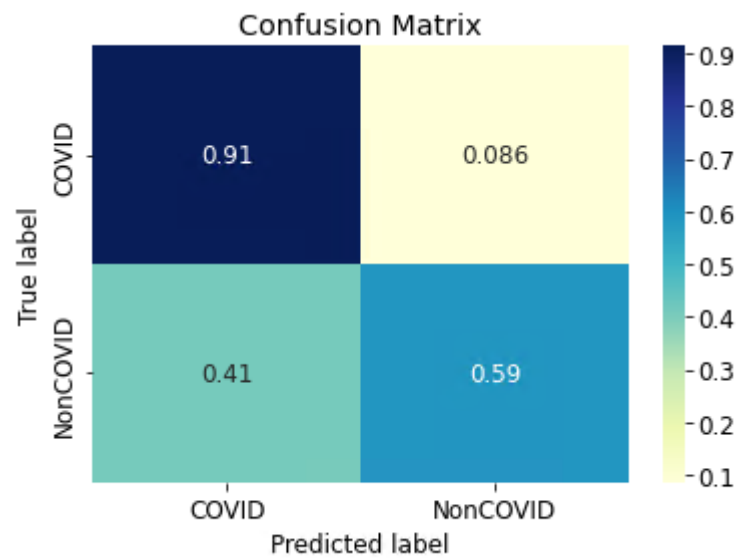


圖32、Confusion Matrix of ResNet50

6-3-2. Classification Report

	precision	recall	f1-score	support
0	0.79	0.91	0.85	139
1	0.80	0.59	0.68	80
accuracy			0.79	219
macro avg	0.80	0.75	0.76	219
weighted avg	0.79	0.79	0.79	219

圖33、Classification Report of ResNet50

6-4. Xception

6-4-1. Confusion Matrix

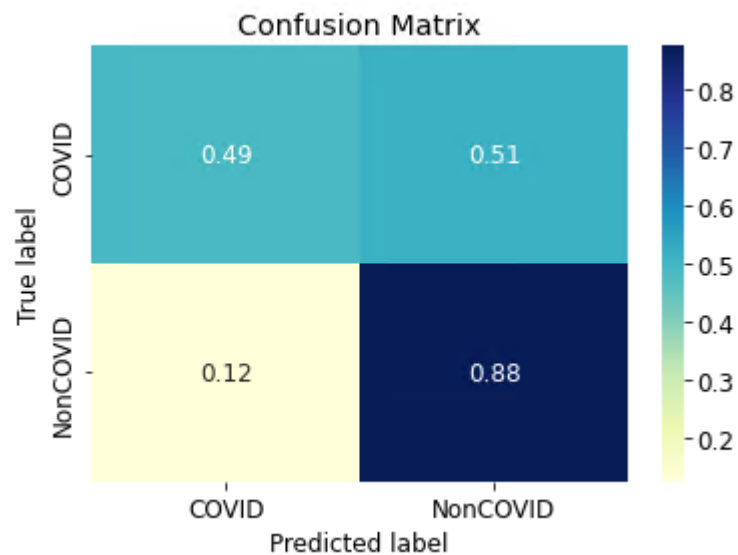


圖34、Confusion Matrix of Xception

6-4-2. Classification Report

```

              precision    recall  f1-score   support

     0       0.77       0.49       0.60        70
     1       0.66       0.88       0.75        80

 accuracy         0.69        150
 macro avg       0.72       0.68       0.67        150
 weighted avg    0.71       0.69       0.68        150

```

圖35、Classification Report of Xception

6-5. DenseNet169

```

average recall: 1.0000
average precision: 0.6744
F1: 0.8056
average accuracy: 0.7627
average AUC: 0.8818

```

圖36、Classification Report of DenseNet169

七、結論及未來展望

Method		precision	recall	f1-score
InceptionV3	macro avg	0.86	0.77	0.79
	weighted avg	0.84	0.82	0.81
VGG19	macro avg	0.78	0.78	0.77
	weighted avg	0.78	0.77	0.77
ResNet50	macro avg	0.80	0.75	0.76
	weighted avg	0.79	0.79	0.79
Xception	macro avg	0.72	0.68	0.67
	weighted avg	0.71	0.69	0.68
DenseNet169		0.6744	1	0.7627

表5、實驗結果比較

	Network	f1-score
--	---------	----------

Experimental results	InceptionV3	0.81
	VGG19	0.77
	ResNet50	0.79
	Xception	0.68
	DenseNet169	0.7627
Medical Image Processing and Deep Learning to Diagnose COVID-19 with CT Images	ANN	0.972
	SVM	0.985

表6、實驗結果與參考論文比較

見表5，由於資料平均，可以看到macro avg和weighted avg列資料誤差差不多在0.2以內，這邊以weight avg做分析。

根據” Medical Image Processing and Deep Learning to Diagnose COVID-19 with CT Images” 中的模型表現大於97%(見表6)，然而在這邊模型的表現卻沒有達到那個程度，這邊認為在特徵擷取的部分，我們沒有更好的將紋理編碼以及灰度共生矩陣的特性納入資料及做處理，這是我們認為可以改進的地方。

關於這個結果，我們猜測可以在前處理以及特徵擷取上進行調整，雖然目前還沒有清晰的原始碼，複合式的CNN模型未來也可以嘗試，對於課上應用到的方法我們學到很多，在過程中可調整的部分也是族多不及備載，未來我們將進行更多嘗試以改善結果。

參考資料

[1]Avaliable:<https://zh.wikipedia.org/wiki/2019%E5%86%A0%E7%8A%B6%E7%97%85%E6%AF%92%E7%97%85%E7%96%AB%E6%83%85>

[2]Avaliable:<https://sites.google.com/cdc.gov.tw/2019ncov/global>

[3]Ai T., Yang Z., Hou H., Zhan C., Chen C., Lv W., Tao Q., Sun Z., Xia L. Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: A report of 1014 cases. Radiology. 2020;296(2):E32 – E40. doi: 10.1148/radiol.2020200642.

[4]Irma Permata Sari,Widodo,Murien Nugraheni,Putra Wanda,2020,” A Basic Concept of Image Classification for Covid19 Patients Using Chest CT Scan and Convolutional Neural Network” ,2020 1st International Conference on Information Technology, Advanced Mechanical and Electrical Engineering (ICIT AMEE) DOI: 10.1109,pp. 175

[5]Avaliable:<https://www.mohw.gov.tw/fp-4626-56171-1.html>

[6]Felix Ritter, Tobias Boskamp, André Homeyer, Hendrik Laue, Michael Schwier, Florian Link, and Heinz-Otto Peitgen,” Medical Image Processing” ,60 IEEE PULSE,NOVEMBER/DECEMBER 2011

- [7]Xingyi Yang,Xuehai He,Jinyu Zhao,Yichen Zhang ,Shanghang Zhang,Pengtao Xie," COVID-CT-Dataset: A CT Image Dataset about COVID-19"
- [8]Abbas Mazrouei Sebdani,Amir Mostafavi" Medical Image Processing and Deep Learning to Diagnose COVID-19 with CT Images" ,
- [9] "Coronavirus disease 2019 deep learning models: Methodologic considerations"
- [10] "A Novel Medical Diagnosis model for COVID-19 infection detection based on Deep Features and Bayesian Optimization,"
- [11]Available:<https://github.com/UCSD-AI4H/COVID-CT>
- [12]Available:<https://zh.wikipedia.org/wiki/%E4%BA%8C%E5%80%BC%E5%8C%96>
- [13]Available:<https://zh.wikipedia.org/wiki/%E7%9B%B4%E6%96%B9%E5%9B%BE%E5%9D%87%E8%A1%A1%E5%8C%96>
- [14]Available:https://medium.com/@cindylin_1410/%E6%B7%BA%E8%AB%87-opencv-%E7%9B%B4%E6%96%B9%E5%9C%96%E5%9D%87%E8%A1%A1%E5%8C%96-ahe%E5%9D%87%E8%A1%A1-clahe%E5%9D%87%E8%A1%A1-ebc9c14a8f96
- [15]<https://commons.wikimedia.org/w/index.php?curid=668605>
- [16]Available:https://docs.opencv.org/3.4/dc/dd3/tutorial_gaussian_median_blur_bilateral_filter.html
- [17]Mathieu Aubry (INRIA, ENPC),Sylvain Paris (Adobe), Sam Hasinoff (Google),Jan Kautz (UCL), and Frédo Durand (MIT)," Fast Local Laplacian Filters:Theory and Applications"
- [18]"Local Laplacian Filters: Edge-aware Image Processing with a Laplacian Pyramid"
- [19]Available:https://github.com/KerouichaReda/Fast_LLF_python
- [20]Available:https://github.com/arsho/local_binary_patterns
- [21]Available:<https://chtseng.wordpress.com/2016/11/17/scikit-learn-%E5%B1%80%E9%83%A8%E4%BA%8C%E5%80%BC%E6%A8%A1%E5%BC%8F/>
- [22]Available:https://github.com/arsho/local_binary_patterns/blob/master/lbp.py
- [23]Available:https://www.researchgate.net/figure/Formulas-to-calculate-texture-features-based-on-GLCM_tbl2_342149372
- [24]Available:<https://github.com/kaushikjadhav01/COVID-19-Detection-Flask-App-based-on-Chest-X-rays-and-CT-Scans>
- [25]Naufal Hilmizen, Alhadi Bustamam, Devvi Sarwinda"The Multimodal Deep Learning for Diagnosing COVID-19 Pneumonia from Chest CTScan and X-Ray Images "
- [26]Swati Patil, Akshay. Golellu ,"Classification of COVID-19 CT Images using Transfer Learning Models"
- [27]Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens ," Rethinking the Inception Architecture for Computer Vision"
- [28]Available:https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
- [29]Available:<https://cloud.google.com/tpu/docs/inception-v3-advanced>
- [30]Karen Simonyan * ,Andrew Zisserman , " VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION"

- [31]Available:<https://github.com/fchollet/deep-learning-models/>
- [32]Kaiming He, Xiangyu, Zhang ,Shaoqing Ren, Jian Sun,” Deep Residual Learning for Image Recognition”
- [33]Available:https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
- [34]Franc ,ois Chollet,” Xception: Deep Learning with Depthwise Separable Convolutions”
- [35]Gao Huang*, Zhuang Liu*, Laurens van der Maaten ,Kilian Q. Weinberger, “Densely Connected Convolutional Networks”