

利用卷積神經網路實作花卉辨識器

— 使用 TensorFlow 與 PyTorch

巨資三 A 07170184 陳亞萱

巨資一 B 09170244 梁育誠

社會三 A 07115101 陳朱鋆

目錄

壹、 背景	3
貳、 動機與目的	3
參、 專有名詞解釋	4,5
肆、 方法	5
伍、 結果	6-28
陸、 組員心得	29
柒、 參考文獻	30

壹、 背景

全連接神經網路(Fully-connect Neural Network, FNN)，是一種多個神經元的連接模式，其特色為上一層的神經元與下一層所有神經元相接。在前向傳播的過程，經過上一層神經元對應的權值進行加權和運算，再通過激活函數，如 ReLu 或 Sigmoid 函數，得到本層神經元的輸出，過程即為每一個神經元接收上一層的輸入進行簡單邏輯回歸操作，至全部神經元從輸入層，到輸出層依次進行邏輯回歸為止；在反向傳播的過程，利用前向傳播所得的預測值，與真實值的差別來計算損失，根據損失函數來反方向計算每一層的梯度，自最後一層向前改變每一層的權重，即更新參數，其核心概念為損失對每一層的每一個參數求梯度的鏈式求導法則。全連接神經網路通過不斷的前向與反向傳播，調整神經網路的權重，至預設的疊代次數或較佳程度的樣本學習時，即完成訓練，雖然全連接神經網路在處理單色、小尺寸的圖片上可以達到好的辨識效果，但是遇到色彩複雜、大尺寸的圖片時，會因為增加特徵值，權重參數量遽增，若是訓練資料不足，會產生擬合過度的問題，無法良好地調適其他資料或預測未來的觀察結果，此時，卷積神經網路(Convolutional Neural Networks, CNN)可以解決上述所產生的問題，其基本架構分為，卷積層與池化層負責特徵萃取、密集層負責分類，在卷積層的部分，會利用 Kernel 函數得到降階的特徵圖，其中，會設定垂直與水平偵測過濾器來擷取圖片特徵，色彩方面，由於彩色圖片由 RGB 三原色組成，因此需要建立三個通道的過濾器來計算卷積；此外，為了防止輸出圖片被萃取得過小張，利用 Padding 的做法，將圖片周圍補上零，加大輸出圖片的尺寸，同時也強化邊緣特徵。由此可知，卷積神經網路適合在處理色彩複雜與大尺寸圖片時使用。

貳、 動機與目的

花卉在日常的應用十分廣泛，諸如食品、藥用、染料，經過 NASA 科學家發現，某些種類也能有效清除空氣中的毒素、致癌物與細菌，例如，向日葵有助於將土壤中的鉛濃度降低 43%，在日本的廣島與福島皆利用其吸取土壤中的輻射；洋甘菊可以有效治療扭傷、關節炎；金盞花則是一種抗炎藥，用於治療輕微傷口、疼痛與腫脹。由此可見，花卉對於人類健康與環境保護皆扮演重要的角色，因此，對其辨識與分類能使健康保健與保育環境的品質得到保障，有鑒於花卉種類繁多，或是相同品種卻有不同顏色與外觀，有些則是不同品種卻外觀相似，透過人工進行分類不僅需要專家來執行，而且當資料量龐大時，處理上會相當耗時，因此，本組欲以卷積神經網路來應用於花卉的分類。

卷積神經網路是模仿人類大腦的認知方式，例如人類在辨識圖像時，會藉由彩色的點、線、面將其構成不同的形狀。其至今應用於多個面向，包含影像辨識、視訊分析、自然語言處理、標靶藥物發現，以及著名的 AlphaGo 對戰南韓圍棋的棋手，讓研究者對深度學習有更廣泛的認知。在開發上，也有專為處理深度學習的框架，例如 TensorFlow、Torch、Caffe、Keras、MxNet、CNTK 與 Deeplearning4j 等。本組欲透過卷積神經網路，以及深度學習的框架，以期能夠對不同角度、大小、光影之下的花卉圖像，利用特徵萃取，將花卉有效分類。

參、專有名詞解釋

一、機器學習

機器學習是讓電腦或機器能對已知的資料進行分析、計算、建立模型並對未知的資料進行預測。依照訓練資料、產出判別與過程和結果的不同，機器學習可以分為監督式學習(Supervised Learning)、非監督式學習(Unsupervised Learning)與增強式學習(Reinforcement Learning)。

監督式學習需要倚靠大量的人工前置作業，將所有可能的特徵標記，電腦將從標籤化的資訊中分析模式，透過修正對比誤差達到精準的預測，雖然使得監督式學習的準確率高，但是當資訊量增加，將難以對所有資料進行標記，面對未知的領域時幾乎完全無法運作；以 Netflix 的精準個性化推薦系統為例，透過用戶的觀看歷史紀錄得知其的喜好類型，再利用推薦系統呈現給用戶不同類型的劇集海報，或是記錄用戶經常看的演員，就以該演員的劇照作為推薦海報(36r, 2020)。

非監督式學習適用於資料探勘初期，其依照關聯性去歸類，找出潛在規則，再形成集群，不會對資訊有是非的判別，也正因為如此，可能造成不具重要性的特徵被過度放大，導致結果偏誤或無意義的分群問題；以鴻海的「FOXCONN NxVAE」為例，在良率高的成熟產品線，瑕疵樣本取得困難，因此運用正面表列的模型訓練方式，以易取得的正樣本進行光學檢測運算，得以增加模型的容錯能力，藉此降低 50% 以上的產線檢測人力(王郁倫，2021)。

增強式學習則不須給機器任何資料，機器透過環境的正向與負向回饋，直接從互動的刺激去學習與回饋；以 Wayve.ai 的自駕車為例，自駕車需要考慮可行區域、不同限速與躲避障礙等問題，有些自駕車與強化學習相結合，如軌跡優化、動態路徑、最優控制以及高速路中的情景學習等，而 Wayve.ai 透過四個卷積層與三個全連接層的深層神經網路，利用深度強化學習算法來訓練自駕車於白天駕駛(Derrick Mwit, 2021)。

二、深度學習

深度學習發展自類神經網路，使用多節點且分層的方式運算，擁有高效能的 GPU 提高數值與矩陣間運算的速度，與各式感測器所提供大量的訓練資料，得以發展深度學習。其具有多層隱藏層，前一層的輸出被當作更深層的輸入，透過隱藏層的模型建構來學習更複雜且抽象的特徵，藉以提高預測的準確率。因此，深度學習成為只要將資料輸入類神經網路，就可以自行抽出特徵的特徵學習(feature learning)。

深度學習能辨識圖像資料與波形資料這類無法符號化的資料，如今已廣泛應用於語音辨識、影像辨識與自然語言處理等，例如 Face2Face 的應用，利用人臉捕捉，在視頻裡即可將表情轉移至影像人物上，同樣的原理也應用於 3D 場景重建與特效；Pix2Pix 則是可以向電腦輸入一幅簡筆畫或一些色塊，即可讓電腦輸出新的圖片，將黑白的圖像轉為彩色、將白天的場景變成夜晚等；此外，不僅能夠用深度學習辨識字體之間的差異，還能讓電腦具備利用多種風格字體來書寫的能力(量子位，2017)。

三、卷積神經網路

卷積神經網路具有從原始資料中自動學習出特徵的能力，相較於淺層模型來說，能夠自我學習到足以表達樣本空間的特徵，稱之為「深度特徵」，透過加深層數，層數愈多的運算能學習到更複雜且抽象的特徵。並且運用局部連接神經元與權值共享，能夠減少大量的網路參數與運算，圖像具有平移不變性，因此在同一張圖像裡，可以利用相同的過濾器來提取特徵，達到權值共享，多個過濾器能提取多個特徵，得以將圖像內容進行完整訓練。此外，在卷積層的部分，不同於全連接神經網路，因為圖像在空間上，局部的像素關聯性較強，因此透過局部感知域(local receptive fields)進行圖像感知，在下一層將這些局部感知的訊息整合，即可得到全區域的訊息(李兆健，2017)。

肆、 方法

本組運用卷積神經網路(convolutional neural networks, CNN)的深度學習框架，TensorFlow 與 Torch 進行花卉影像之自動辨識，花卉的種類為五種，包含雛菊、蒲公英、玫瑰、向日葵與鬱金香，資料集中的影像為花卉不同的大小、顏色、角度與光影變化等。本組首先對不同花卉的影像進行偵測與定位，然後針對不同花卉特徵進行分類，再將各類的特徵進行機器訓練與測試，將分類出來花卉進行特徵比對，若比對的結果不符合群體特徵的花卉，則再改變此花卉的參數，最後利用訓練完整的模型，預測符合該特徵之花卉的種類。

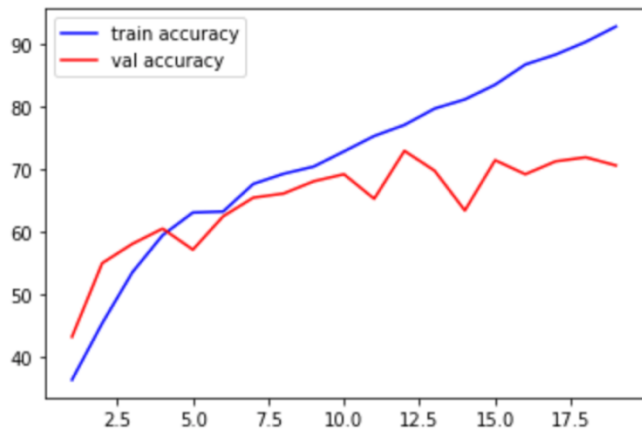
伍、 結果

一、PyTorch 花卉辨識器

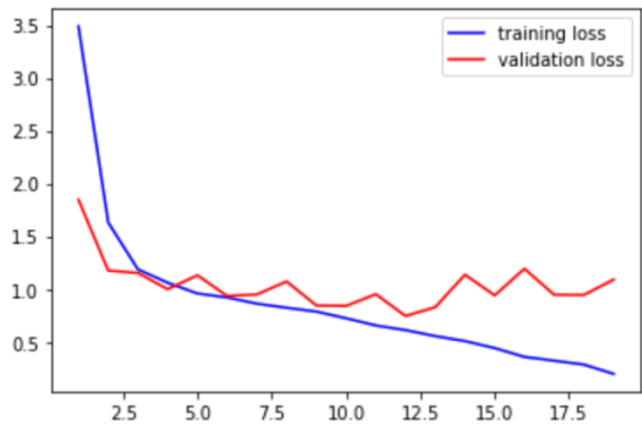
1. 圖表一顯示彩色圖像在 learning rate 為 0.001 時，訓練的準確率為 71.82%，測試的準確率為 65.28%。

▼圖表一、Learning Rate = 0.001 (RGB)

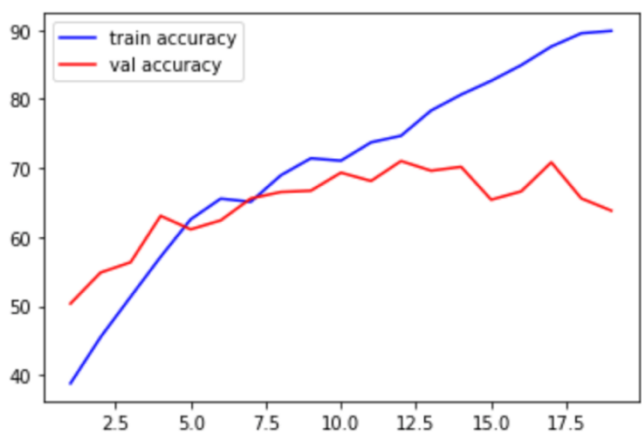
		第一次		第二次		第三次	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1/19	0.3631	0.4318	0.3881	0.5037	0.3462	0.4692
Epoch	2/19	0.4535	0.5495	0.4550	0.5486	0.4712	0.5411
Epoch	3/19	0.5350	0.5804	0.5135	0.5636	0.5242	0.5636
Epoch	4/19	0.5942	0.6047	0.5715	0.6308	0.5792	0.6514
Epoch	5/19	0.6304	0.5710	0.6258	0.6112	0.6319	0.6187
Epoch	6/19	0.6319	0.6243	0.6558	0.6243	0.6438	0.5720
Epoch	7/19	0.6762	0.6542	0.6512	0.6561	0.6727	0.6944
Epoch	8/19	0.6923	0.6607	0.6896	0.6654	0.6835	0.6850
Epoch	9/19	0.7038	0.6804	0.7142	0.6673	0.7288	0.6486
Epoch	10/19	0.7281	0.6916	0.7108	0.6935	0.7338	0.6888
Epoch	11/19	0.7527	0.6523	0.7373	0.6813	0.7685	0.6299
Epoch	12/19	0.7704	0.7290	0.7469	0.7103	0.7942	0.6757
Epoch	13/19	0.7965	0.6972	0.7835	0.6963	0.7927	0.7028
Epoch	14/19	0.8112	0.6336	0.8065	0.7019	0.8392	0.6692
Epoch	15/19	0.8346	0.7140	0.8265	0.6542	0.8462	0.7215
Epoch	16/19	0.8669	0.6916	0.8492	0.6664	0.8704	0.7047
Epoch	17/19	0.8827	0.7121	0.8762	0.7084	0.8962	0.7299
Epoch	18/19	0.9031	0.7187	0.8954	0.6561	0.9065	0.7308
Epoch	19/19	0.9277	0.7056	0.8988	0.6383	0.9173	0.7065
AVG.		0.7134	0.6475	0.7050	0.6461	0.7182	0.6528



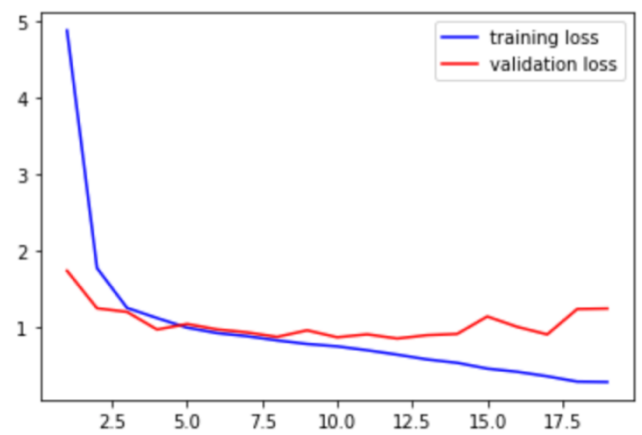
▲圖表一之一、第一次訓練與測試精準度
(輸入層 120x120)



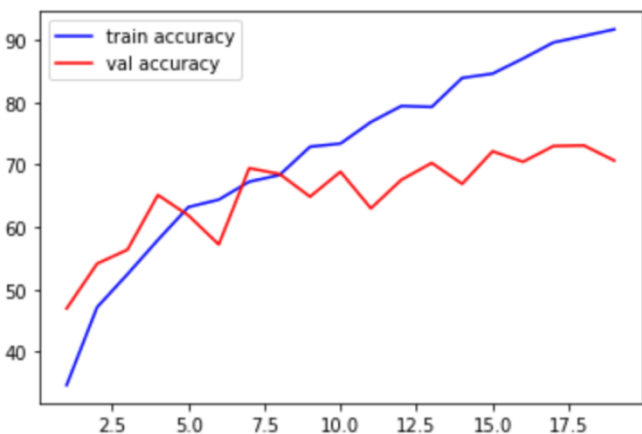
▲圖表一之二、第一次訓練與測試損耗
(輸入層 120x120)



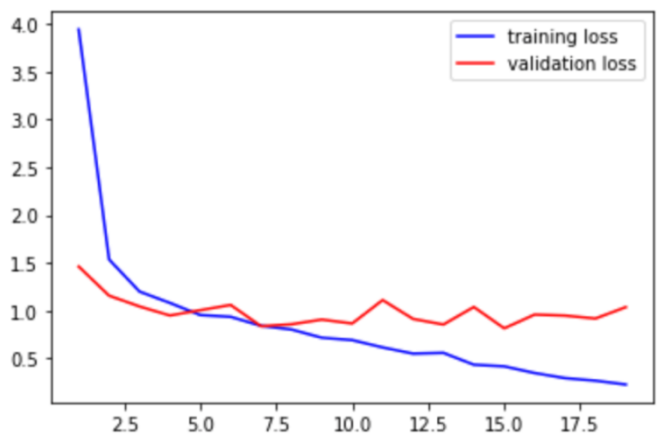
▲圖表一之三、第二次訓練與測試精準度
(輸入層 120x120)



▲圖表一之四、第二次訓練與測試損耗
(輸入層 120x120)



▲圖表一之五、第三次訓練與測試精準度
(輸入層 120x120)

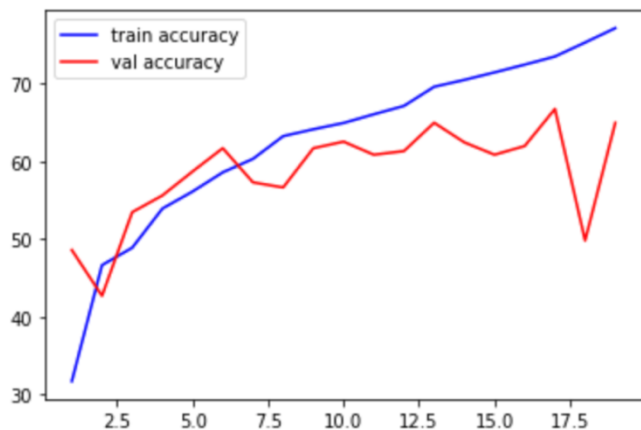


▲圖表一之六、第三次訓練與測試損耗
(輸入層 120x120)

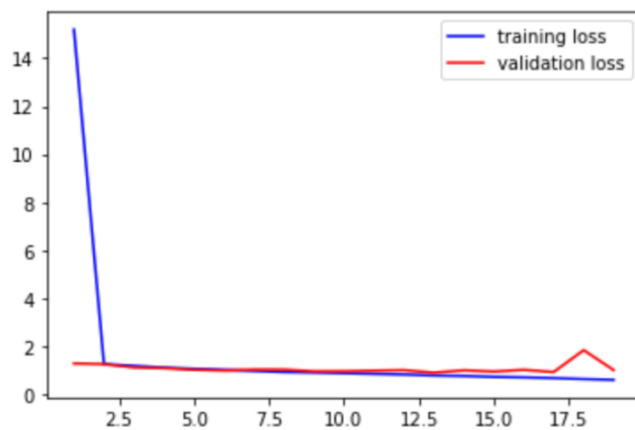
2. 圖表二顯示彩色圖像在 learning rate 為 0.01 時，訓練的準確率為 61.92%，測試的準確率為 57.37%，圖中顯示訓練與測試三次下來，準確率並沒有提升，並且較 learning rate 為 0.001 時低。

▼圖表二、Learning Rate = 0.01 (RGB)

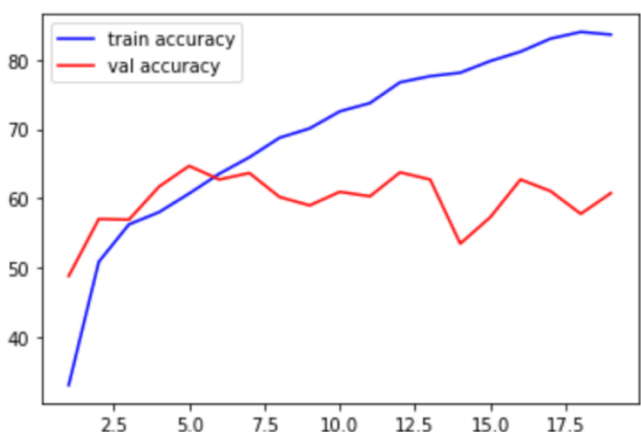
		第一次		第二次		第三次	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1/19	0.3173	0.4860	0.3304	0.4879	0.2969	0.4598
Epoch	2/19	0.4665	0.4271	0.5085	0.5701	0.4573	0.4963
Epoch	3/19	0.4888	0.5346	0.5623	0.5692	0.5073	0.5159
Epoch	4/19	0.5392	0.5561	0.5800	0.6168	0.5350	0.5318
Epoch	5/19	0.5612	0.5869	0.6069	0.6467	0.5550	0.5654
Epoch	6/19	0.5858	0.6168	0.6354	0.6271	0.5773	0.5542
Epoch	7/19	0.6031	0.5729	0.6592	0.6364	0.5835	0.5364
Epoch	8/19	0.6323	0.5664	0.6873	0.6019	0.6019	0.6019
Epoch	9/19	0.6412	0.6168	0.7008	0.5897	0.6092	0.5822
Epoch	10/19	0.6492	0.6252	0.7254	0.6093	0.6219	0.6318
Epoch	11/19	0.6604	0.6084	0.7373	0.6028	0.6454	0.5822
Epoch	12/19	0.6712	0.6131	0.7673	0.6374	0.6612	0.5944
Epoch	13/19	0.6958	0.6495	0.7762	0.6271	0.6769	0.4664
Epoch	14/19	0.7046	0.6243	0.7812	0.5346	0.6965	0.6514
Epoch	15/19	0.7142	0.6084	0.7981	0.5729	0.7142	0.5981
Epoch	16/19	0.7242	0.6196	0.8115	0.6271	0.7296	0.6299
Epoch	17/19	0.7346	0.6673	0.8304	0.6103	0.7554	0.6280
Epoch	18/19	0.7527	0.4981	0.8400	0.5776	0.7665	0.6430
Epoch	19/19	0.7712	0.6495	0.8362	0.6075	0.7742	0.6196
AVG.		0.6270	0.5856	0.6934	0.5975	0.6192	0.5731



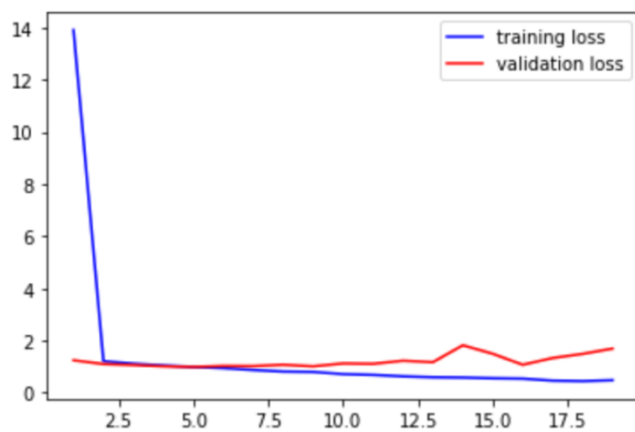
▲圖表二之一、第一次訓練與測試精準度
(輸入層 120x120)



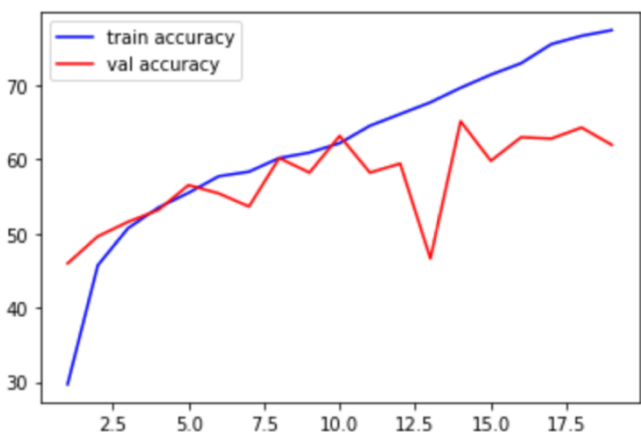
▲圖表二之二、第一次訓練與測試損耗
(輸入層 120x120)



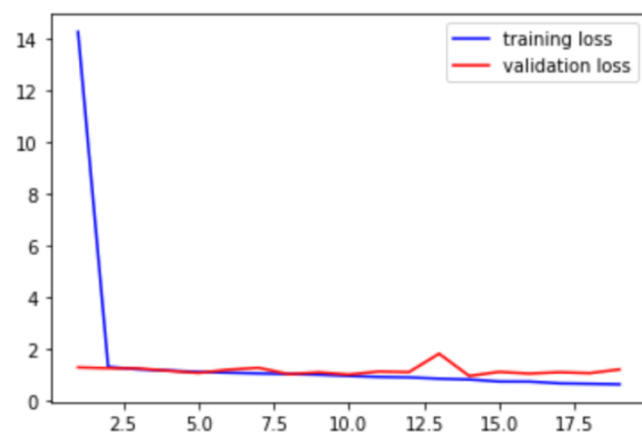
▲圖表二之三、第二次訓練與測試精準度
(輸入層 120x120)



▲圖表二之四、第二次訓練與測試損耗
(輸入層 120x120)



▲圖表二之五、第三次訓練與測試精準度
(輸入層 120x120)

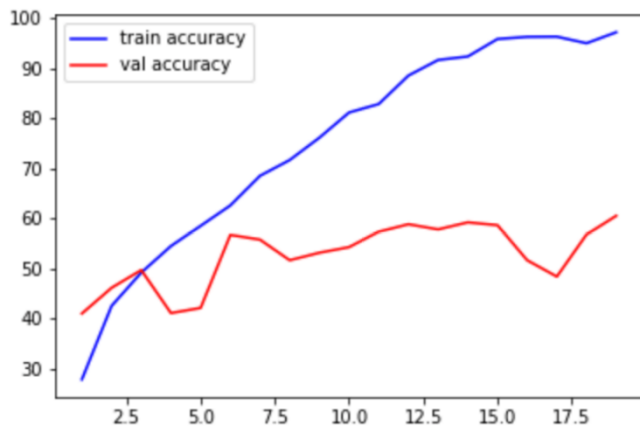


▲圖表二之六、第三次訓練與測試損耗
(輸入層 120x120)

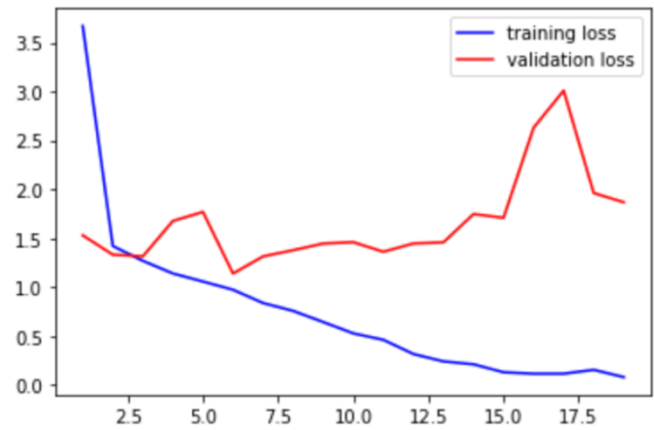
3. 圖表三顯示灰階圖像在 learning rate 為 0.001 時，訓練的準確率為 69.45%，測試的準確率為 52.05%，與彩色圖像在 learning rate 為 0.001 時相比，訓練與測試的準確率並未提升。

▼圖表三、Learning Rate = 0.001 (Grey)

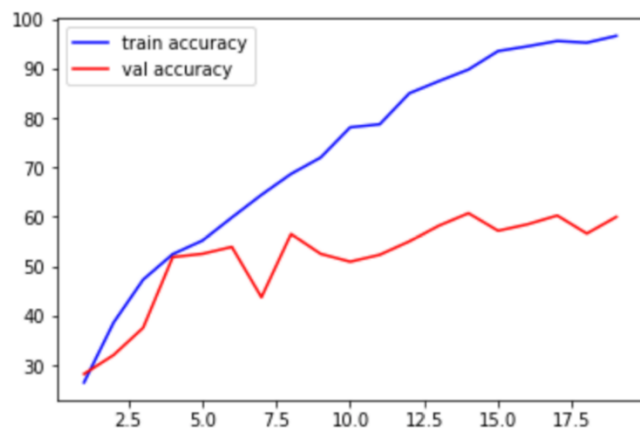
		第一次		第二次		第三次	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1/19	0.2777	0.4093	0.2650	0.2822	0.2688	0.3636
Epoch	2/19	0.4246	0.4607	0.3862	0.3206	0.3612	0.3935
Epoch	3/19	0.4912	0.4963	0.4731	0.3757	0.4150	0.4402
Epoch	4/19	0.5442	0.4103	0.5246	0.5187	0.4550	0.4579
Epoch	5/19	0.5846	0.4206	0.5519	0.5252	0.5138	0.4636
Epoch	6/19	0.6254	0.5664	0.5988	0.5393	0.5619	0.3449
Epoch	7/19	0.6846	0.5570	0.6446	0.4374	0.6035	0.5037
Epoch	8/19	0.7162	0.5159	0.6869	0.5654	0.6350	0.5495
Epoch	9/19	0.7608	0.5308	0.7200	0.5252	0.6935	0.5692
Epoch	10/19	0.8112	0.5421	0.7812	0.5093	0.7227	0.5523
Epoch	11/19	0.8281	0.5729	0.7873	0.5234	0.7631	0.5673
Epoch	12/19	0.8850	0.5879	0.8500	0.5505	0.8050	0.5804
Epoch	13/19	0.9162	0.5776	0.8746	0.5822	0.8492	0.5860
Epoch	14/19	0.9231	0.5916	0.8981	0.6075	0.8888	0.5645
Epoch	15/19	0.9581	0.5860	0.9354	0.5720	0.9054	0.5916
Epoch	16/19	0.9623	0.5159	0.9450	0.5850	0.9058	0.5785
Epoch	17/19	0.9627	0.4832	0.9558	0.6028	0.9365	0.5794
Epoch	18/19	0.9496	0.5673	0.9523	0.5664	0.9450	0.5888
Epoch	19/19	0.9715	0.6047	0.9662	0.6000	0.9665	0.6140
AVG.		0.7514	0.5261	0.7261	0.5152	0.6945	0.5205



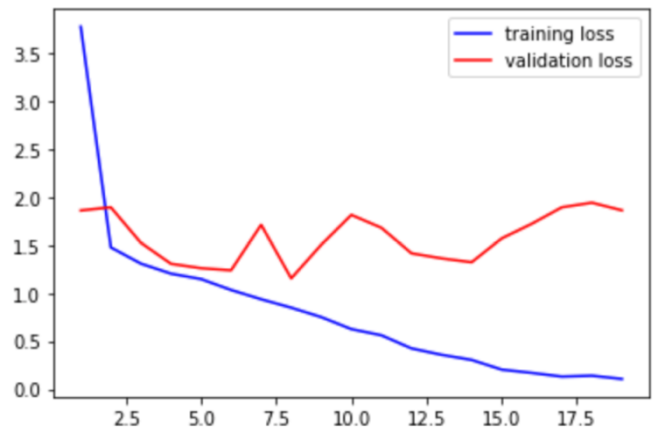
▲圖表三之一、第一次訓練與測試精準度
(輸入層 120x120)



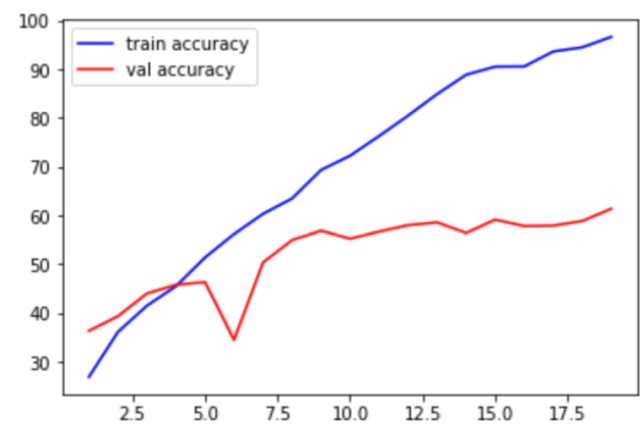
▲圖表三之二、第一次訓練與測試損耗
(輸入層 120x120)



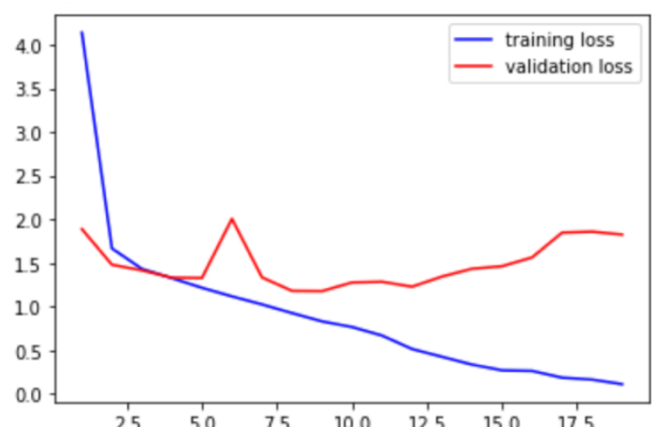
▲圖表三之三、第二次訓練與測試精準度
(輸入層 120x120)



▲圖表三之四、第二次訓練與測試損耗
(輸入層 120x120)



▲圖表三之五、第三次訓練與測試精準度
(輸入層 120x120)

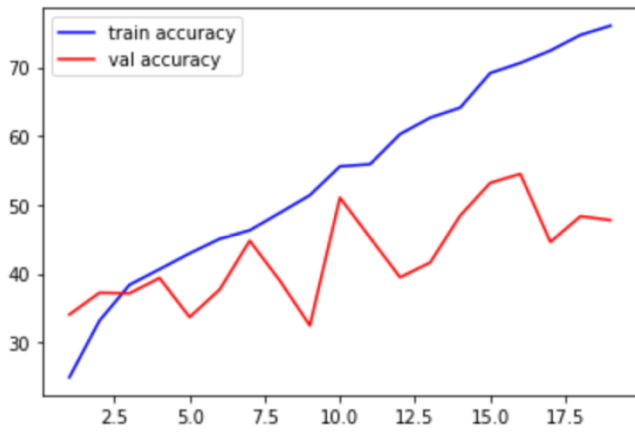


▲圖表三之六、第三次訓練與測試損耗
(輸入層 120x120)

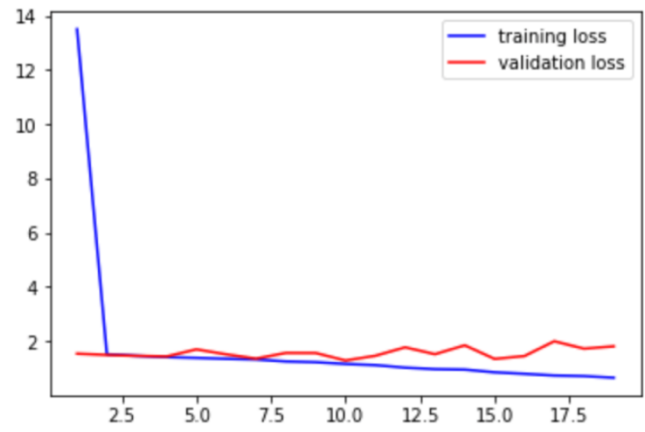
4. 圖表四顯示灰階圖像在 learning rate 為 0.01 時，訓練的準確率為 59.43%，測試的準確率為 39.58%，圖中顯示訓練與測試三次下來，準確率並沒有提升，並且與彩色圖像在 learning rate 為 0.01 時相比，訓練與測試的準確率並未較高。

▼圖表四、Learning Rate = 0.01 (Grey)

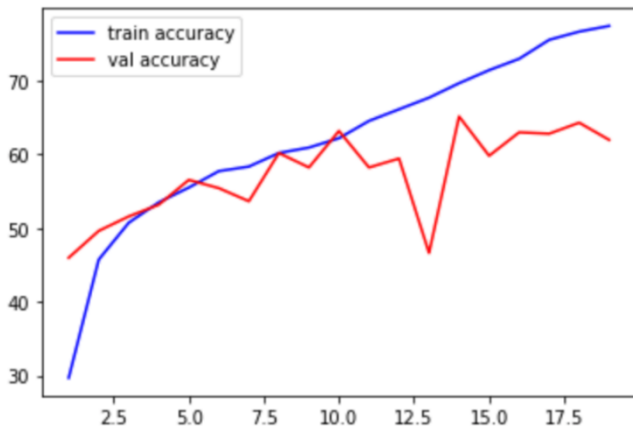
		第一次		第二次		第三次	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1/19	0.2488	0.3402	0.2969	0.4598	0.2788	0.3486
Epoch	2/19	0.3312	0.3720	0.4573	0.4963	0.3658	0.3150
Epoch	3/19	0.3835	0.3710	0.5073	0.5159	0.4015	0.3346
Epoch	4/19	0.4062	0.3935	0.5350	0.5318	0.4288	0.3383
Epoch	5/19	0.4292	0.3364	0.5550	0.5654	0.4577	0.4065
Epoch	6/19	0.4504	0.3766	0.5773	0.5542	0.4888	0.4318
Epoch	7/19	0.4627	0.4477	0.5835	0.5364	0.5065	0.2785
Epoch	8/19	0.4881	0.3897	0.6019	0.6019	0.5242	0.3794
Epoch	9/19	0.5138	0.3243	0.6092	0.5822	0.5538	0.3000
Epoch	10/19	0.5558	0.5103	0.6219	0.6318	0.6123	0.4467
Epoch	11/19	0.5558	0.4523	0.6454	0.5822	0.6112	0.4785
Epoch	12/19	0.6027	0.3944	0.6612	0.5944	0.6850	0.4626
Epoch	13/19	0.6265	0.4159	0.6769	0.4664	0.7000	0.3710
Epoch	14/19	0.6412	0.4841	0.6965	0.6514	0.7185	0.4579
Epoch	15/19	0.6915	0.5318	0.7142	0.5981	0.7404	0.5084
Epoch	16/19	0.7062	0.5449	0.7296	0.6299	0.7650	0.4187
Epoch	17/19	0.7242	0.4458	0.7554	0.6280	0.8012	0.2495
Epoch	18/19	0.7469	0.4832	0.7665	0.6430	0.8112	0.4850
Epoch	19/19	0.7064	0.4776	0.7742	0.6196	0.8412	0.5093
AVG.		0.5406	0.4259	0.6192	0.5731	0.5943	0.3958



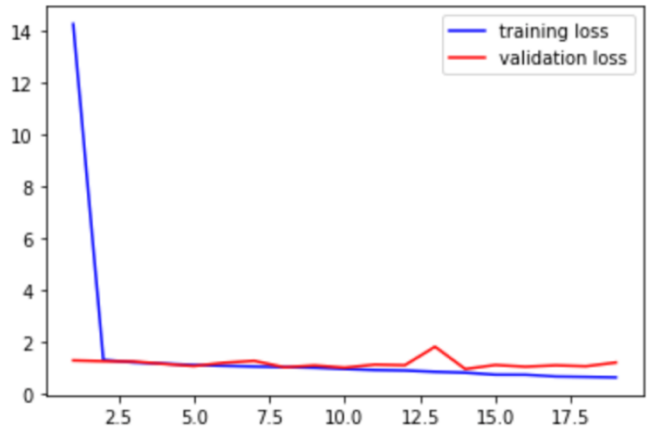
▲圖表四之一、第一次訓練與測試精準度
(輸入層 120x120)



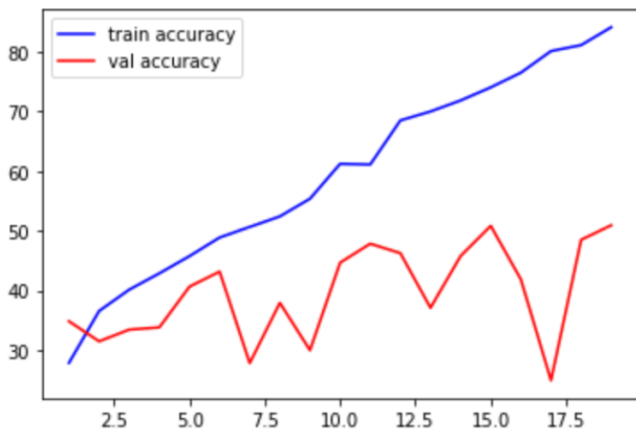
▲圖表四之二、第一次訓練與測試損耗
(輸入層 120x120)



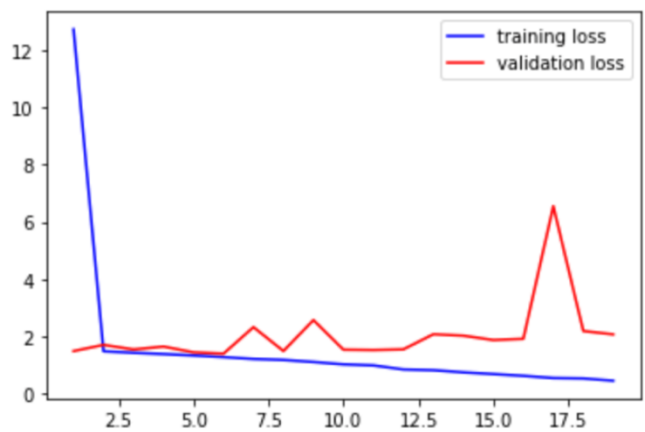
▲圖表四之三、第二次訓練與測試精準度
(輸入層 120x120)



▲圖表四之四、第二次訓練與測試損耗
(輸入層 120x120)



▲圖表四之五、第三次訓練與測試精準度
(輸入層 120x120)



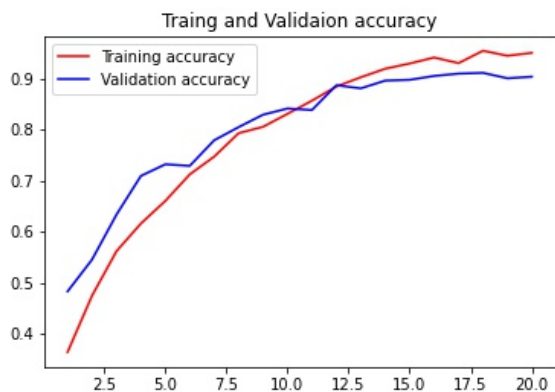
▲圖表四之六、第三次訓練與測試損耗
(輸入層 120x120)

二、TensorFlow 花卉辨識器

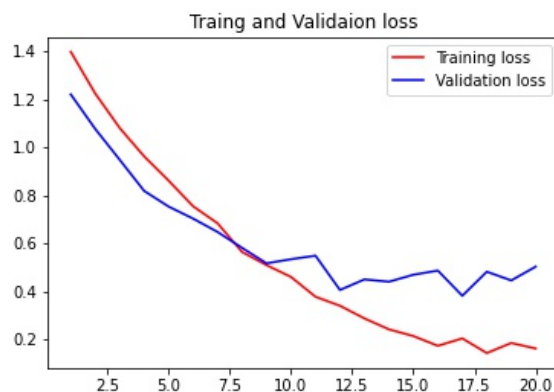
1. 由圖表五、六與七可以得知在輸入層為(120,120,3)時在訓練集與測試集的準確度上達到 98.32% 與 91.30%。

▼圖表五、輸入層為(120,120,3)

		第一次		第二次		第三次	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1/20	0.3647	0.4833	0.9565	0.9106	0.9788	0.9197
Epoch	2/20	0.4748	0.5455	0.9565	0.9000	0.9818	0.9091
Epoch	3/20	0.5619	0.6333	0.9659	0.9061	0.9799	0.9121
Epoch	4/20	0.6160	0.7091	0.9569	0.9091	0.9822	0.9136
Epoch	5/20	0.6602	0.7318	0.9580	0.9045	0.9830	0.9136
Epoch	6/20	0.7121	0.7288	0.9735	0.9106	0.9837	0.9182
Epoch	7/20	0.7469	0.7788	0.9667	0.8985	0.9875	0.9152
Epoch	8/20	0.7927	0.8045	0.9697	0.9167	0.9799	0.9076
Epoch	9/20	0.8051	0.8288	0.9758	0.9091	0.9845	0.9167
Epoch	10/20	0.8305	0.8409	0.9792	0.9136	0.9841	0.9121
Epoch	11/20	0.8562	0.8379	0.9849	0.9152	0.9883	0.9182
Epoch	12/20	0.8838	0.8864	0.9799	0.9121	0.9856	0.9030
Epoch	13/20	0.9020	0.8803	0.9754	0.9212	0.9841	0.9061
Epoch	14/20	0.9187	0.8955	0.9629	0.9152	0.9773	0.9091
Epoch	15/20	0.9289	0.8970	0.9671	0.9136	0.9849	0.9061
Epoch	16/20	0.9406	0.9045	0.9705	0.9197	0.9807	0.9167
Epoch	17/20	0.9296	0.9091	0.9773	0.9030	0.9792	0.9167
Epoch	18/20	0.9538	0.9106	0.9697	0.8924	0.9898	0.9152
Epoch	19/20	0.9440	0.9000	0.9743	0.9091	0.9864	0.9197
Epoch	20/20	0.9497	0.9030	0.9777	0.9227	0.9815	0.9106
AVG.		0.7886	0.8005	0.9699	0.9102	0.9832	0.9130



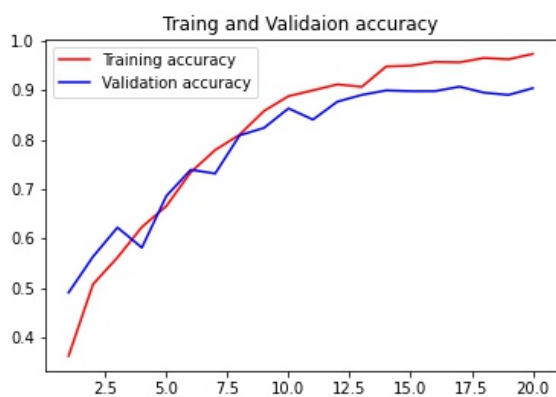
▲圖表五之一、訓練與測試精準度
(輸入層 120x120)



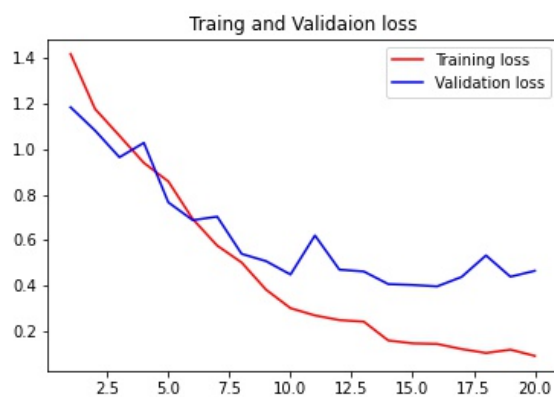
▲圖表五之二、訓練與測試損耗
(輸入層 120x120)

▼圖表六、輸入層為(150,150,3)

		第一次		第二次		第三次	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1/20	0.3628	0.4909	0.9330	0.8591	0.9705	0.9091
Epoch	2/20	0.5078	0.5636	0.9361	0.9000	0.9644	0.9136
Epoch	3/20	0.5619	0.6227	0.9463	0.9015	0.9747	0.8985
Epoch	4/20	0.6235	0.5818	0.9425	0.9106	0.9724	0.9076
Epoch	5/20	0.6659	0.6864	0.9467	0.9030	0.9690	0.9106
Epoch	6/20	0.7344	0.7394	0.9508	0.9076	0.9754	0.9106
Epoch	7/20	0.7794	0.7318	0.9519	0.8909	0.9784	0.9136
Epoch	8/20	0.8104	0.8091	0.9417	0.9076	0.9724	0.9106
Epoch	9/20	0.8585	0.8242	0.9531	0.9106	0.9599	0.8939
Epoch	10/20	0.8884	0.8636	0.9595	0.9091	0.9625	0.8955
Epoch	11/20	0.9001	0.8409	0.9659	0.9076	0.9762	0.9030
Epoch	12/20	0.9122	0.8773	0.9595	0.9106	0.9796	0.9030
Epoch	13/20	0.9073	0.8909	0.9603	0.9106	0.9743	0.9000
Epoch	14/20	0.9485	0.9000	0.9671	0.9152	0.9754	0.9121
Epoch	15/20	0.9501	0.8985	0.9512	0.8955	0.9788	0.9045
Epoch	16/20	0.9576	0.8985	0.9622	0.9136	0.9811	0.9106
Epoch	17/20	0.9569	0.9076	0.9622	0.9197	0.9845	0.9121
Epoch	18/20	0.9656	0.8955	0.9712	0.9106	0.9720	0.9121
Epoch	19/20	0.9629	0.8909	0.9671	0.9106	0.9811	0.9106
Epoch	20/20	0.9739	0.9045	0.9678	0.9136	0.9807	0.9030
AVG.		0.8114	0.7909	0.9548	0.9054	0.9742	0.9067



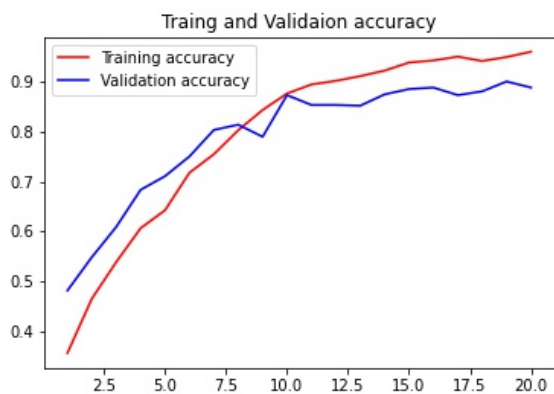
▲圖表六之一、訓練與測試精準度
(輸入層 150x150)



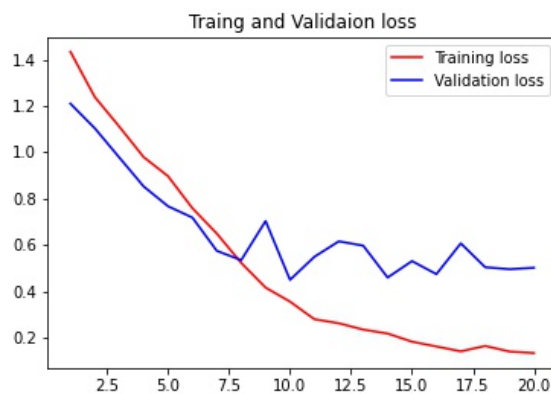
▲圖表六之二、訓練與測試損耗
(輸入層 150x150)

▼圖表七、輸入層為(180,180,3)

		第一次		第二次		第三次	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1/20	0.3568	0.4818	0.9330	0.8939	0.9777	0.9045
Epoch	2/20	0.4654	0.5485	0.9629	0.8879	0.9701	0.9091
Epoch	3/20	0.5388	0.6091	0.9648	0.8894	0.9724	0.8924
Epoch	4/20	0.6065	0.6833	0.9538	0.8894	0.9724	0.9106
Epoch	5/20	0.6425	0.7106	0.9489	0.8803	0.9777	0.9061
Epoch	6/20	0.7174	0.7500	0.9557	0.8955	0.9735	0.9197
Epoch	7/20	0.7544	0.8030	0.9580	0.8818	0.9856	0.9091
Epoch	8/20	0.8025	0.8136	0.9557	0.9106	0.9803	0.9045
Epoch	9/20	0.8430	0.7894	0.9629	0.8848	0.9834	0.9030
Epoch	10/20	0.8759	0.8727	0.9659	0.8939	0.9834	0.9076
Epoch	11/20	0.8941	0.8530	0.9671	0.8864	0.9788	0.9121
Epoch	12/20	0.9012	0.8530	0.9599	0.9091	0.9750	0.9076
Epoch	13/20	0.9107	0.8515	0.9709	0.9030	0.9762	0.8970
Epoch	14/20	0.9217	0.8742	0.9728	0.910	0.9781	0.8985
Epoch	15/20	0.9379	0.8848	0.9709	0.8985	0.9720	0.9015
Epoch	16/20	0.9421	0.8879	0.9720	0.9030	0.9777	0.9000
Epoch	17/20	0.9497	0.8727	0.9694	0.9076	0.9728	0.9045
Epoch	18/20	0.9410	0.8803	0.9622	0.9045	0.9792	0.8955
Epoch	19/20	0.9489	0.9000	0.9781	0.9061	0.9807	0.9015
Epoch	20/20	0.9595	0.8873	0.9750	0.9076	0.9765	0.9061
AVG.		0.7955	0.7903	0.9630	0.8972	0.9772	0.9045



▲圖表七之一、訓練與測試精準度
(輸入層 180x180)

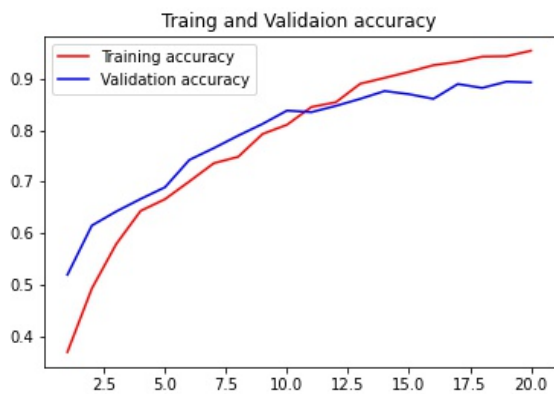


▲圖表七之二、訓練與測試損耗
(輸入層 180x180)

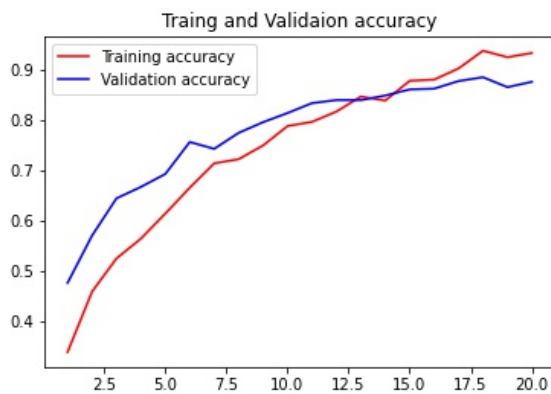
2. 圖表八紀錄當 batch size 分別設為 32、64 的情況在第一次測試中 batch size32 在訓練集準確度較優，而在測試集上略精確於 batch size64。

▼圖表八、Batch-Size

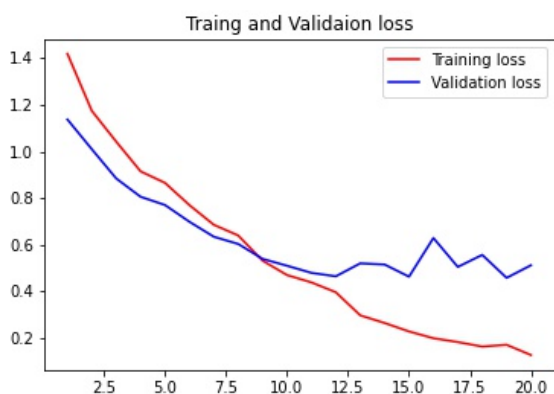
		Batch size = 32		Batch size = 64	
		訓練	測試	訓練	測試
Epoch	1/20	0.3700	0.5197	0.3383	0.4758
Epoch	2/20	0.4930	0.6152	0.4582	0.5697
Epoch	3/20	0.5793	0.6424	0.5244	0.6439
Epoch	4/20	0.6436	0.6667	0.5638	0.6667
Epoch	5/20	0.6663	0.6894	0.6137	0.6924
Epoch	6/20	0.7007	0.7424	0.6652	0.7561
Epoch	7/20	0.7359	0.7652	0.7136	0.7424
Epoch	8/20	0.7484	0.7894	0.7219	0.7742
Epoch	9/20	0.7927	0.8121	0.7491	0.7955
Epoch	10/20	0.8104	0.8379	0.7877	0.8136
Epoch	11/20	0.8449	0.8348	0.7961	0.8333
Epoch	12/20	0.8540	0.8470	0.8165	0.8394
Epoch	13/20	0.8899	0.8606	0.8464	0.8394
Epoch	14/20	0.9012	0.8758	0.8384	0.8485
Epoch	15/20	0.9130	0.8697	0.8778	0.8606
Epoch	16/20	0.9258	0.8606	0.8801	0.8621
Epoch	17/20	0.9323	0.8894	0.9024	0.8773
Epoch	18/20	0.9425	0.8818	0.9376	0.8848
Epoch	19/20	0.9432	0.8939	0.9247	0.8652
Epoch	20/20	0.9538	0.8924	0.9330	0.8758
AVG.		0.7820	0.7893	0.7444	0.7758



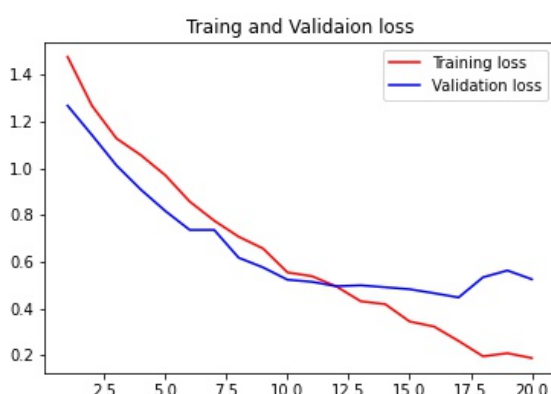
▲圖表八之一、訓練與測試精準度
(batch-size = 32)



▲圖表八之二、訓練與測試精準度
(batch-size = 64)



▲圖表八之三、訓練與測試損耗
(batch-size = 32)

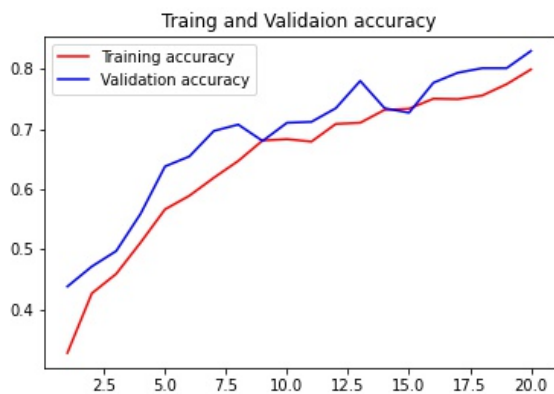


▲圖表八之四、訓練與測試損耗
(batch-size = 64)

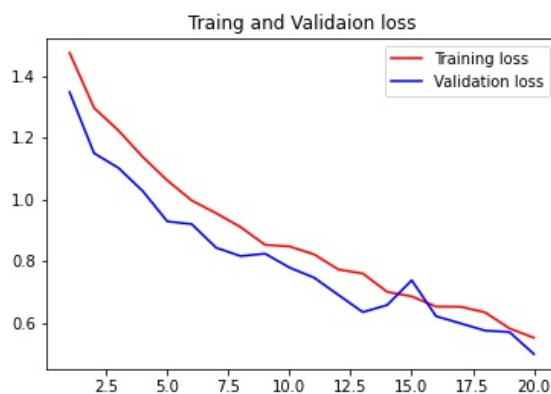
3. 圖表九嘗試對資料集進行 Data Augmentation(數據增強)，進行 Randomflip()水平與垂直翻轉，第一次的訓練數值偏低，因此我們訓練了三次，然而成效並未有過多的提升，儘管到了第三次訓練集準確度可以達到 93%以上，然而從測試集的準確度可以明顯看出來模型已呈現 overfitting (過度擬合)的狀態。

▼圖表九、Randomflip()

		第一次		第二次		第三次	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1/20	0.3273	0.4379	0.7988	0.8182	0.8861	0.8818
Epoch	2/20	0.4264	0.4712	0.8070	0.8364	0.8994	0.8788
Epoch	3/20	0.4589	0.4970	0.8180	0.8061	0.8944	0.8833
Epoch	4/20	0.5112	0.5591	0.8089	0.8242	0.8994	0.8970
Epoch	5/20	0.5664	0.6379	0.8214	0.8182	0.9035	0.8970
Epoch	6/20	0.5891	0.6545	0.8316	0.8121	0.9168	0.8591
Epoch	7/20	0.6190	0.6970	0.8271	0.8273	0.9179	0.8879
Epoch	8/20	0.6470	0.7076	0.8392	0.8394	0.9240	0.9076
Epoch	9/20	0.6810	0.6803	0.8453	0.8667	0.9156	0.8606
Epoch	10/20	0.6833	0.7106	0.8524	0.8682	0.9130	0.9000
Epoch	11/20	0.6792	0.7121	0.8517	0.8848	0.9217	0.9106
Epoch	12/20	0.7087	0.7348	0.8509	0.8485	0.9202	0.8818
Epoch	13/20	0.7106	0.7803	0.8702	0.8773	0.9251	0.8879
Epoch	14/20	0.7321	0.7348	0.8774	0.8682	0.9258	0.8697
Epoch	15/20	0.7340	0.7273	0.8683	0.8864	0.9300	0.8833
Epoch	16/20	0.7507	0.7773	0.8740	0.8848	0.9357	0.8970
Epoch	17/20	0.7499	0.7939	0.8680	0.8833	0.9262	0.8818
Epoch	18/20	0.7560	0.8015	0.8910	0.8848	0.9209	0.9197
Epoch	19/20	0.7749	0.8015	0.8895	0.8864	0.9391	0.8909
Epoch	20/20	0.7995	0.8303	0.8952	0.8682	0.9224	0.8970
AVG.		0.6453	0.6873	0.8493	0.8545	0.9169	0.8886



▲圖表九之一、訓練與測試精準度
(randomflip)

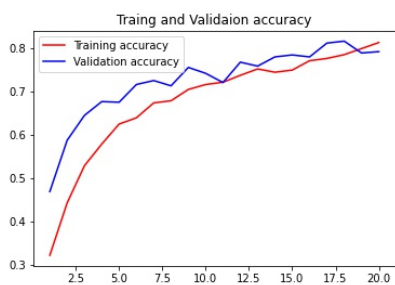


▲圖表九之二、訓練與測試損耗
(randomflip)

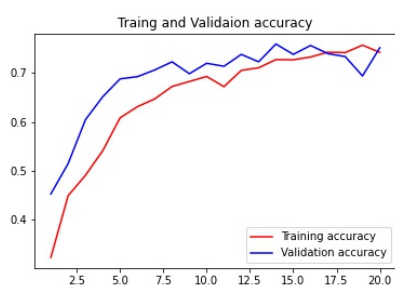
4. 圖表十嘗試對資料集進行 Data Augmentation(數據增強)，使用 RandomRotation()分別以 0.1、0.2、0.3 的比例旋轉，然而我們可以 r 僅僅在 0.1 的狀況下唯一超過 80%的準確度，我們推測可能跟訓練次數過少有關。

▼圖表十、RandonRotation ()

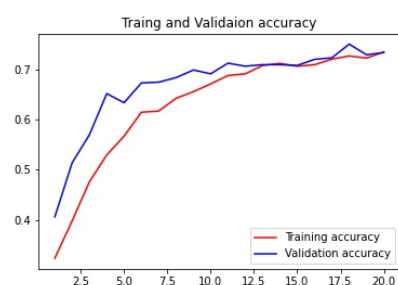
		0.1		0.2		0.3	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1/20	0.3227	0.4697	0.3235	0.4530	0.3235	0.4061
Epoch	2/20	0.4434	0.5879	0.4495	0.5152	0.3980	0.5136
Epoch	3/20	0.5293	0.6455	0.4911	0.6045	0.4760	0.5697
Epoch	4/20	0.5793	0.6773	0.5414	0.6515	0.5293	0.6515
Epoch	5/20	0.6254	0.6758	0.6084	0.6879	0.5668	0.6333
Epoch	6/20	0.6398	0.7167	0.6311	0.6924	0.6145	0.6727
Epoch	7/20	0.6742	0.7258	0.6470	0.7061	0.6167	0.6742
Epoch	8/20	0.6792	0.7136	0.6723	0.7227	0.6421	0.6833
Epoch	9/20	0.7053	0.7561	0.6826	0.6985	0.6553	0.6985
Epoch	10/20	0.7166	0.7424	0.6928	0.7197	0.6708	0.6909
Epoch	11/20	0.7219	0.7212	0.6720	0.7136	0.6875	0.7121
Epoch	12/20	0.7378	0.7682	0.7053	0.7379	0.6909	0.7061
Epoch	13/20	0.7526	0.7591	0.7106	0.7227	0.7075	0.7091
Epoch	14/20	0.7450	0.7803	0.7272	0.7591	0.7117	0.7091
Epoch	15/20	0.7499	0.7848	0.7268	0.7379	0.7060	0.7076
Epoch	16/20	0.7715	0.7803	0.7325	0.7561	0.7090	0.7197
Epoch	17/20	0.7768	0.8121	0.7423	0.7394	0.7200	0.7227
Epoch	18/20	0.7855	0.8167	0.7416	0.7333	0.7264	0.7500
Epoch	19/20	0.7995	0.7894	0.7567	0.6939	0.7223	0.7288
Epoch	20/20	0.8135	0.7924	0.7420	0.7515	0.7344	0.7333
AVG.		0.6785	0.7258	0.6498	0.6898	0.6304	0.6696



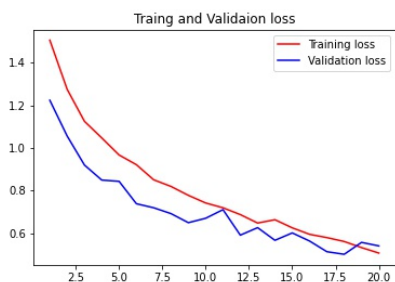
▲圖表十之一、訓練與測試精準度
(RandomRotation(0.1))



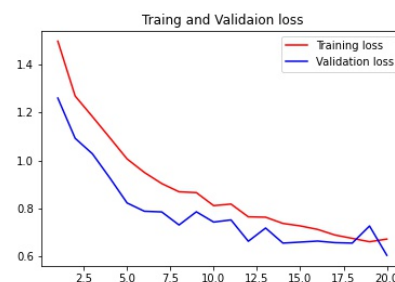
▲圖表十之二、訓練與測試精準度
(RandomRotation(0.2))



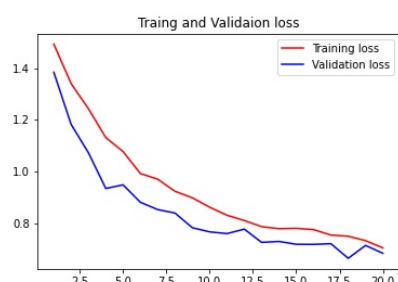
▲圖表十之三、訓練與測試精準度
(RandomRotation(0.3))



▲圖表十之四、訓練與測試損耗
(RandomRotation(0.1))



▲圖表十之五、訓練與測試損耗
(RandomRotation(0.2))

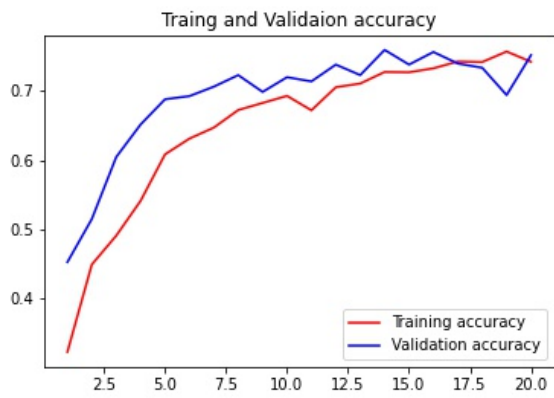


▲圖表十之六、訓練與測試損耗
(RandomRotation(0.3))

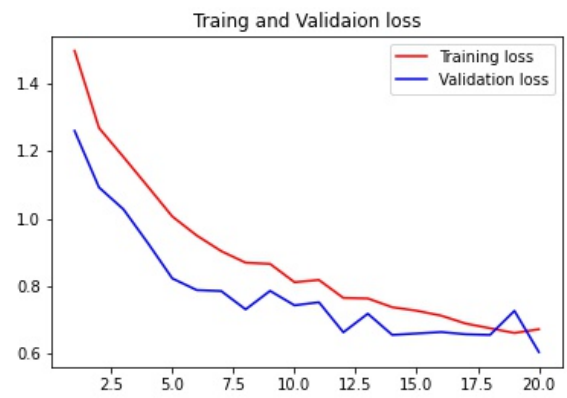
5. 圖表十一嘗試對資料集進行 Data Augmentation(數據增強)。鑒於前述之結果推論為訓練次數過少，因此在圖表七我們使用 RandomRotation(0.1)重複訓練三次，我們可以看到最後的準確度有些微的上升，訓練集準確度最高將近 92%，然而速度過慢，因此我們嘗試透過調整其他參數來達到同樣效果。

▼圖表十一、RandonRotation (0.1)

		第一次		第二次		第三次	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1/20	0.3227	0.4697	0.7889	0.7667	0.8801	0.8788
Epoch	2/20	0.4434	0.5879	0.7942	0.8197	0.8816	0.8682
Epoch	3/20	0.5293	0.6455	0.8150	0.8197	0.8801	0.8682
Epoch	4/20	0.5793	0.6773	0.8089	0.8061	0.8857	0.8712
Epoch	5/20	0.6254	0.6758	0.8169	0.8409	0.8952	0.8485
Epoch	6/20	0.6398	0.7167	0.8142	0.8030	0.8714	0.8379
Epoch	7/20	0.6742	0.7258	0.8120	0.8303	0.8785	0.8667
Epoch	8/20	0.6792	0.7136	0.8358	0.8273	0.8785	0.8485
Epoch	9/20	0.7053	0.7561	0.8464	0.8061	0.8884	0.8697
Epoch	10/20	0.7166	0.7424	0.8411	0.8409	0.9031	0.9000
Epoch	11/20	0.7219	0.7212	0.8589	0.8348	0.8888	0.8636
Epoch	12/20	0.7378	0.7682	0.8536	0.8636	0.8861	0.8909
Epoch	13/20	0.7526	0.7591	0.8513	0.8394	0.9024	0.8803
Epoch	14/20	0.7450	0.7803	0.8392	0.8394	0.9005	0.8818
Epoch	15/20	0.7499	0.7848	0.8570	0.8242	0.8903	0.8758
Epoch	16/20	0.7715	0.7803	0.8581	0.8439	0.9016	0.8985
Epoch	17/20	0.7768	0.8121	0.8619	0.8470	0.9126	0.8955
Epoch	18/20	0.7855	0.8167	0.8664	0.8561	0.9156	0.8803
Epoch	19/20	0.7995	0.7894	0.8710	0.8606	0.9111	0.8712
Epoch	20/20	0.8135	0.7924	0.8721	0.8606	0.9156	0.8924
AVG.		0.6785	0.7258	0.8381	0.8315	0.8934	0.8744



▲圖表十一之一、訓練與測試精準度
(randonrotation)



▲圖表十一之二、訓練與測試損耗
(randonrotation)

6. 圖表十二調整 epoch 數量 10、20、30、40，以 epoch10 表現為最差，epoch40 雖然有 100% 的預測，然後已經發生了嚴重的 overfitting (過度擬合) 的問題，因此使用相較平穩的 epoch20 或 epoch30。

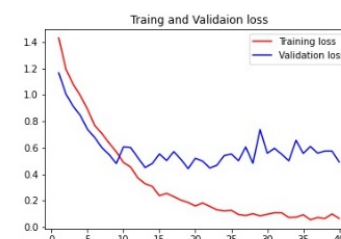
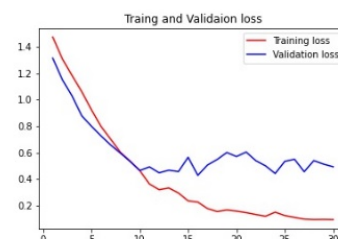
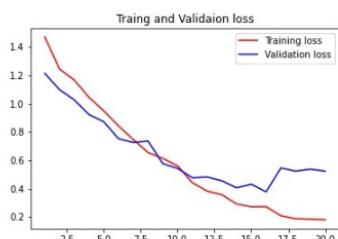
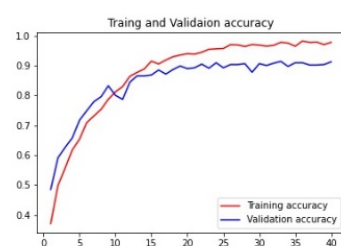
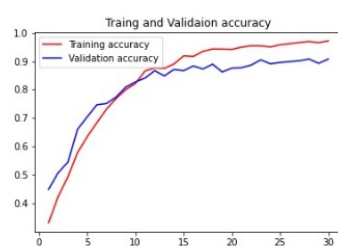
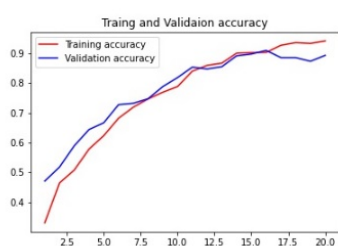
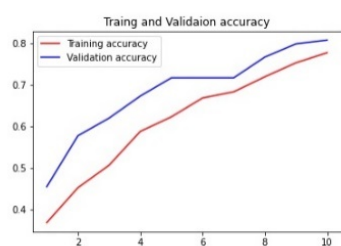
▼圖表十二、Epoch=10, 20, 30

		Epoch = 10		Epoch = 20		Epoch =30	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1	0.3681	0.4545	0.3314	0.4712	0.3330	0.4500
Epoch	2	0.4521	0.5773	0.4650	0.5182	0.4257	0.5091
Epoch	3	0.5062	0.6197	0.5081	0.5894	0.4941	0.5455
Epoch	4	0.5876	0.6727	0.5785	0.6439	0.5789	0.6606
Epoch	5	0.6228	0.7167	0.6235	0.6667	0.6349	0.7045
Epoch	6	0.6682	0.7167	0.6818	0.7273	0.6841	0.7470
Epoch	7	0.6829	0.7167	0.7189	0.7318	0.7314	0.7515
Epoch	8	0.7193	0.7667	0.7465	0.7470	0.7692	0.7742
Epoch	9	0.7529	0.7985	0.7688	0.7879	0.8002	0.8091
Epoch	10	0.7771	0.8076	0.7881	0.8182	0.8218	0.8273
Epoch	11	-	-	0.8396	0.8530	0.8657	0.8409
Epoch	12	-	-	0.8589	0.8470	0.8763	0.8667
Epoch	13	-	-	0.8664	0.8545	0.8748	0.8485
Epoch	14	-	-	0.8997	0.8909	0.8903	0.8712
Epoch	15	-	-	0.9016	0.8970	0.9187	0.8667
Epoch	16	-	-	0.9024	0.9091	0.9168	0.8833
Epoch	17	-	-	0.9262	0.8848	0.9345	0.8727
Epoch	18	-	-	0.9353	0.8848	0.9429	0.8894
Epoch	19	-	-	0.9327	0.8727	0.9425	0.8621
Epoch	20	-	-	0.9410	0.8924	0.9414	0.8758
Epoch	21	-	-	-	-	0.9493	0.8773
Epoch	22	-	-	-	-	0.9542	0.8864
Epoch	23	-	-	-	-	0.9538	0.9045
Epoch	24	-	-	-	-	0.9501	0.8909
Epoch	25	-	-	-	-	0.9576	0.8955
Epoch	26	-	-	-	-	0.9614	0.8985
Epoch	27	-	-	-	-	0.9656	0.9015
Epoch	28	-	-	-	-	0.9690	0.9076
Epoch	29	-	-	-	-	0.9652	0.8924
Epoch	30	-	-	-	-	0.9712	0.9076
AVG.		0.6137	0.6847	0.7607	0.7744	0.8325	0.8139
PRED,		83.74%		97.69%		99.51%	

▼圖表十二、Epoch=40

Epoch = 40

		訓練	測試			訓練	測試
Epoch	1	0.3062	0.4848	Epoch	21	0.9330	0.8924
Epoch	2	0.4805	0.5909	Epoch	22	0.9473	0.9045
Epoch	3	0.5432	0.6258	Epoch	23	0.9556	0.8909
Epoch	4	0.6159	0.6576	Epoch	24	0.9420	0.9091
Epoch	5	0.6551	0.7167	Epoch	25	0.9516	0.8924
Epoch	6	0.7022	0.7485	Epoch	26	0.9691	0.9030
Epoch	7	0.7265	0.7788	Epoch	27	0.9659	0.9030
Epoch	8	0.7644	0.7955	Epoch	28	0.9575	0.9061
Epoch	9	0.7821	0.8318	Epoch	29	0.9741	0.8773
Epoch	10	0.8065	0.8000	Epoch	30	0.9629	0.9061
Epoch	11	0.8282	0.7864	Epoch	31	0.9654	0.9000
Epoch	12	0.8550	0.8439	Epoch	32	0.9659	0.9076
Epoch	13	0.8644	0.8652	Epoch	33	0.9793	0.9136
Epoch	14	0.8805	0.8652	Epoch	34	0.9749	0.8970
Epoch	15	0.9090	0.8682	Epoch	35	0.9537	0.9091
Epoch	16	0.8946	0.8848	Epoch	36	0.9790	0.9091
Epoch	17	0.9164	0.8712	Epoch	37	0.9786	0.9015
Epoch	18	0.9207	0.8864	Epoch	38	0.9793	0.9015
Epoch	19	0.9322	0.8985	Epoch	39	0.9678	0.9030
Epoch	20	0.9473	0.8894	Epoch	40	0.9793	0.9121
				AVG.	0.8653	0.8432	
				PRED.	100%		



▲圖表十二之二、訓練與測試損耗

(epoch=10,20,30,40)

7. 圖表十三為調整 epoch 數量，然而是將圖片以灰階處理後再進行測試，以 epoch30 在灰階圖片上表現最佳。

▼圖表十三、Epoch

		Epoch = 15		Epoch = 30	
		訓練	測試	訓練	測試
Epoch	1	0.2256	0.2455	0.2350	0.3000
Epoch	2	0.2623	0.3000	0.3126	0.3727
Epoch	3	0.3147	0.3545	0.3612	0.4470
Epoch	4	0.3631	0.4424	0.4097	0.4939
Epoch	5	0.4108	0.4621	0.4731	0.4439
Epoch	6	0.4482	0.5318	0.4718	0.4924
Epoch	7	0.4785	0.5636	0.5362	0.5924
Epoch	8	0.5315	0.6167	0.5691	0.4985
Epoch	9	0.5639	0.6500	0.5925	0.6136
Epoch	10	0.6105	0.6652	0.6594	0.5955
Epoch	11	0.6383	0.7045	0.6747	0.7500
Epoch	12	0.6605	0.7379	0.7209	0.7409
Epoch	13	0.6988	0.7485	0.7288	0.7394
Epoch	14	0.7069	0.7652	0.7747	0.7909
Epoch	15	0.7252	0.7621	0.7835	0.8106
Epoch	16	-	-	0.8039	0.8273
Epoch	17	-	-	0.8302	0.8424
Epoch	18	-	-	0.8655	0.8303
Epoch	19	-	-	0.8608	0.8318
Epoch	20	-	-	0.8732	0.8652
Epoch	21	-	-	0.8707	0.8545
Epoch	22	-	-	0.8792	0.8515
Epoch	23	-	-	0.8938	0.8561
Epoch	24	-	-	0.9054	0.8591
Epoch	25	-	-	0.9108	0.8394
Epoch	26	-	-	0.9130	0.8667
Epoch	27	-	-	0.9027	0.8500
Epoch	28	-	-	0.9172	0.8636
Epoch	29	-	-	0.9295	0.8561
Epoch	30	-	-	0.9191	0.8773
AVG.		0.2546	0.285	0.7193	0.7151
PRED,		89.72%		97.22%	

8. 圖表十四與圖表十五將做使用灰階圖片與使用彩虹圖片預測比較，在訓練過程中可以看到在相同的條件下，以灰階圖片訓練相較起來成長較為平均，而以彩色圖片在這樣的條件下訓練是最為準確的。

▼圖表十四、Grey

		第一次		第二次		第三次	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1	0.2191	0.2773	0.2367	0.3030	0.2339	0.2424
Epoch	2	0.2449	0.3273	0.2802	0.3303	0.2499	0.3076
Epoch	3	0.3224	0.3136	0.3369	0.3879	0.3197	0.3879
Epoch	4	0.3585	0.3985	0.3553	0.4394	0.3773	0.4606
Epoch	5	0.3855	0.4697	0.4462	0.4970	0.4088	0.4530
Epoch	6	0.4278	0.4697	0.4672	0.5167	0.4438	0.5242
Epoch	7	0.4807	0.4742	0.5273	0.5455	0.4867	0.5455
Epoch	8	0.5094	0.5652	0.5522	0.5712	0.5415	0.6136
Epoch	9	0.5407	0.5955	0.5930	0.6045	0.5944	0.5727
Epoch	10	0.6111	0.6970	0.6319	0.6606	0.6137	0.6318
Epoch	11	0.6333	0.6530	0.6724	0.6485	0.6630	0.7273
Epoch	12	0.6758	0.6879	0.6889	0.6879	0.6849	0.7258
Epoch	13	0.6947	0.7636	0.7140	0.7273	0.7376	0.7621
Epoch	14	0.7186	0.7788	0.7439	0.7636	0.7592	0.7439
Epoch	15	0.7410	0.7667	0.7649	0.7530	0.7948	0.7955
Epoch	16	0.7665	0.8030	0.7811	0.7833	0.7954	0.7939
Epoch	17	0.7998	0.8076	0.8035	0.7970	0.8298	0.8227
Epoch	18	0.8205	0.8197	0.7971	0.8015	0.8593	0.8212
Epoch	19	0.8316	0.8470	0.8094	0.7985	0.8609	0.8242
Epoch	20	0.8438	0.8288	0.8281	0.7864	0.8896	0.8303
Epoch	21	0.8664	0.8470	0.8235	0.7848	0.8968	0.8030
Epoch	22	0.8784	0.8682	0.8224	0.8212	0.8936	0.8364
Epoch	23	0.8908	0.8712	0.8700	0.8212	0.9026	0.8470
Epoch	24	0.8973	0.8697	0.8719	0.8106	0.9169	0.8439
Epoch	25	0.8986	0.8682	0.8691	0.7985	0.8975	0.8455
Epoch	26	0.9024	0.8652	0.8634	0.8242	0.9035	0.8636
Epoch	27	0.9208	0.8500	0.8791	0.8182	0.9260	0.8530
Epoch	28	0.9215	0.8576	0.8886	0.8288	0.9250	0.8485
Epoch	29	0.9147	0.8621	0.8911	0.8303	0.9297	0.8636
Epoch	30	0.9032	0.8515	0.8970	0.8273	0.9405	0.8591
AVG.		0.6873	0.698	0.6902	0.6856	0.7092	0.7017
PRED,		98.60%		94.20%		99.84%	

▼圖表十五、RGB

		第一次		第二次		第三次	
		訓練	測試	訓練	測試	訓練	測試
Epoch	1	0.2485	0.3758	0.2797	0.4864	0.2918	0.4712
Epoch	2	0.3955	0.5182	0.4720	0.5409	0.4660	0.5576
Epoch	3	0.4875	0.6197	0.5520	0.5773	0.5447	0.6348
Epoch	4	0.5681	0.6667	0.6108	0.6197	0.5931	0.7030
Epoch	5	0.6084	0.6864	0.6423	0.6045	0.6522	0.7045
Epoch	6	0.6475	0.7303	0.6697	0.6545	0.7091	0.7576
Epoch	7	0.6922	0.7439	0.6994	0.7045	0.7483	0.7394
Epoch	8	0.7417	0.7924	0.7475	0.7227	0.8045	0.7652
Epoch	9	0.7603	0.7879	0.7776	0.7485	0.8288	0.8182
Epoch	10	0.7673	0.8000	0.7944	0.8106	0.8248	0.8485
Epoch	11	0.8219	0.8288	0.8081	0.8076	0.8639	0.8333
Epoch	12	0.8254	0.8318	0.8369	0.8258	0.8838	0.8485
Epoch	13	0.8449	0.8409	0.8623	0.8394	0.8987	0.8758
Epoch	14	0.8475	0.8091	0.8876	0.8712	0.9249	0.8742
Epoch	15	0.8663	0.8591	0.8985	0.8833	0.9253	0.8712
Epoch	16	0.8994	0.8712	0.9196	0.8939	0.9368	0.8955
Epoch	17	0.8985	0.8667	0.9301	0.8894	0.9532	0.8864
Epoch	18	0.9015	0.8727	0.9182	0.9045	0.9447	0.9000
Epoch	19	0.9045	0.8606	0.9458	0.8818	0.9576	0.8985
Epoch	20	0.9066	0.8788	0.9425	0.9091	0.9549	0.9015
Epoch	21	0.9322	0.8848	0.9537	0.9045	0.9542	0.9106
Epoch	22	0.9406	0.8955	0.9383	0.9152	0.9589	0.9091
Epoch	23	0.9338	0.9030	0.9576	0.9000	0.9714	0.8939
Epoch	24	0.9361	0.9015	0.9656	0.9167	0.9657	0.9106
Epoch	25	0.9459	0.9015	0.9633	0.8939	0.9768	0.9015
Epoch	26	0.9468	0.9106	0.9589	0.9152	0.9637	0.8879
Epoch	27	0.9572	0.9076	0.9682	0.8909	0.9564	0.8864
Epoch	28	0.9587	0.9061	0.9572	0.9106	0.9657	0.8424
Epoch	29	0.9633	0.9152	0.9533	0.9015	0.9580	0.9106
Epoch	30	0.9578	0.9136	0.9705	0.9106	0.9699	0.9121
AVG.		0.8035	0.8093	0.8261	0.8078	0.8449	0.8250
PRED,		100%		98.84%		100%	

陸、 組員心得

梁育誠：

這次的實作對我來說是一個新的嘗試，透過 cnn 模型的實作，發現了許多很酷的新玩意，除了課堂上的知識外，更能從網路上各位大神的身上學習，也感謝組員之間的配合與鼓勵。

陳亞萱：

這次的期末報告對我來說是一個相對困難的實作，畢竟學習到的內容是從來沒有碰觸過的領域，因此在實際演練上，只能透過查找許多網路上的資料以及觀看老師在課堂上提供的投影片，才逐步完成這次 CNN 的模型，其中我遇到最大的難關就是，每訓練一次模型就要花費 1.5 小時，不同的模式和參數都要重複訓練 3 次，而且中間還不能斷線，因此花費了非常久的時間才得到結果。

最後感謝組員的相互幫忙，才可以順利完成這次的期末專題，也謝謝老師這學期的認真教導，課堂中老師很都會用心的把實用的程式碼跟應用都完整的呈現給我們，讓我對於人工智慧有更佳的瞭解！

陳朱鑒：

透過這次的實作，讓我對 cnn 有更多的了解，原本上課聽的時候，覺得一層一層講解下來很容易理解，但是真正實作，才發現很多細節與邏輯並不是真正了解，對程式也是不理解。在撰寫書面報告時，也利用老師的 ppt 與網路資源而更清楚 cnn 的架構；在查有關程式的資料時，透過網路資料慢慢理解程式語言與邏輯，也透過組員的解釋更能了解程式的部分。謝謝組員之間的幫忙，才能完成這次的作品。

組內分工：

程式撰寫：陳亞萱(PyTorch)、梁育誠(TensorFlow)

書面報告：陳朱鑒

柒、 參考文獻

李兆健，2017，《卷積神經網路應用於中文字手寫風格辨識》。台南：國立成功大學工程科學系碩士論文。

36r，2020，〈Netflix 的個性化推薦演算法，重點居然是在這一張一張的海報縮

圖上?〉，T 客邦，[Netflix 的個性化推薦演算法，重點居然是在這一張一張的海報縮圖上？ | T 客邦 \(techbang.com\)](#)，取用日期：2021 年 5 月 30 日。

王郁倫，2021，〈鴻海研發非監督式學習演算法，智慧工廠人力再砍 50%！品管將被 AI 取代?〉，數位時代，[鴻海研發非監督式學習演算法，智慧工廠人力再砍 50%！品管將被 AI 取代？](#)

[\(bnext.com.tw\)](#)，取用日期：2021 年 5 月 30 日。

量子位，2017，〈20 個令人驚嘆的深度學習應用 Demo + Paper + Code〉，[20 個令人驚嘆的深度學習應用，Demo+Paper+Code - 每日頭條 \(kknews.cc\)](#)，取用日期：2021 年 5 月 30 日。

Derrick Mwiti，2021，〈10 Real-Life Applications of Reinforcement Learning〉，neptuneai，[10 Real-Life Applications of Reinforcement Learning - neptune.ai](#)，取用日期：2021 年 5 月 30 日。