

國立臺灣大學電機資訊學院資訊工程學系

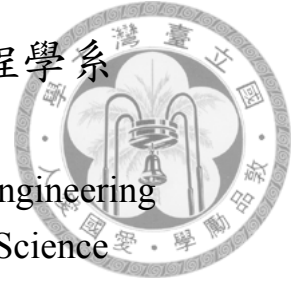
碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis



以 CNN 進行植物圖片的辨識以及經過處理後的再辨識
Plant Image Recognition with CNN and Re-classification

陳奕瑄

Yi-Hsuan Chen

指導教授：張智星博士

Advisor: Jyh-Shing Roger Jang, Ph.D.

中華民國 107 年 7 月

July, 2018



致謝

感謝造就現在的我的一切。



中文摘要

本研究的目標為利用卷積神經網路來辨識照片中的植物所歸屬的類別。在模型架構方面採用了知名的 VGG-16 模型，並使用遷移式學習 (transfer learning) 來降低學習至參數收斂所需要的時間。本次使用的資料集為擁有 500 種品種、約 10 萬張圖片的植物照片資料集。為了對植物特徵或主體較不明顯的圖片更進一步的處理，在一般的辨識過程後，加入了 Unseen Category Query Identification 辨識法來選取出這些辨識度不足的圖片，之後對選取出的圖片進行圖像分割，嘗試將植物主體與背景或雜訊分離，藉此來強調植物特徵。執行完前述步驟之後，選出包含植物主體的圖片，將其輸入模型進行再辨識，並實行其它實驗來比較這些做法的成效。

關鍵字：植物辨識、機器學習、深度學習、VGG 模型、卷積神經網路、圖像分割、K-means 分群法



Abstract

In this work, we want to recognize the species of plants in a picture by using Convolutional Neural Networks (CNN). We use the VGG-16 model in our experiments. To make the training process converged efficiently, we train the model by leveraging transfer learning. The dataset we use is made up of 500 species consisting of approximate 100,000 plant images. We employ Unseen Category Query Identification (UCQI) after the prediction step and picking those images which don't have obvious features or main bodies. For those picked images, image segmentation is used for separating plant from other objects and background noise. We choose the segmented images containing plant main body for re-classification. Detail comparisons between the proposed method and baselines are shown on the experimental part.

Key words: Plant recognition, machine learning, deep learning, VGG model, convolution neuron network, image segmentation, K-means clustering



目錄

致謝	i
中文摘要	ii
Abstract	iii
目錄	iv
圖片目錄	vi
表格目錄	viii
1 緒論	1
1.1 研究主題簡介	1
1.2 實驗方法簡介	2
1.3 章節概述	3
2 相關研究介紹	4
2.1 卷積神經網路簡介	4
2.1.1 卷積層	5
2.1.2 池化層	6
2.1.3 全連接層	6
2.1.4 Dropout	6
2.1.5 激勵函數	7
2.1.6 Softmax 層	9



2.2	AlexNet	9
2.3	VGG-16 模型	10
2.4	未知類別的辨識	10
2.5	圖像分割	13
2.5.1	K-平均分群法	13
3	實驗設定	16
3.1	資料集	16
3.2	模型參數設定	17
3.3	圖片前處理	20
3.3.1	資料增強	21
3.4	UCQI 的修改	22
3.5	圖片後處理	24
3.6	實驗環境	26
4	實驗結果與分析	27
4.1	模型架構與訓練方法	27
4.2	圖片再辨識	31
4.2.1	實驗一 新增訓練資料	34
4.2.2	實驗一 背景補色	37
5	結論與未來展望	40
5.1	回顧與結論	40
5.2	改進方向與未來展望	41
5.2.1	模型架構	41
5.2.2	測試方法	42
5.2.3	UCQI 的門檻值	43
5.2.4	圖片後處理	45
	參考文獻	48



圖片目錄

1.1	本次實驗的簡單示意圖	3
2.1	卷積神經網路架構示意圖	4
2.2	卷積示意圖	5
2.3	利用卷積進行邊緣偵測的例子	5
2.4	對特徵圖做最大池化 (max pooling) 的例子	6
2.5	過擬合 (overfitting)	7
2.6	激勵函數與其對應的輸出曲線	8
2.7	CNN 的辨識流程圖	9
2.8	AlexNet 架構圖	10
2.9	VGG-16 模型架構圖	11
2.10	假設情境：模型辨識三個輸入 α 、 β 、 γ 產生的結果	12
2.11	利用大津演算法進行二值化圖像分割	13
2.12	K-平均分群法	14
2.13	以 K-平均分群法進行圖像分割	15
3.1	同屬於植物：爵床的 3 張圖片	16
3.2	原資料集的圖片數量分布圖	17
3.3	Top-500 資料集的圖片數量分布圖	18
3.4	在 VGG-16 模型後加入新的輸出層	19
3.5	學習速率的變化圖	20
3.6	多重尺度訓練	22
3.7	隨機裁剪	22



3.8	與門檻值比較後決定是否採用測試資料的輸出做為預測結果	24
3.9	圖片進行圖像分割的結果與質心位置	25
4.1	AlexNet 訓練過程記錄	27
4.2	AlexNet 學習速率曲線圖	28
4.3	VGG-16 訓練過程記錄	29
4.4	VGG-16 (不使用遷移式學習) 訓練過程記錄	29
4.5	VGG-16 (不使用多重尺度學習) 訓練過程記錄	30
4.6	實驗流程圖	32
4.7	一些被選出的圖片	32
4.8	圖片分割與再辨識的結果	33
4.9	加入額外的訓練資料	34
4.10	加入額外的訓練資料後所進行的訓練記錄	35
4.11	加入分割圖片訓練模型後的再辨識結果	36
4.12	背景補色的流程圖	37
4.13	背景補色	38
4.14	經過圖像分割與背景補色後辨識正確的結果	38
4.15	原圖、分割圖與補色圖	39
5.1	殘差學習區塊	42
5.2	Inception 架構	42
5.3	多重尺度測試	43
5.4	多重圖片測試	43
5.5	不理想的圖像分割成果	46
5.6	不理想的分割圖片選擇成果	47



表格目錄

4.1	模型架構與辨識率的比較	30
4.2	不使用遷移式學習	30
4.3	不使用多重尺度訓練	31
4.4	綜合比較	31
4.5	加入額外的訓練資料	35
4.6	再辨識結果比較	39



Chapter 1

緒論

1.1 研究主題簡介

在最近幾年，由於硬體設備的進步，許多受限於硬體規格而無法實現的概念或是結果不理想的方法，都得到了改善與實踐，機器學習（machine learning）便是其中一個例子，記憶體的增加與硬體運算速度的成長，令研究者能夠使用更加深層與複雜的類神經網路架構。近年來，受惠於圖形處理器（graphics processing unit, GPU）的進步與其能夠進行平行運算的特性，許多研究開始使用 GPU 來加速類神經網路的訓練過程，大幅降低原本只使用 CPU 所需要的訓練時間，這項改變使得整個機器學習與類神經網路領域開始蓬勃發展。

隨著機器學習與類神經網路的相關研究不斷得到進展，電腦對於大數據的分析能力有著飛躍性的進步，這也令某些領域獲得突破性的發展。如何從一張圖片中辨識出特定的物件一直是電腦視覺領域中非常熱門的議題，試著讓電腦解讀圖片的內容，進而自動分類或分離出關鍵資訊，至今仍有許多學術研究不斷鑽研探討，回顧最近幾年，許多研究也開始使用類神經網路來處理這項議題，由 ImageNet 舉辦的 ILSVRC（ImageNet Large Scale Visual Recognition Competition）[1] 圖像辨識大賽，在 2012 年由類神經網路架構 AlexNet [2] 奪得冠軍後，後續幾年的冠軍皆是由類神經網路拿下，足見該領域的發展潛力，以及眾研究者的投入，而要利用類神經網路處理電腦視覺領域中圖像分類的問題，最常見的方法為建構出一個卷積神經網路（convolution neuron network, CNN）模型，利用圖形處理器進行深度學習（deep learning），讓模型學習到各種物件的特徵，

獲得從圖片中辨識物件的能力。關於卷積神經網路的整體架構以及模型是如何學習到特徵等將會在第 2.1 節說明。

在一般研究中，辨識目標並不限於任何領域，以 ILSVRC 為例，其所使用的資料庫 [1,3]，每張圖片皆被歸類在 1,000 個類別之中，這 1,000 類包含了各種動物、植物、物品等等。本篇研究不同的地方在於，目標是辨識出圖片中的植物所規屬的品種，因此將專注於植物辨識的部分，所使用之資料集只包含植物相關的圖片。類似的研究 [4] 中，其使用的資料集皆為植物葉子的圖片，對於辨識圖片中的葉子屬於何種植物的問題，利用卷積神經網路模型進行辨識的結果已經可以得到很高的準確率，但這些圖片是已經事先處理過，整張圖片當中保留完整的葉片輪廓，並完成去背處理，相較之下，本研究所使用的則是生活情境下可以取得的植物照片，較貼近一般的使用情況，同時因為原始圖片並沒有經過處理，可能包含其它與植物本身不相關的雜訊，成功辨識的難度也較高。

1.2 實驗方法簡介

本研究的目標為辨識出圖片中植物的品種。實驗分為兩個部分：第一部分為利用不同種類的植物圖片對卷積神經網路模型進行訓練，學習到各個品種的特徵後再進行辨識，以此評估這個模型的效能。本次所使用的模型為知名的 VGG-16 模型 [5]，同時實作遷移式學習 (transfer learning) [6]，載入預先利用巨量資料 ILSVRC-2012 資料集 [1,3] 訓練出的模型參數，再將本研究所使用的植物圖片資料集輸入進模型進行參數微調。另外，實驗當中會對訓練資料額外進行處理，並與用未處理資料進行訓練得出的模型進行比較，來驗證這些額外處理的成效；第二部分則是利用卷積神經網路輸出的結果，選出辨識度不足的圖片，並保留這些圖片的辨識結果。在預想中，可能是因為圖片的雜訊過多，導致植物本體的特徵無法被模型正確認識，因此在測試過程中被認定為辨識度不足。對於這些圖片，將會額外進行其它處理，以強調植物本身的特徵，具體來說，將對這些圖片進行圖像分割 (image segmentation)，嘗試將植物主體與背景分離，再利用分離後的圖片重新進行辨識，以得到較好的預測結果。

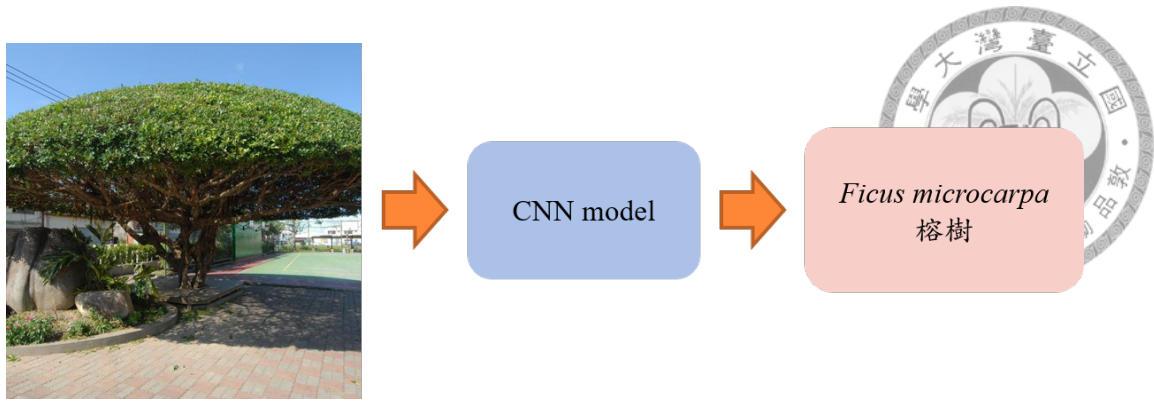


Figure 1.1: 本次實驗的簡單示意圖

1.3 章節概述

本論文共有四個章節：第一章為緒論，簡介本研究的主題背景與實驗方法；第二章將會介紹目前相關的研究，包含卷積神經網路的架構說明、本次所使用的網路模型、如何利用模型的輸出結果挑選辨識度不足的圖片、以及圖像分割的方法；第三章則是實驗的實作項目與細部設定，包含資料集的說明、既有方法的修改、模型的參數等細節；第四章為實驗結果，將會說明實驗項目與該實驗得到的結果；第五章為結論以及未來展望，說明本次實驗中得到的結論，以及之後對於實驗可能的修改方向與嘗試目標。



Chapter 2

相關研究介紹

2.1 卷積神經網路簡介

卷積神經網路為深度神經網路（ deep neuron network ）的一個應用。所謂的深度神經網路，主要是利用機器學習領域中的深度學習方法，建構出足夠多層（深度夠深）的模型，使得模型有足夠多的參數可以變動，並具有較高的學習能力，而卷積神經網路便是基於此項特點，架構出一個深度，並且可以學習圖片中物件的特徵，加以辨識的模型。卷積神經網路主要可分為兩個部分：擷取物件特徵的卷積層（ convolution layer ）、以及利用擷取出的特徵進行分類的全連階層（ fully-connected layer ）。

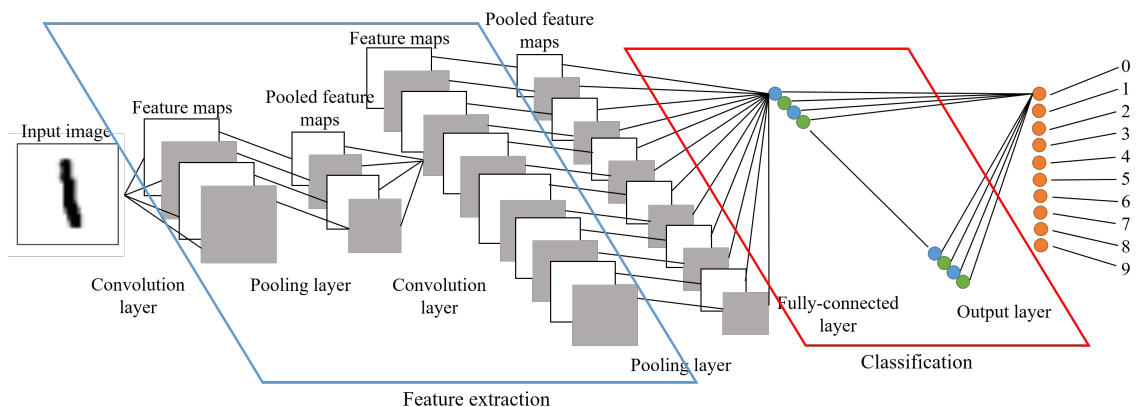


Figure 2.1: 卷積神經網路架構示意圖



2.1.1 卷積層

在電腦視覺領域中經常使用卷積 (convolution) 運算來進行影像處理，步驟如下 (Figure 2.2)：

- (1) 選定圖片上的一個點
- (2) 劃出以該點為中心的 $n \times n$ 範圍
- (3) 給予這 $n \times n$ 個點各自的權重 (weight)，這些權重稱為卷積核 (convolution kernel)
- (4) 將各點乘上權重後相加，這些動作會對圖片上所有的像素點都進行一次，最後算出的就是卷積的結果。

卷積一般用在去除雜點、影像銳化、以及邊緣偵測 (Figure 2.3) 等等，而在卷積神經網路中便是利用多個卷積核擷取出圖片不同的特徵來進行辨識，擷取出來的特徵所組成的資料稱為特徵圖 (feature map)；而在整個神經網路經由學習而變動參數時，卷積核的權重也會跟著更新，藉此擷取出更精確的特徵以利全連接層進行分類。

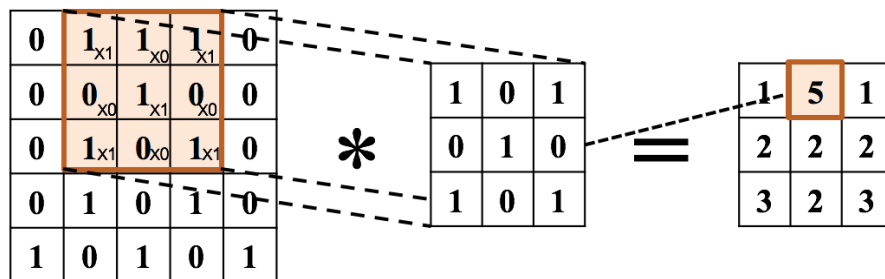


Figure 2.2: 卷積示意圖

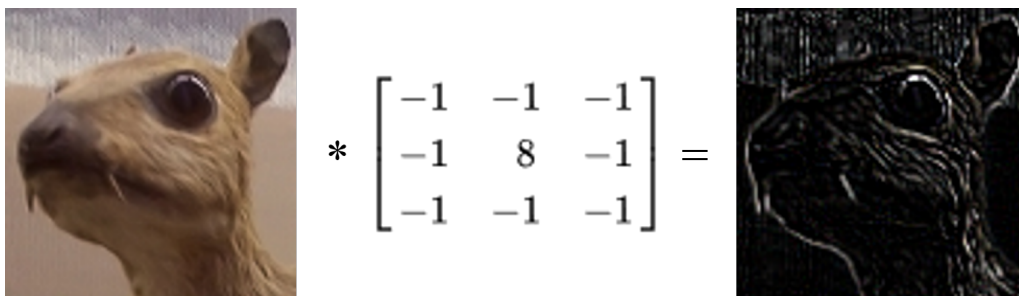


Figure 2.3: 利用卷積進行邊緣偵測的例子¹

¹ 圖片來源：[https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))，圖片授權皆採用創用 CC 姓名標示 - 相同方式分享 3.0，原圖片作者皆為 Michael Plotke



2.1.2 池化層

在卷積層之後，有時會加入池化層（pooling layer）來達到減少資料量以及凸顯特徵的目的。池化（pooling）與卷積類似，一開始會定義一個 $n \times n$ 的範圍，不同的是並不會給這些點各自的權重，而是從這些點中取出最大值來當作下一步的特徵；這樣做的好處是可以減少資料、降低運算量，且因為從中擷取出範圍中的最大值，也就是擷取出最具代表性的特徵，因此可以達成在減少資料量的同時也保留圖片中物件的特徵。

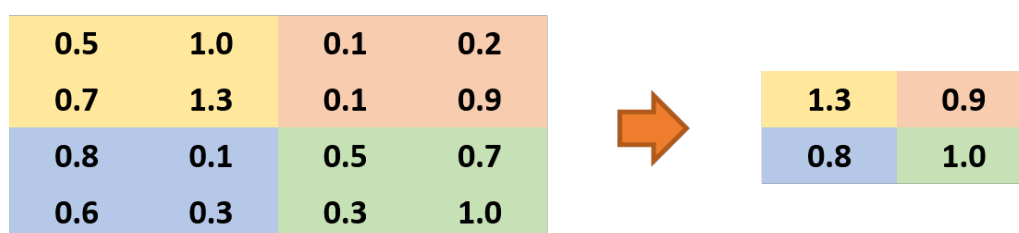


Figure 2.4: 對特徵圖做最大池化（max pooling）的例子

2.1.3 全連接層

在卷積層將圖片特徵擷取出來後，就會將這些特徵輸入至全連接層進行分類。在全連接層中，每一項特徵都會被給予權重，而不同的節點對於不同特徵所給予的權重也會不同，這代表著該節點所負責辨識的物件所具有的特徵，愈明顯或愈具代表性的特徵的權重值就會愈大，反之則愈小，而在卷積層擷取出的特徵代表的是圖片中物件的特徵，同樣地愈明顯的特徵值就愈大，因此當擷取出的特徵值與特徵對應到的權重兩者皆大時，該節點所輸出的值會就高於其他節點的值。

2.1.4 Dropout

Dropout [7] 由 G. E. Hinton 等人在 2012 年提出，並應用在前述的 AlexNet 之中。當訓練資料通過神經網路時，依照事先決定好的機率，刻意去「關閉」某些節點（具體的方法為，令這些節點的輸出為 0），使得每次訓練資料所經過、啟用的皆是不同的節點。實作 dropout 的目的在於防止模型太過偏向訓練資料所造成的過擬合（overfitting, Figure 2.5），如同前面所述，每次訓練資料經過模型時會

啟用不同的節點進行訓練，減少了層與層之間節點的關聯性，相對地增加更多的隨機性與泛用性。

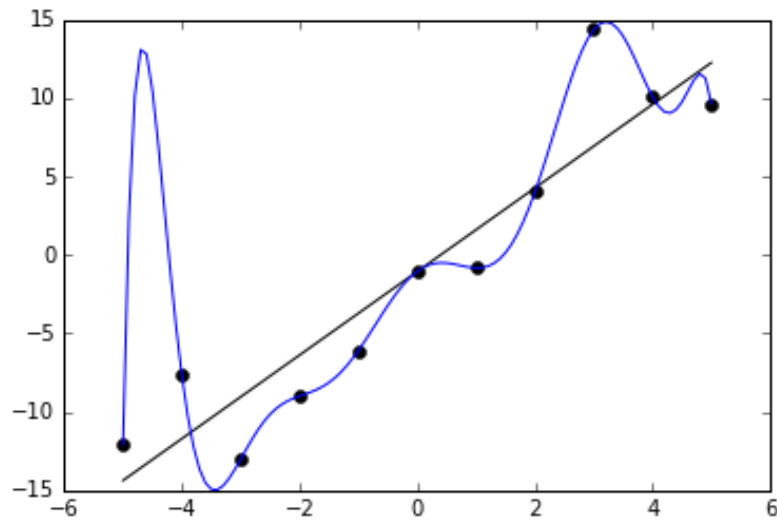


Figure 2.5: 過擬合 (overfitting)³。藍線代表的是過擬合模型，雖然藍線完美符合圖中的資料點，但在預測新資料時，黑線會有較低的誤差

2.1.5 激勵函數

對於模型的輸入 x ，第一層的權重為 W_1 ，則 x 經過第一層後得到的輸出 y_1 為：

$$y_1 = W_1 x \quad (2.1)$$

接著，第一層的輸出 y_1 會做為第二層的輸入，同樣地，對於權重為 W_2 的第二層， y_1 經過第二層後的輸出為：

$$y_2 = W_2 y_1 \quad (2.2)$$

根據 (2.1) 和 (2.2)，可以將第二層的輸出 y_2 改寫，並得到以下式子：

$$\begin{aligned} y_2 &= W_2 y_1 \\ &= W_2 W_1 x \end{aligned} \quad (2.3)$$

³ 圖片來源：<https://en.wikipedia.org/wiki/Overfitting>，圖片授權採用創用 CC 姓名標示 - 相同方式分享 4.0，圖片作者為 Ghiles



在一般情況下，模型的輸入 x 通常為一個擁有 n 個元素的向量，而每一層的權重 W_i 則是大小為 $m \times n$ 的矩陣，其中 m 表示該層所擁有的節點數，每個節點對應到輸入 x 的 n 個元素各自有設定不同的權重。根據矩陣的結合律 (associative laws)，(2.3) 可以再被改寫為以下式子：

$$\begin{aligned} y_2 &= W_2 W_1 x \\ &= (W_2 W_1) x \\ &= W' x \end{aligned} \tag{2.4}$$

在 (2.4) 中，第一層的權重 W_1 和第二層的權重 W_2 被合併成一個矩陣 W' ，這代表著若是依照上述情形設計一個深度學習的模型，不論模型架構再怎麼深，所有層的權重皆可以合併為一個矩陣，等同於輸入 x 只經過了一層的模型後便輸出結果，而為了避免發生上述的情況，模型的層與層之間通常會插入一個非線性函數 (nonlinear function) 去改變一層的輸出值，在類神經網路領域中，這些函數被稱為激勵函數 (activation function)。

常見的激勵函數有 Sigmoid、Hyperbolic tangent、以及 ReLU 等 (Figure 2.6)，如前面所述，它們皆為非線性函數，下一節將會提到的 Softmax 也是一種激勵函數，但 Softmax 主要用於最後的輸出層而非插入於層與層之間。

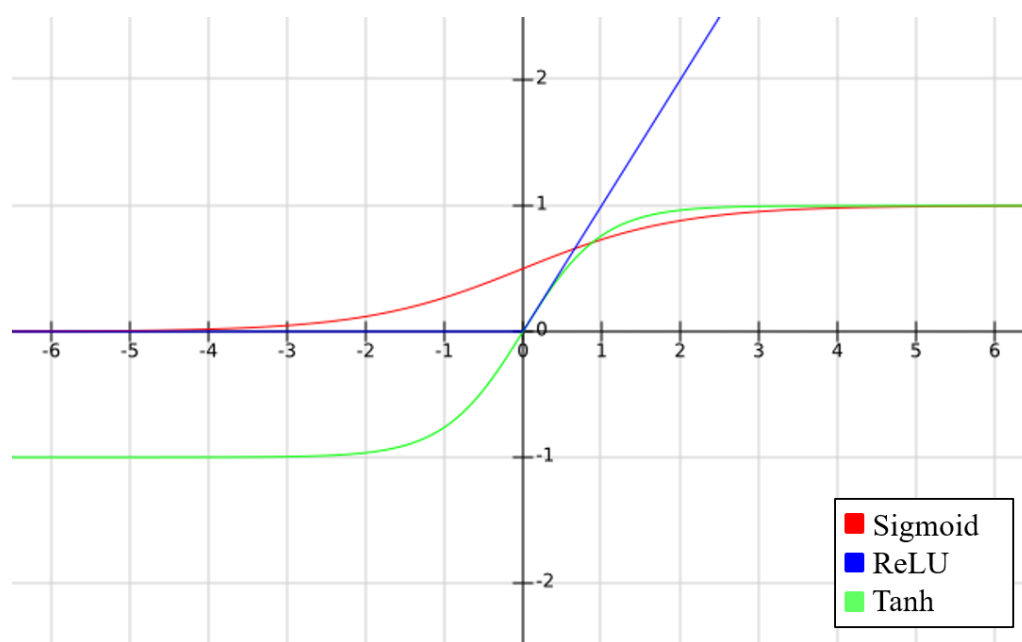


Figure 2.6: 激勵函數與其對應的輸出曲線

2.1.6 Softmax 層

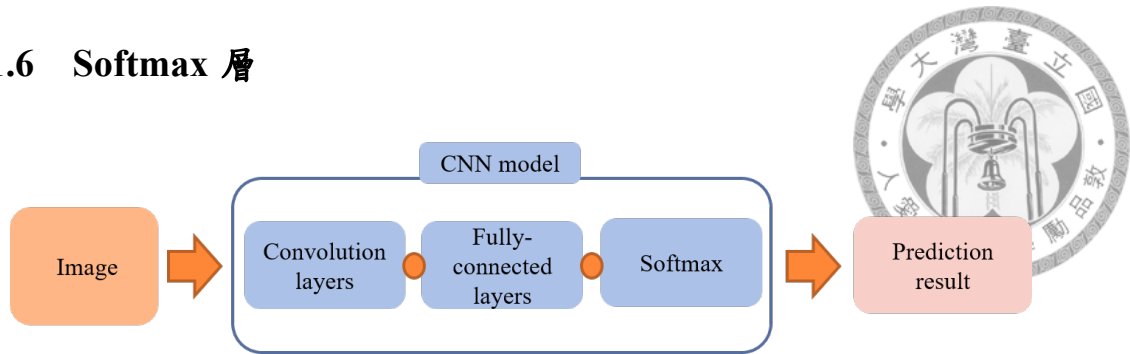


Figure 2.7: CNN 的辨識流程圖

CNN 的辨識過程（如 Figure 2.7）中，在經過卷積層及全連接層後，通常會再經過 Softmax 層，才會輸出最後的結果。Softmax 函數的數學式如 (2.5) 所示：

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}, j = 1, \dots, K \quad (2.5)$$

這樣的用意在於該函數會將輸入值壓縮在 $(0, 1)$ 之間，並且讓所有元素的總和為 1，因此在全連接層經過 Softmax 層之後，每一項輸出值可以看作是物件被分類到該類別的機率大小，一般情況下將會以其中最大值的對應類別做為分類的結果。

2.2 AlexNet

CNN 的模型架構與最後預測的準確率息息相關，而自 CNN 開始發展以來，如何建構出好的模型架構一直是一項重要的課題。2012 年，Alex Krizhevsky 等人首次以 CNN 參加 ILSVRC [1] 並且得到了冠軍，他們提出的 AlexNet [2] 在 ILSVRC-2012 資料集 [1,3] 擁有 top-5 錯誤率⁴16.4% 的成績，大幅超越當年其他參賽者以及去年冠軍的 25.8%。

做為早期 CNN 代表的 AlexNet，其架構並不複雜，僅由 5 層卷積層、2 層全連接層加上輸入層以及池化層組成，主要原因是受限於當時的硬體規格影響，只能建構出以現在眼光看來深度較為不足的架構，但 Alex Krizhevsky 等人在文章 [2] 中提出了許多影響 CNN 架構深遠的概念，像是使用 ReLU 激勵函數取代之前常用

⁴Top-5 錯誤率 (Top-5 error)：係指在預測類別時輸出最有可能的 5 項類別，且實際類別不在這 5 個類別的資料所佔的比率

的 sigmoid 和 tanh、資料增強 (data augmentation)、最大池化層等等，其中最關鍵的為加深網路架構有助於辨識率的上升。

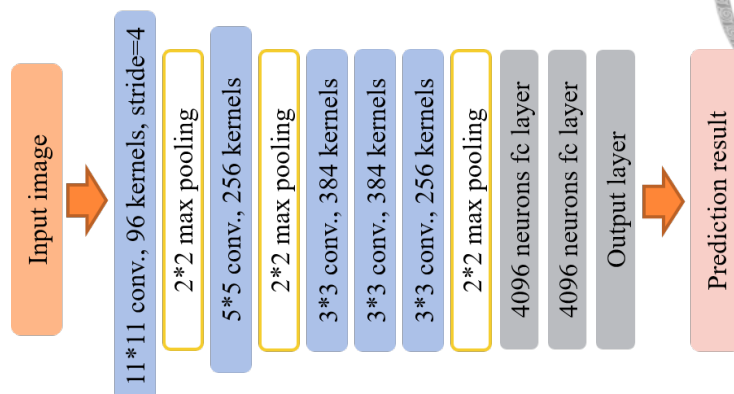


Figure 2.8: AlexNet 架構圖⁵

2.3 VGG-16 模型

另一個知名模型架構是由 Karen Simonyan 等人在 2014 年提出 [5]，特徵是以較小的卷積核以及較深的架構為特點，俗稱 VGG 模型 (源自於 Karen 等人所屬的 Visual Geometry Group)，其表現最好的模型在 ILSVRC-2012 資料集有著 top-5 錯誤率 6.8% 的成績，在卷積神經網路領域中極具代表性。

在 [5] 中，Karen 等人提出了若將卷積核縮小，便可以加深模型的深度，其中提到的一個模型架構的卷積層與全連接層共有 16 層，再加上額外的池化層所組成，俗稱 VGG-16 模型 (Figure 2.9)，該模型的卷積核大小皆為 3×3，而池化層的核為 2×2，隨著深度提升，卷積層所使用的卷積核數量也跟著提升，最後經由 2 層節點數為 4096 的全連接層以及輸出層進行分類。該模型在 ILSVRC-2012 資料集的表現為 top-5 錯誤率 7.2%。

2.4 未知類別的辨識

未知類別的辨識，原文為 Unseen Category Query Identification [8]，簡稱 UCQI，由 Siang Thye Hang 等人提出，用來推斷輸入圖片的物件類別是否存在於模型訓練

⁵跨步 (stride)：係指兩個卷積運算的中心點相差多少個像素點的值。以 stride=4 來說，以座標 (0,0) 為中心執行第一次卷積，則第二次將會以 (4,0) 為中心點。若無標示則 stride 設定為 1。另外，除非特別註記，一般來說橫向跨步與縱向跨步會設定為同樣的值



Figure 2.9: VGG-16 模型架構圖

時所給定的種類內。在 [8] 中提出了一個情境：Figure 2.10 代表著 3 個不同的輸入 α 、 β 、 γ 分別進入 4 個類別 (Class 0 ~ Class 3) 的分類模型後，該模型全連接層的輸出結果。第 2.1.3 節中提到，全連接層的輸出反映了物件特徵與該節點所對應類別的相關性，因此可以將該層的輸出視為對各類別的分數 (score)，分數愈高則圖片中的物件與該類別愈相似，而理想的情況就如同 α 的輸出結果：其中一類的分數特別突出且為正數，代表 α 與第 0 類的物件相似度很高；但實際上也可能存在如同 β 的輸出結果，雖然有分數為正值的類別，值本身的大小卻很低，抑或是如同 γ 的輸出，所有的類別分數均為負值，但是在一般情況下，即使輸出結果如同上述 β 或 γ 的輸出，模型最後依然會將分數最大者做為最後的預測結果，以 β 和 γ 這兩個例子來說，模型將預測 β 屬於第 2 類的類別， γ 則被分類至第 0 類，而從輸出的值等於分數 (相似度) 的觀點來看，這兩者的分類有很大機率會是錯誤的。

之所以提出這個情境的原因在於，原論文所使用的 PlantCLEF 2016 資料庫 [9] 在測試資料中包含了未知類別 (unseen category) 的圖片，這些圖片並不屬於一開始訓練模型時所給定的類別，因此如 Figure 2.10 中 β 、 γ 的情況是有可能發生的，輸入的圖片有可能皆不相似於模型辨識的類別，若能找出這些未知類別的圖片並排除在測試過程外，抑或是說，將這些圖片歸類在一個額外的類別，皆可以提升最後模型辨識的準確率。

如第 2.1.6 節所述，全連接層的輸出會再通過 Softmax 層才得到最後的結果，但在這項方法中，利用的並非 Softmax 層的輸出，而是全連接層的輸出，而 UCQI 的辨識方法便是利用全連接層輸出的分數與相似度成正相關的關係，找出每個類別之中最小的分數做為門檻值 (threshold)。詳細的步驟如下：

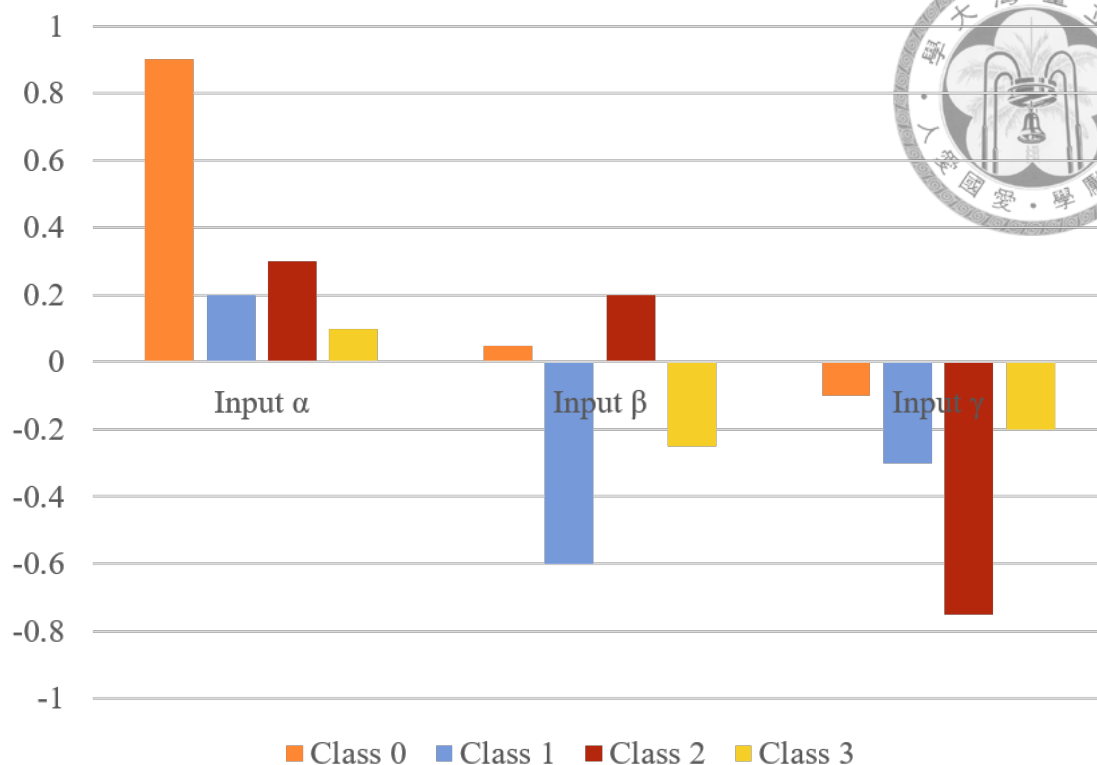


Figure 2.10: 假設情境：模型辨識三個輸入 α 、 β 、 γ 產生的結果

(1) 假設將 M 張訓練資料分類成 N 類，全連接層會輸出 $M \times N$ 的分數矩陣 $S_{M \times N}$

(2) 從分數矩陣中將分類錯誤的資料結果移除（因為是基於訓練資料做處理，因此可以得知該圖片的正確類別），得到新的分數矩陣 $S_{M' \times N}$ ($M' \leq M$)

(3) 找出各類別的門檻值 $t = [t_1, \dots, t_N]$ ，其中 t_i 代表第 i 類別的門檻值，定義如 (2.6)：

$$t_i = \min_{k \in [1, M']} S_{k,i}, i = 1, \dots, N \quad (2.6)$$

在測試階段，若測試資料所有類別的分數皆小於該類別對應的門檻值，則會將這項資料視為未知的類別。

由於本次實驗中所使用的資料庫測試資料與 PlantCLEF 2016 資料庫不同，並不包含未知類別的圖片，因此將會對此方法稍作修改，用來選出特徵較不明顯、與同一類的訓練圖片相比，相似度較低的圖片，並對這些圖片進行後處理以及再辨識。



2.5 圖像分割

在電腦視覺領域中，圖像分割係指將圖像細分成不同部分的程序，譬如將一張圖片的前景與背景分離出來即是屬於圖像分割的範疇。圖像分割主要的用意是改變圖片的表現方式，使得該圖片更容易分析，一般的作法是給予圖片每一個像素點各自的標籤，而同樣標籤的點屬於同一類，代表在圖片上這些點呈現的是同一物件或區域，藉由這樣的方法來強調或找出圖片的重點以利分析。

目前有許多能夠實現圖像分割的方法 [10]：分析灰階像素值，將圖片二值化的大津演算法 [11]、基於邊緣偵測的圖像分割 [12]、以及利用群聚分析 (clustering analysis) 對圖片的像素進行分群等。在第 2.5.1 節中將會介紹群聚分析的其中一種方法：K-平均分群法 (K-means clustering)。

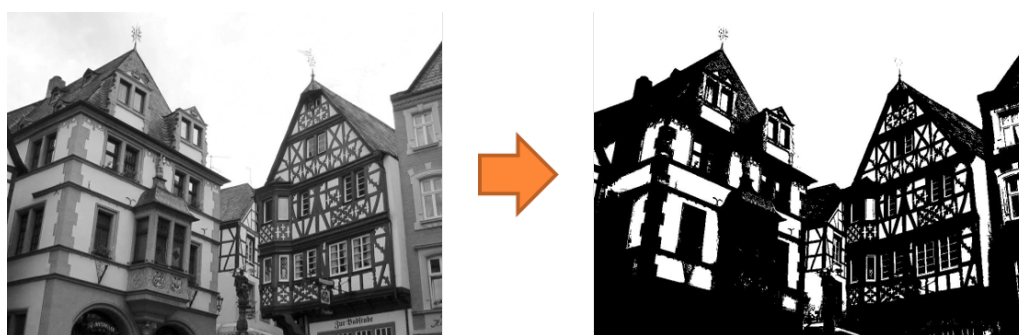


Figure 2.11: 利用大津演算法進行二值化圖像分割⁶

2.5.1 K-平均分群法

K-平均分群法 [13] 會將資料點歸類成 k 個分群， k 的數值必須事先決定好，而每個資料點與該資料點所屬分群中心的距離為最小 (與其他分群中心比較)，詳細的步驟如下：

(1) 從所有的資料點中隨機選出 k 點，做為各個分群的中心

(2) 對每個資料點計算該點到 k 個中心的距離，找出距離最小的中心，並將資料點歸類為該中心的分群。這裡的距離採用歐幾里得距離 (Euclidean distance, 又稱歐氏距離)，對於兩點 $x = (x_1, \dots, x_n)$ 和 $y = (y_1, \dots, y_n)$ 的歐幾里得距離定義

⁶圖片來源：https://en.wikipedia.org/wiki/Otsu%27s_method，圖片授權皆採用創用 CC 姓名標示 - 相同方式分享 1.0，左圖作者為 <http://www.freepotos.lu/>，右圖作者為 Pikez33



如 (2.7) 所示：

$$\begin{aligned}
 d(x, y) &= d(y, x) \\
 &= \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2} \\
 &= \sqrt{\sum_{i=1}^n (x_i - y_i)^2}
 \end{aligned} \tag{2.7}$$

(3) 根據 (2.8) 計算每個分群的平均，做為新的分群中心。其中 $S_i^{(t)}$ 代表第 i 分群的資料點所形成的集合。

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \tag{2.8}$$

(4) 重複步驟 (2)(3) 直到分群中心幾乎不再變動，或是已執行規定的迭代次數為止。

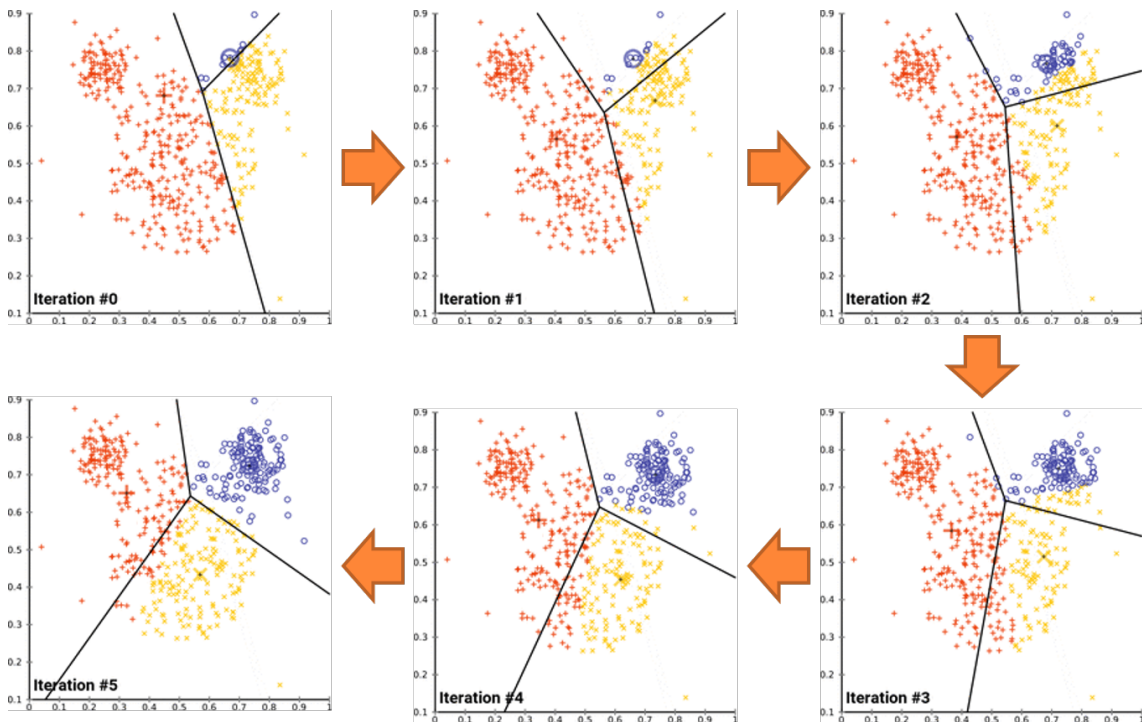


Figure 2.12: K-平均分群法。⁸每次迭代的最後找出各分群的中心做為下一次迭代的中心，不斷更新以找到最佳的分群

⁸ 圖片來源：https://en.wikipedia.org/wiki/K-means_clustering，圖片授權皆採用創用 CC 姓名標示一相同方式分享 4.0，原圖片作者為 Chire

根據以上步驟，最後所有資料點都會被歸類到一個最適合的分群中，這與圖像分割將屬於同一物件的像素點歸類為同一類，並與其他物件的像素點有所分別的概念很類似，若將能夠代表數位圖片的資訊（例如：每個像素點的 RGB 值、像素點的座標等）做為資料點的資料進行 K-平均分群法，便可以利用該演算法實現圖像分割，將屬於同樣物件的像素點歸類為同一類（Figure 2.13）。

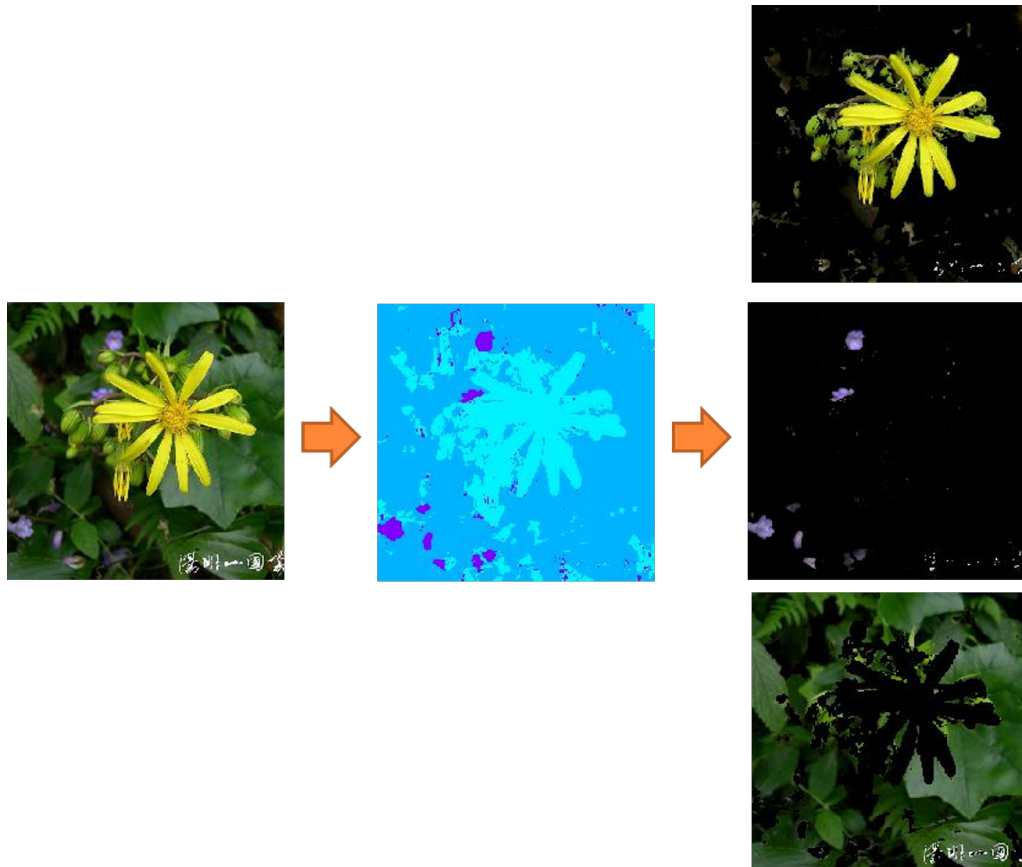


Figure 2.13: 以 K-平均分群法進行圖像分割。左圖為原圖；中圖為歸類後的結果，同樣顏色代表屬於同一分群；右圖則是依照歸類結果所分割出的三張圖片



Chapter 3

實驗設定

3.1 資料集

本次研究使用的資料集包含約 2400 種類的植物，圖片總量約有 200000 張，來源皆為網路搜尋。在同一種植物類別的資料集中，由於可能包含該類植物的不同部位，因此圖片之間會有很大的差異性，以 Figure 3.1 為例子，此為其中一種植物：爵床 (*Justicia procumbens*) 的資料集所包含的 3 張圖片，可以發現即使這些圖片是屬於同一類，其中所包含的植物部位，抑或是說，包含的特徵也有很大的差異。



Figure 3.1: 同屬於植物：爵床的 3 張圖片

Figure 3.2 為 2400 類中圖片數量的分布圖，橫軸為圖片數量的區間，縱軸為類別的數量，長條圖所表示的是代表圖片數量落入該區間的類別有多少；從 Figure 3.2 來看，大部分的植物類別圖片數量落在 $[0, 45]$ 之間，其次為 $(45, 90]$ 之間，有鑑於對於卷積神經網路來說，資料量的多寡與最後模型的效能有很重大的



關聯性，為避免因為資料的不足造成模型辨識率的下降，在此選出資料量最多的 500 類別，以這 500 類別所形成的子資料集做為本次實驗所用，為求方便，在之後的文章將會以 top-500 資料集來稱呼這個子資料集。

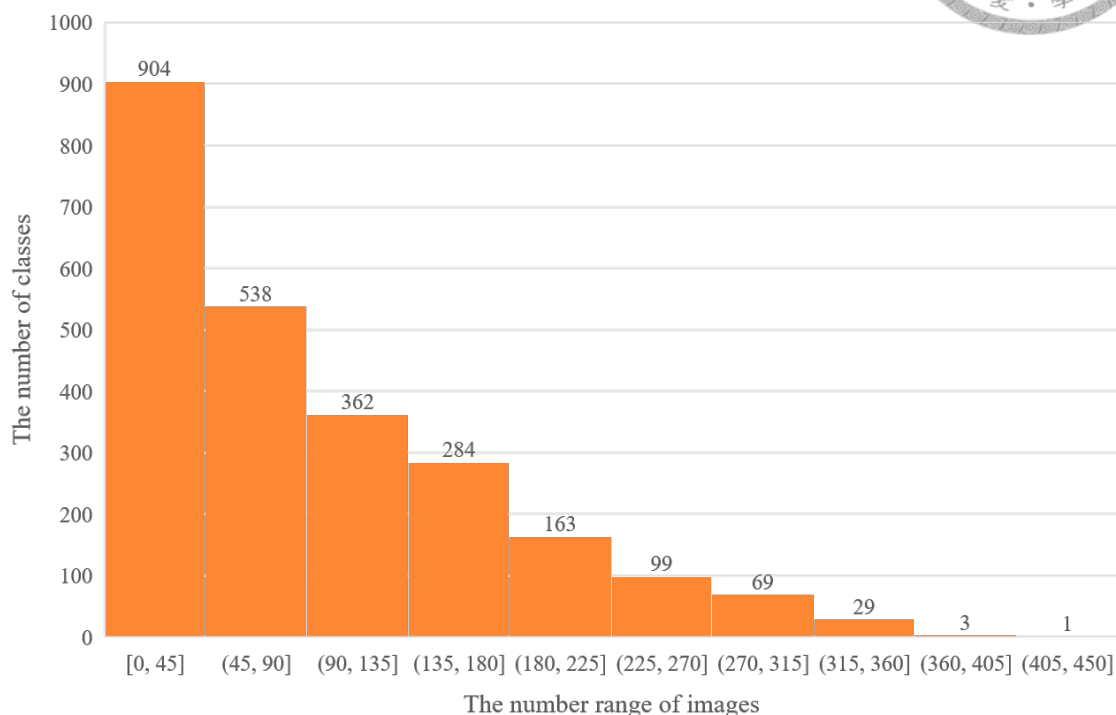


Figure 3.2: 原資料集的圖片數量分布圖

Figure 3.3 為 top-500 資料集的圖片數量分布圖，其中圖片量最少的類別所含有的數量為 158 張，最多的則有 450 張，而圖片數量介於 [158, 187] 之間的類別數量最多，共有 172 類。Top-500 資料集包含了 110934 張圖片，在此從每個類別中隨機選取 90% 做為訓練資料，剩餘的 10% 做為測試資料，處理過後的結果為：訓練資料的圖片數量總和 99673 張，測試資料則為 11261 張。

3.2 模型參數設定

模型選擇

本次實驗主要使用的卷積神經網路模型為 VGG-16 模型，另外實驗初期也使用過 AlexNet，兩者的模型架構已於第 2.2 節與第 2.3 節說明過，在此不再贅述。以下所說明的參數設定皆以 VGG-16 的設定為主，有關於 AlexNet 的參數將在第 4 章

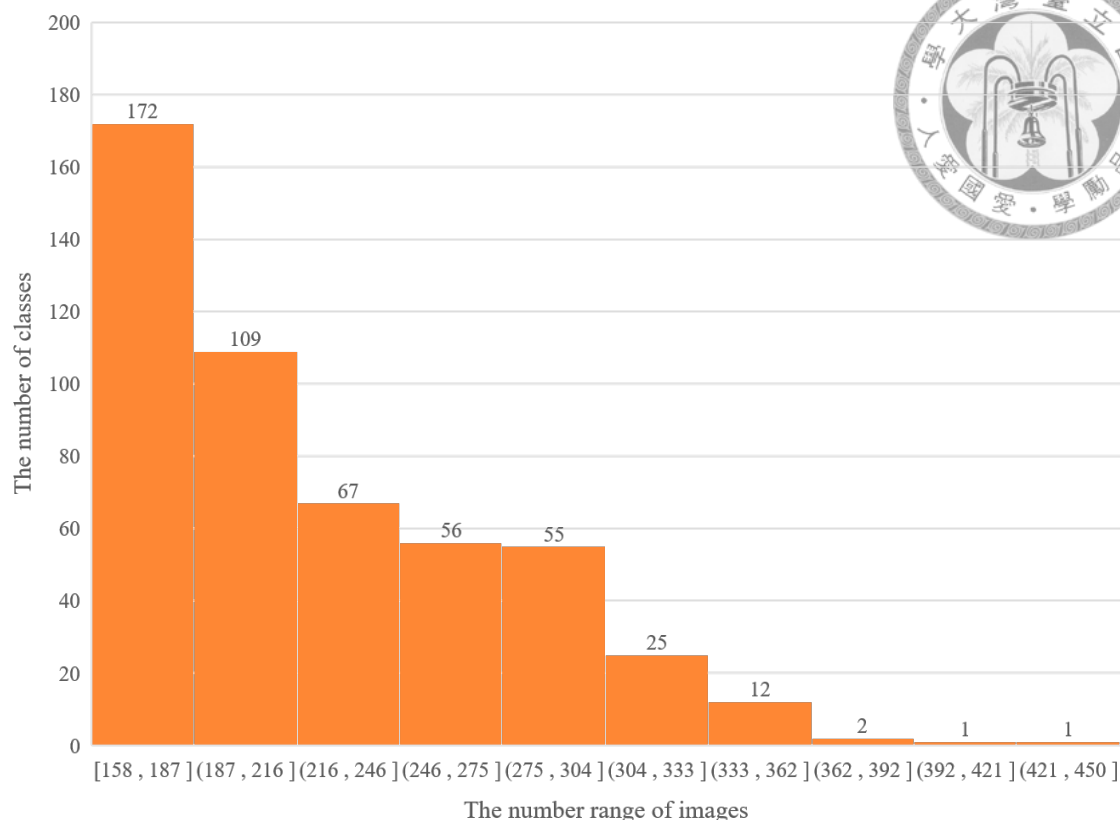


Figure 3.3: Top-500 資料集的图片數量分布圖

實驗結果比較時再做說明。

激勵函數

激勵函數的設定與模型的原出處 [5] 相同，對每一層卷積層以及全連接層的輸出使用 ReLU (Rectified Linear Unit) 函數，定義如 (3.1)，如同 Figure 2.6 中的藍線所示，ReLU 會將負的輸入值變為 0，而保留正的輸入值：

$$ReLU(x) = \max(0, x) \quad (3.1)$$

權重初始化

對於卷積層以及全連接層的權重，利用遷移式學習 (transfer learning) [6]，預先載入使用 ILSVRC-2012 資料集訓練出的權重，再利用本次實驗的 top-500 資料集進行權重的微調 (finetune) 以讓模型符合本研究的需求。使用遷移式學習的好處在於，一來載入已經訓練好的權重可以減少訓練時間，與從零開始的訓練比

較，模型只需要微調這些權重即可；二來這些預載的權重是使用較大量的資料訓練出來，先前介紹資料集（第 3.1 節）時提到，卷積神經網路的辨識率與訓練資料的多寡有著很緊密的關聯性，對於以 ILSVRC-2012 資料集－訓練資料數量約 120 萬所訓練出的模型，其效能是可以期待的。由於預先載入的 VGG-16 模型是用於 ILSVRC-2012 資料集的辨識，在最後的預測時會將圖片預測為 1000 類中的其中一類，為了符合本實驗分類成 500 類的需求，在模型的後面會再加上一層 500 個節點的全連接層做為輸出層（Figure 3.4），而該層的權重初始化則是使用 Xavier uniform initialization [14]（又稱 Glorot uniform initialization），其初始化的權重會形成 $[-x, x]$ 的均勻分布（uniform distribution）， x 的求法如 (3.2)，其中 N_{in} 與 N_{out} 分別代表該層輸入與輸出的資料數量：

$$x = \sqrt{\frac{6}{(N_{in} + N_{out})}} \quad (3.2)$$

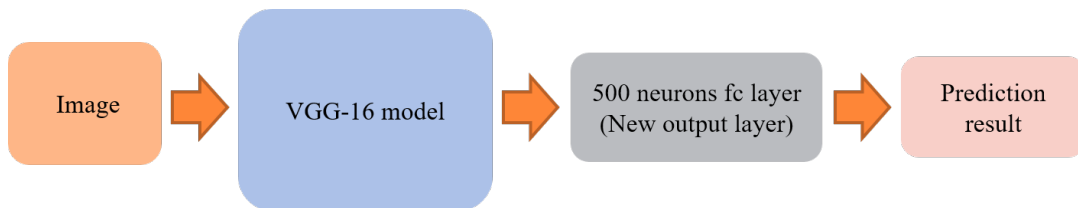


Figure 3.4: 在 VGG-16 模型後加入新的輸出層

優化器

模型的優化器（optimizer）使用 Stochastic Gradient Descent [15,16] with momentum [17]，動量（momentum）設為 0.9，並且使用 Nesterov Momentum [18]。

學習速率

學習速率會隨著訓練的過程而降低，具體的變化曲線如 Figure 3.5，數學式則為 (3.3) 所示，其中 $R_{initial}$ 代表起始的學習速率（learning rate），設定為 0.001， R_{final} 則是學習速率降低到最後的最小值，設定為 0.00001， $global_step$ 代表目前為訓練過程中的第幾個 epoch¹， $decay_step$ 表示在訓練到第幾個 epoch 時將學習

¹Epoch：訓練一個 epoch 代表所有訓練資料皆經過一次訓練過程，因此實際上的迭代（iteration）次數為 $epoch_num \times \lceil data_num / batch_size \rceil$

速率收斂到 R_{final} ，在此設定為 100。

$$R_{decayed} = (R_{initial} - R_{final}) \times \left(1 - \frac{global_step}{decay_step}\right) + R_{final} \quad (3.3)$$

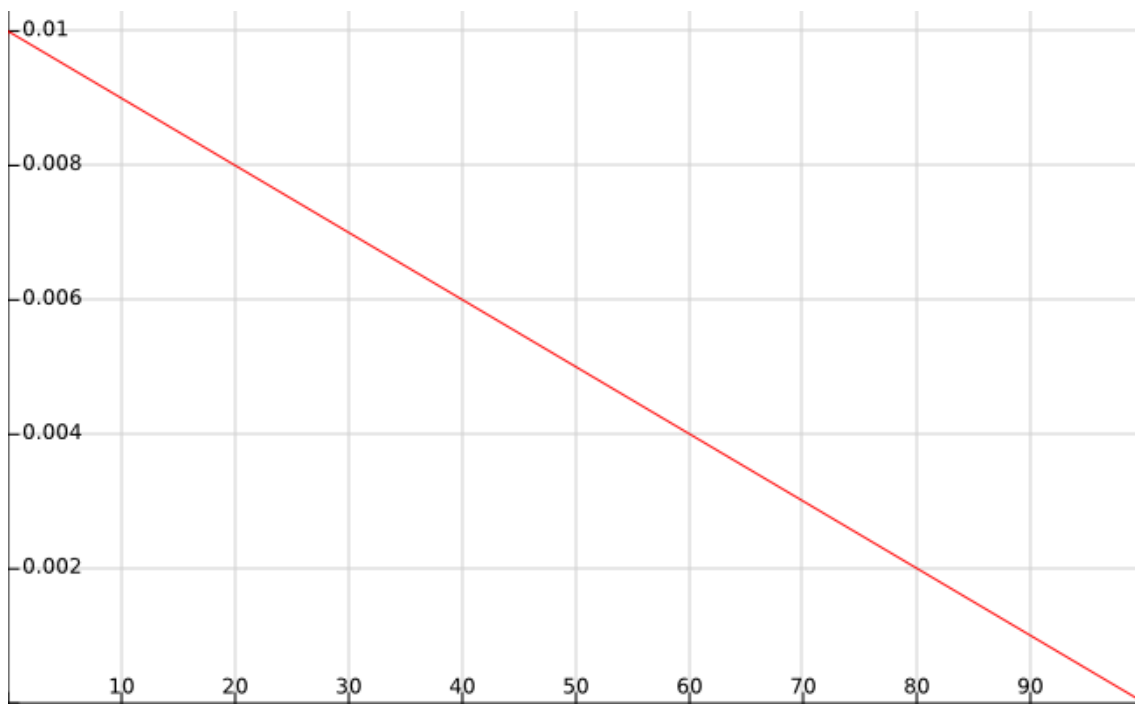


Figure 3.5: 學習速率的變化圖。x 軸為目前的 epoch 數，y 軸表示對應的學習速率

其它設定

Dropout 設定為 50%；損失函數為交叉熵（cross entropy）；批次（batch）大小則為 32；整個訓練過程固定為訓練 100 個 epoch。

3.3 圖片前處理

圖片前處理（image pre-processing）係指在模型訓練之前對圖片進行處理的過程，通常是用以增加資料量或是避免模型因輸出值差距過大造成訓練上可能出現的問題等等。在本次實驗中，會先將每一張圖片的 RGB 值除以 255，使得像素值介於 $[0, 1]$ 之間，再經過後述的資料增強（data augmentation），對訓練圖片進行幾何轉換，藉此增加訓練資料的數量。



3.3.1 資料增強

資料增強是訓練卷積神經網路時經常使用，用來增加訓練資料量的過程之統稱，其主要原理是改變圖片特徵的位置，同時保留特徵與特徵之間的相對關係，藉此增加資料量以避免模型調整權重時過於偏向訓練資料造成的過擬合 (overfitting)，也令模型學習到在不同位置的相同特徵，淡化特徵與圖片中絕對位置的關係。以下將介紹本次實驗所使用的資料增強方法：

水平翻轉

圖片將有 50% 的機率進行水平翻轉。

旋轉角度

圖片將隨機旋轉 0° 、 90° 、 180° 、 270° 之中其中一個角度。

多重尺度訓練

多重尺度訓練 (multi-scale training)，係指不將訓練圖片的大小固定之訓練方法。一般的訓練流程中會事先對圖片進行縮放以符合模型事先規定的輸入圖片大小，但在多重尺度訓練中，訓練圖片的大小並不唯一，而是從一段區間中隨機選擇並縮放至該大小，在 VGG 模型的出處 [5] 中也採用了這種作法，將圖片大小訂定在 $[256, 512]$ 之間。本次實驗中會隨機將圖片縮放至 224×224 、 320×320 、 410×410 、 500×500 的其中一個大小 (Figure 3.6)。

隨機裁剪

承前段所述，模型的輸入圖片大小為事先決定好的定值，若是經過縮放使得圖片大小與模型要求不符，則需要從縮放後的圖片中裁剪出一部份以符合模型需求 (Figure 3.7)，這個步驟稱為隨機裁剪 (random crop)。配合前述的多重尺度訓練，在圖片縮放為較小的尺寸時，實際輸入至模型的圖片佔有原圖片較大的比重，模型可以學習到較為全面的特徵；另一方面，當圖片尺寸放大時，因為進行隨機裁剪的緣故，輸入至模型的圖片只會是原圖片的一部份，模型只會接收到圖片部分的資訊。綜上所述，利用多重尺度訓練以及隨機裁切，每次訓練圖片進入模型

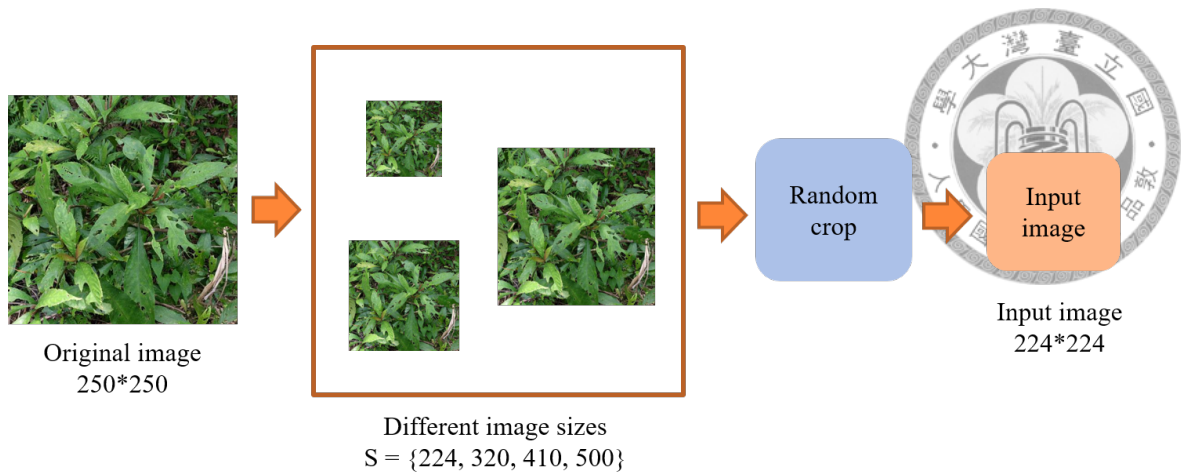


Figure 3.6: 多重尺度訓練

時，給予模型的資訊皆不相同，如此一來可以令模型同時兼顧整體以及局部的特徵資料。

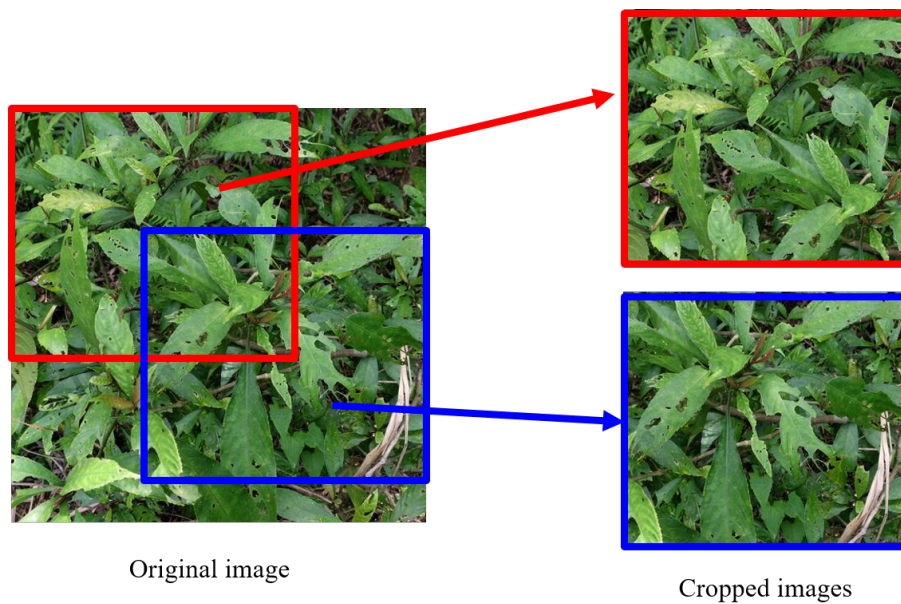


Figure 3.7: 隨機裁剪

3.4 UCQI 的修改

在第 2.4 節中提到，UCQI 是用於排除不在訓練時給定類別的資料，藉此降低因為將這些資料歸類於特定類別而造成的錯誤率，但在本次實驗所使用的 top-500 資料集中並不會出現資料不在給定類別的情況，因此實作時將會對原本的演算法



進行更動以符合本次實驗的使用情境。

原本的 UCQI 是比較個別所有訓練資料的類別分數，選出最低者當作該類別的門檻值，而在測試階段中，輸出的分數會與各自的門檻值做比較，若是所有分數皆小於對應的門檻值，便將該測試資料視為未知的類別。本次實驗中，為了選出辨識度不足或是特徵較不明顯的圖片，將會對門檻值的選出以及測試階段的比較這兩個部分做修改。

門檻值的選出方法

在第 2.4 節的 (2.6) 用數學式表示了門檻值原本的選出方法，原本的方法會比較所有訓練資料的分數來找出門檻值，連同不是屬於該類別的資料也會一起比較，為了選出辨識度不足的圖片，在此將門檻值的選出方式修改為以下式子 (3.4)，其中 I 代表所有的訓練資料之集合， C_i 則為屬於第 i 類的訓練資料所形成之子集合：

$$t_i = \min_{k \in [1, M'], I_k \in C_i} S_{k,i}, i = 1, \dots, N \quad (3.4)$$

與 (2.6) 比較，(3.4) 加入了 $I_k \in C_i$ 這一條件，差別在於選出門檻值的過程中只會比較屬於第 i 類別的訓練資料而非全部，如此修改的用意在於利用已知正確類別的訓練資料，找出最低的分數做為低標，之後的測試資料必須不低於這個低標，代表「至少有如此的相關性」，最後才會被辨識為該類別。為了達到這項目的，還需要修改測試階段時的門檻值比較方式，具體如後述。

測試階段的比較

在修改過後的測試階段中，測試資料會經過與一般 CNN 相同的辨識過程，並得到預測的結果，在這之外還會利用測試資料的分數值做額外的比較，其全連接層的輸出不經過 softmax 層，會直接找出之中的最大值以及對應的類別，再將最大值與對應類別的門檻值做比較，比較結果必須是不低於門檻值，才會確定測試資料歸屬在該類別，否則的話則會將該資料的辨識結果保留，經過後述的圖片處理後進行再辨識。

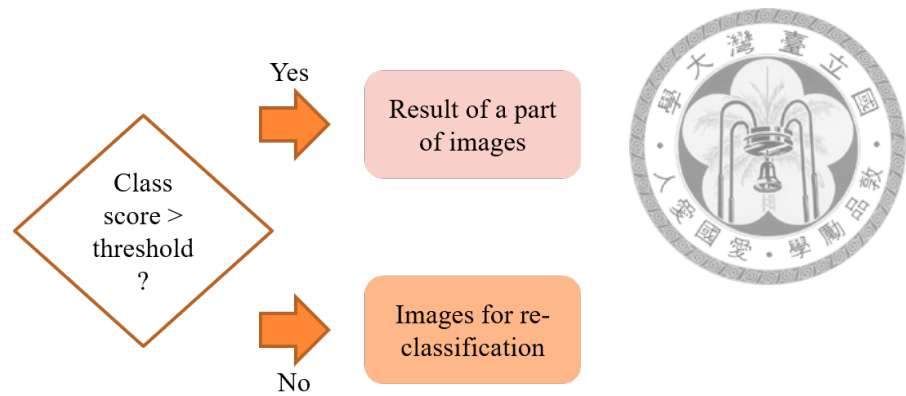


Figure 3.8: 與門檻值比較後決定是否採用測試資料的輸出做為預測結果

3.5 圖片後處理

在測試階段中，一部份的圖片經過第 3.4 節所述的過程後被選取出來，這些圖片雖然可以被歸類到特定的類別，但全連接層的輸出值比先前找出的，該圖片對應類別的門檻值還要低，因此推斷圖片本身的辨識度不足，模型的辨識結果很有可能是錯誤的。發生以上情況的原因可能在於圖片本身太過複雜，植物的主體與背景或其他物件參雜在一起，導致模型在擷取特徵時混入太多雜訊，無法得到正確的資訊，抑或者是植物的主體僅占圖片的一小部分，造成雖然辨識結果是正確的，但就整張圖片來看資訊量太少，因此被選出。這些辨識度不足的圖片會利用第 2.5 節提到的圖像分割，嘗試將植物主體與背景及其他物件分離，並選出最能代表植物主體的圖片進入模型進行再辨識，得到較好的結果。具體的相關設定與步驟將會在以下說明。

圖像分割

第 2.5.1 節中曾經介紹過 K-平均分群法，該演算法是用於將資料點分成特定數量的分群。在本實驗中將會利用 K-平均分群法來進行圖像分割，關於演算法所需要的資料部分，有別於一般所使用的 RGB 色彩空間 (color space)，研究 [19] 顯示 CIELAB 色彩空間可以更好的結果，因此在本實驗中將會先將圖片從 RGB 色彩空間轉換至 CIELAB 色彩空間，再利用其中代表顏色的 a* 值與 b* 值 (L* 值代表亮度) 做為資料進行運算。細部參數的設定則為：歸類成 3 個分群、迭代次數為 5 次。



選擇圖片

經過圖像分割的程序後，測試圖片會被分割成3張圖片，用於再辨識的圖片即是從中選出。選擇的步驟如下：

- (1) 各別計算3張圖片的質心位置
- (2) 各別計算3張圖片的質心到圖片中央的距離
- (3) 選擇距離最小者做為再辨識的圖片

在一般情況下，植物的主體通常會較為接近圖片的中心，以上的選擇步驟便是考慮這點，利用圖像分割將植物主體與背景分離後，計算質心位置並比較質心與圖片中央的距離，以距離做為依據選擇出最能代表植物主體的圖片。以Figure 3.9為例，左上圖為原圖，經過圖像分割的結果為下半部的3張圖片，中上圖則是分群之後的結果，同樣顏色代表屬於同一分群，在右上圖中標示出了3張分割後的圖片各自的質心，3種顏色與中上圖所使用的相同，代表的分群也一致。以質心到圖片中央的距離為基準，則會選擇中下圖做為再辨識的圖片。

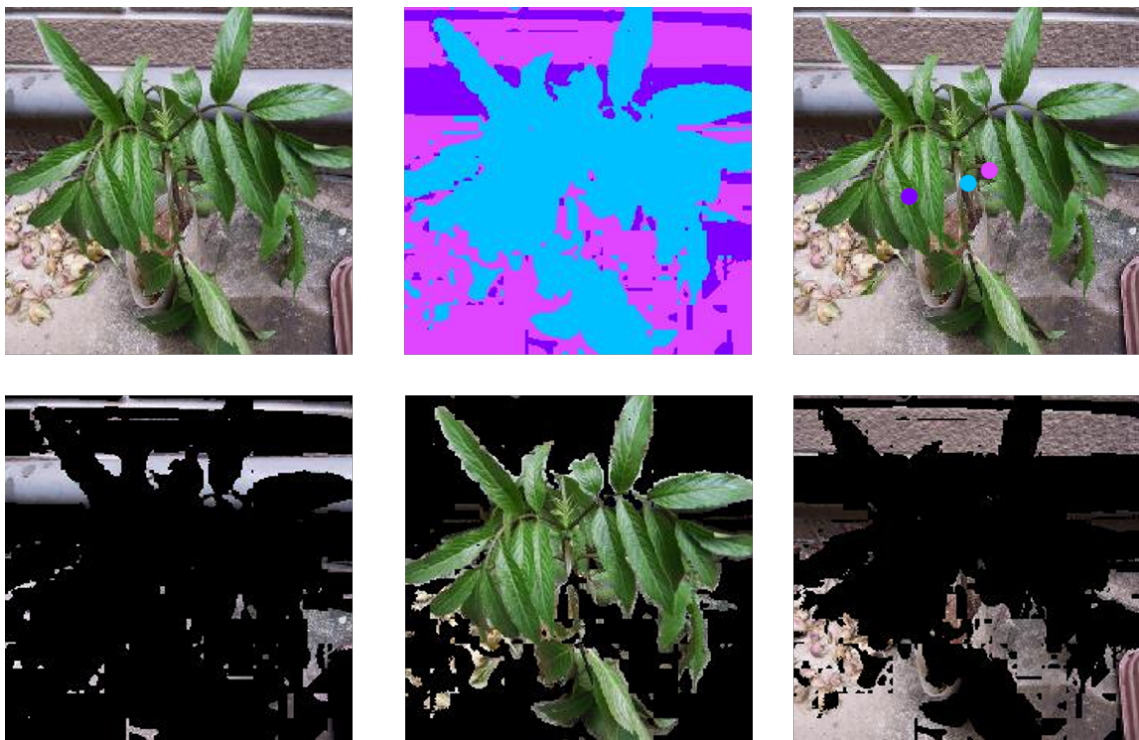


Figure 3.9: 圖片進行圖像分割的結果與質心位置。由左至右，由上到下分別為：原圖、歸類結果、質心位置圖、分割圖片1、分割圖片2、分割圖片3

3.6 實驗環境

本次實驗使用的機器學習框架為 TensorFlow 1.5.0 [20,21]，該框架是由 Google 開發，根據官方網站敘述，TensorFlow 已被許多商業公司採用，並用於實務應用上。TensorFlow 支援 Python、C++ 等程式語言，在本次實驗中使用的語言為 Python 3.6.3。TensorFlow 亦支援 GPU 加速訓練過程，實際應用上，本次實驗採用一張 Nvidia GeForce GTX 1080 Ti 進行訓練，在 GPU 的加速下，每一個 epoch 需要花費約 20 分鐘，在第 3.2 節當中提到，模型設定為 100 個 epoch 後結束訓練，整個訓練過程合計約花費 33 小時。上述提及的實驗皆於 Ubuntu 16.04 作業系統下進行，其它硬體設備如 CPU 型號為 Intel Xeon CPU E5-2630 v4，記憶體 (DRAM) 大小則為 128GB。





Chapter 4

實驗結果與分析

4.1 模型架構與訓練方法

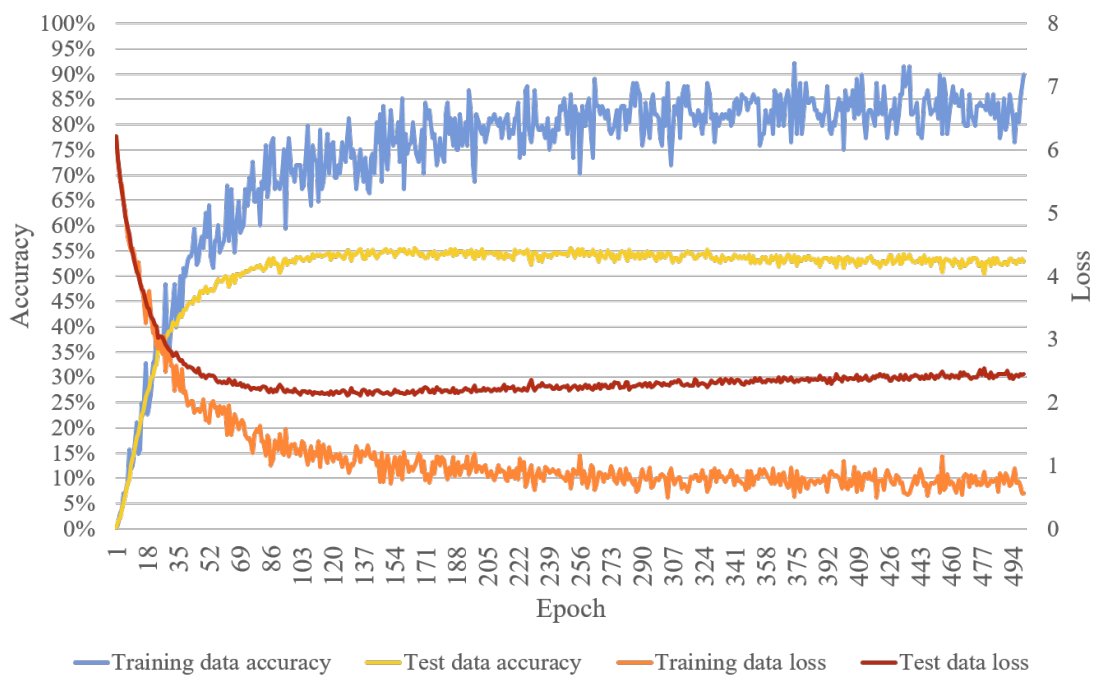


Figure 4.1: AlexNet 訓練過程記錄

Figure 4.1為 AlexNet 的訓練過程記錄圖。在本次實驗中將會比較不同模型架構之間的效能。關於 AlexNet 的模型參數設定與第 3.2節所述的不同處如下：



權重初始化

本次實驗中 AlexNet 並沒有使用遷移式學習，因此網路中的所有權重皆是以 Xavier uniform initialization 進行初始化。

學習速率

初始速率設定為 0.01，整個過程學習速率的變化如 Figure 4.2，數學式如 (4.1)：

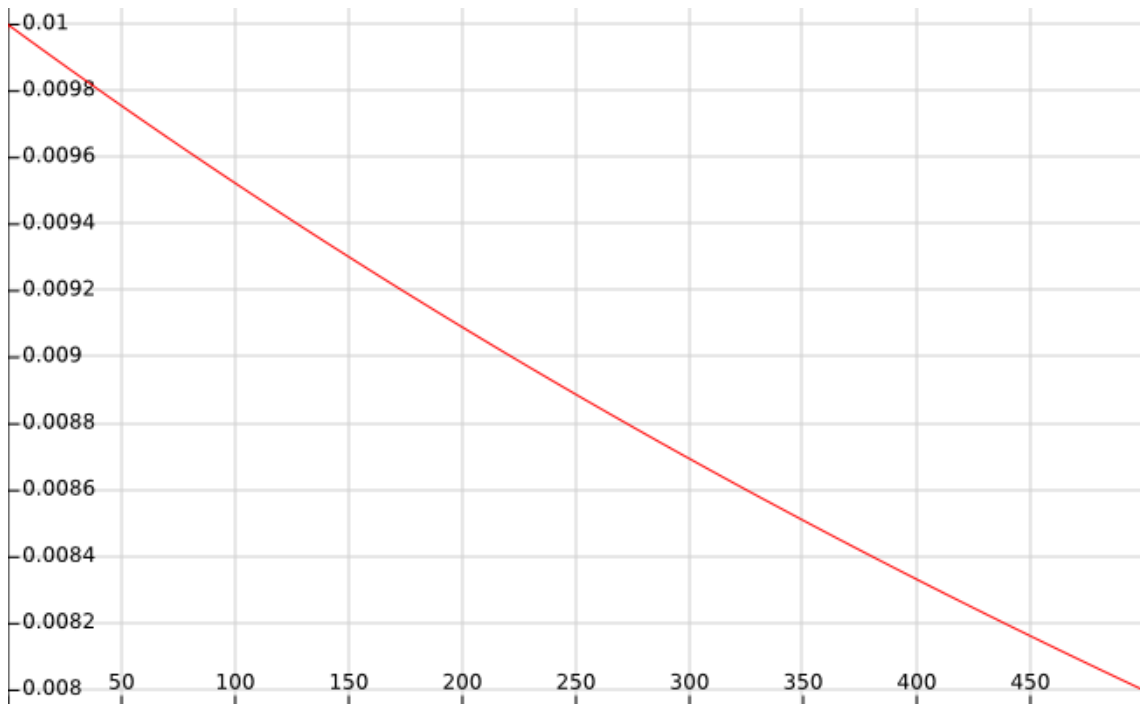


Figure 4.2: AlexNet 學習速率曲線圖

$$R_{decayed} = \frac{R_{initial}}{1 + 0.0005 \times global_step} \quad (4.1)$$

其它設定

由於模型架構較小，可以將 batch 設定為較大的值，最後設定為與 [2] 相同的 128；因為沒有使用遷移式學習的關係，需要比較長的訓練時間讓參數收斂，設定為訓練 500 個 epoch。

其它參數如激勵函數、優化器、損失函數、dropout 皆與第 3.2 節所述一致，使用的資料集、圖片前處理、以及資料增強也與第 3.1 節、第 3.3 節一致。

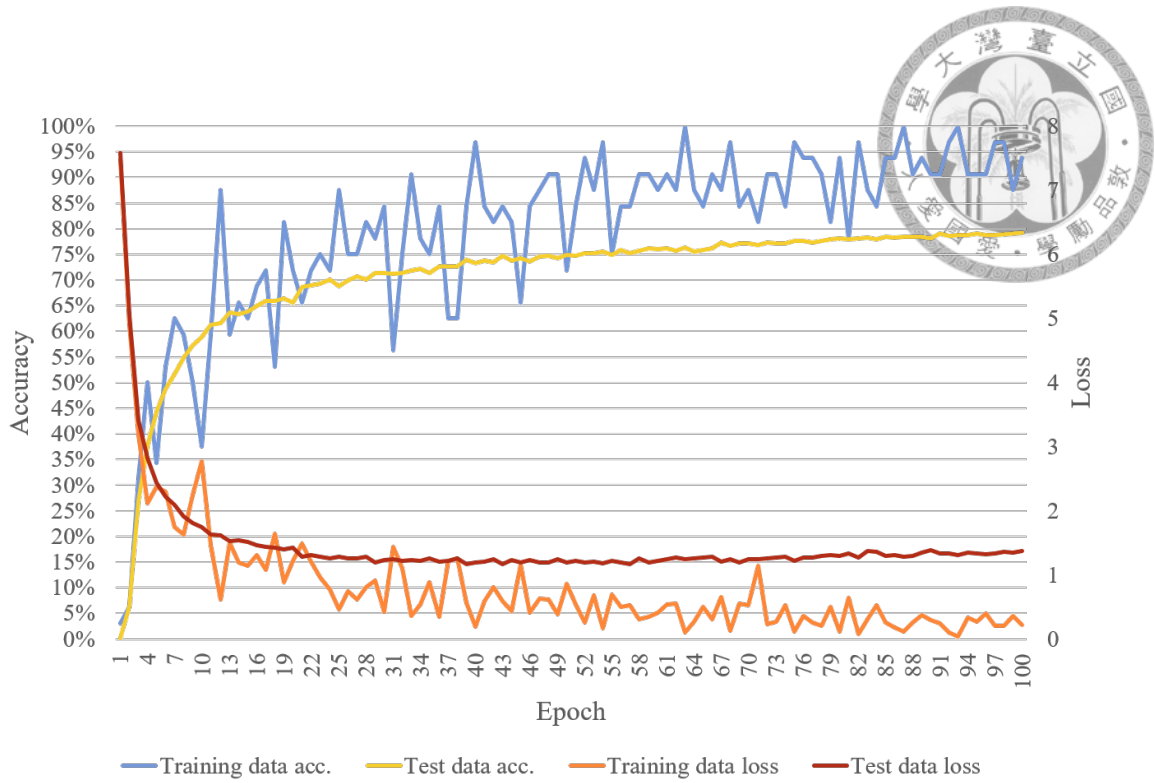


Figure 4.3: VGG-16 訓練過程記錄

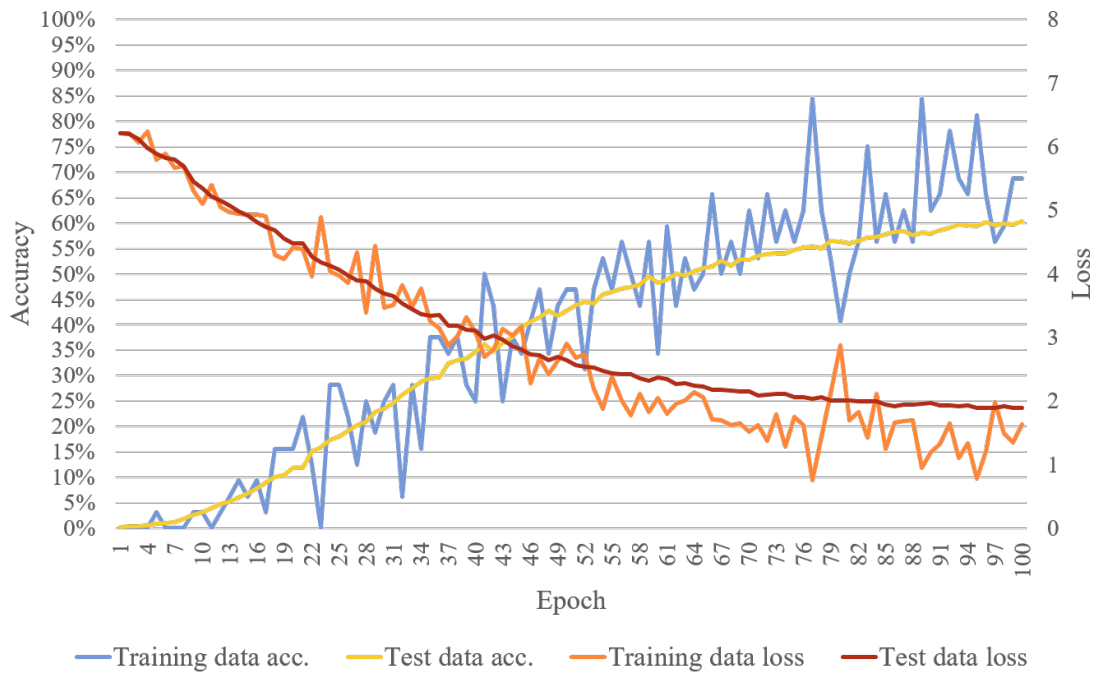


Figure 4.4: VGG-16 (不使用遷移式學習) 訓練過程記錄

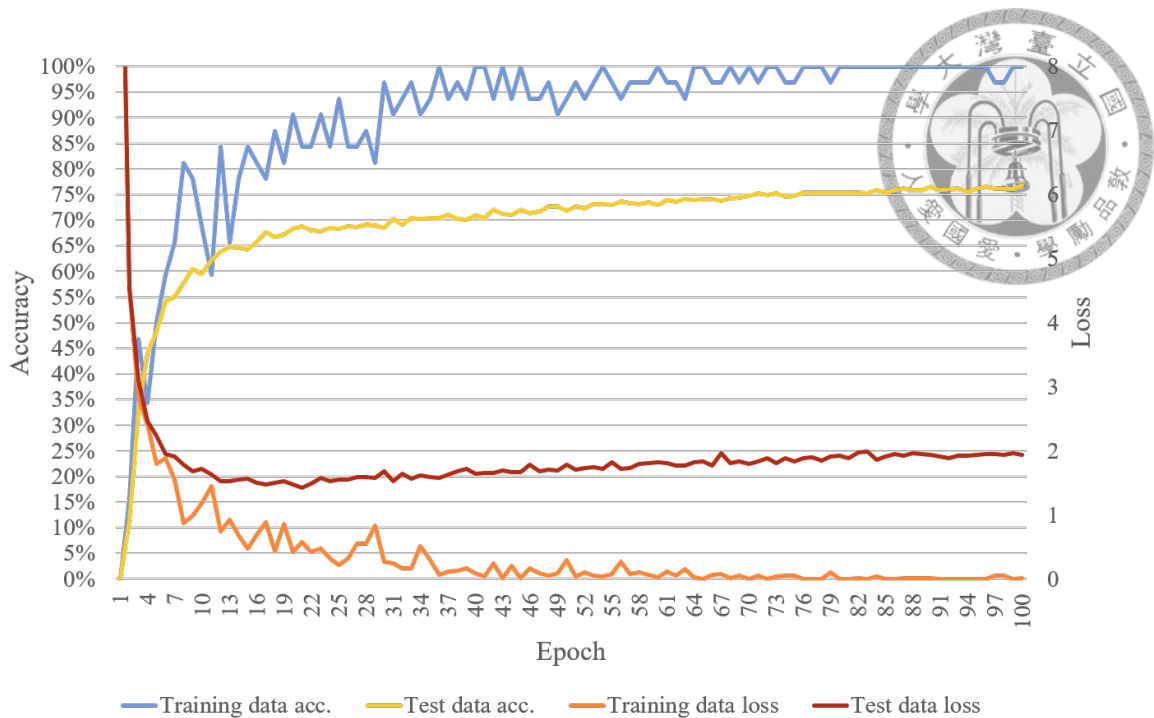


Figure 4.5: VGG-16 (不使用多重尺度學習) 訓練過程記錄

Figure 4.3為 VGG-16 的訓練過程記錄圖。這裡所使用的 VGG-16 相關設定與第 3.2節所述皆相同。Table 4.1為兩個不同模型架構的辨識率之比較表：

Table 4.1: 模型架構與辨識率的比較

	Top-1 辨識率 ¹	Top-5 辨識率 ¹
AlexNet	55.6%	76.08%
VGG-16	79.14%	92.13%

第 3.2節提到了模型權重的初始化方法，本次實驗中是使用遷移式學習，預載大型資料集訓練出的權重進行微調，除了可以得到比較好的成效，也能減少訓練至收斂所花費的時間。Figure 4.4為不使用遷移式學習，所有權重皆為以 Xavier uniform initialization 初始化的結果，訓練的 epoch 同樣設定為 100，其結果如 Table 4.2所示：

Table 4.2: 不使用遷移式學習

	Top-1 辨識率	Top-5 辨識率
VGG-16		
不使用遷移式學習	60.56%	81.13%



第 3.3.1 節中曾介紹資料增強的處理方式，其中有提到利用多重尺度訓練加上隨機裁切，讓模型兼顧全面與局部的特徵資訊，為了測試這項訓練方法的成效，實驗中另外設定了不使用多重尺度訓練的模型，Figure 4.5 為該模型的訓練過程記錄圖，辨識率表示於 Table 4.3：

Table 4.3: 不使用多重尺度訓練

	Top-1 辨識率	Top-5 辨識率
VGG-16 不使用多重尺度訓練	76.21%	90.2%

將以上各種模型架構以及訓練方法的不同所得到的辨識率整理成如下：

Table 4.4: 綜合比較

	Top-1 辨識率	Top-5 辨識率
VGG-16	79.14%	92.13%
AlexNet	55.6%	76.08%
VGG-16 不使用遷移式學習	60.56%	81.13%
VGG-16 不使用多重尺度訓練	76.21%	90.2%

從 Table 4.4 的結果得知，模型架構使用 VGG-16 模型，並且加上多重尺度訓練可以得到當中最高的辨識率；而在同樣的訓練時間內，使用遷移式學習的結果比未使用的對照組還要好，表示遷移式學習能夠有效加速訓練的過程，減少訓練的時間。綜合以上結論，後續都會以 VGG-16、多重尺度訓練、使用遷移式學習的模型進行下一步的實驗。

4.2 圖片再辨識

第 4.1 節的實驗中已經確定好模型的架構與訓練方法的設定，在這一節當中將會實作在第 2.4 節提到的 UCQI 辨識，加上第 3.4 節的修改來找出在測試階段辨識度不足的圖片，經過第 3.5 節提到的圖片後處理方法，對這些圖片以 K-平均分群

¹Top-1/5 辨識率 (Top-1/5 accuracy): 係指在預測類別時輸出 1/5 項類別，且該項/這 5 項預測類別命中實際類別的資料所佔的比率

法進行圖像分割，從分割後的圖片找出最能代表植物本體者，將該圖片輸入模型進行再辨識。整體的流程如 Figure 4.6。

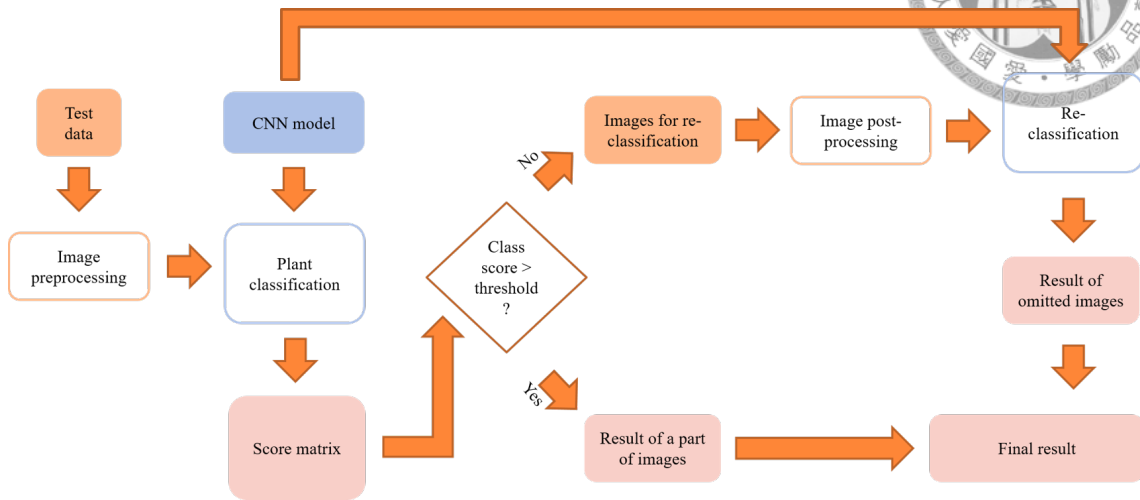


Figure 4.6: 實驗流程圖

實作 UCQI 辨識後，產生的結果如下：

原辨識正確的圖片數量 8912 張

原 top-1 辨識率 79.14%

被選出的圖片數量 290 張

辨識正確的圖片數量（除去被選出的圖片） 8849 張

Top-1 辨識率（除去被選出的圖片） 80.66%



Figure 4.7: 一些被選出的圖片



以上結果可以看出，被選出的圖片有 227 張是擁有錯誤辨識結果的，超過其中一半的比例，若是忽略這些圖片，得到的辨識率會比原有的結果高。

接著繼續對這些圖片實作圖像分割，利用第 3.5 節中提到的，利用質心選擇植物主體的圖片，以這張圖片進行再辨識。實驗的結果如下：

被選出的圖片數量 290 張

分割後的 3 張圖片當中有分類正確者 16 張

分割後選出的圖片為分類正確者 16 張



Figure 4.8: 圖片分割與再辨識的結果。黑線左右兩邊分別為原圖以及圖片分割的輸出，其中藍框為依第 3.5 節所提方法選出的圖片；紅框表示選出的圖片與辨識正確的分割圖片為同一張

Figure 4.8中，上半兩組圖片為再辨識成功的結果，下半兩組為三張分割圖片皆無法辨識出正確類別的結果。

上述的實驗結果中有提到，分割後的圖片當中有分類正確者在 290 張當中有 16 張，觀察它們原來的辨識結果，top-5 分類錯誤的有 1 張，而在 top-5 分類結果正確的 15 張中，top-1 分類正確的有 11 張。

被選出來進行再辨識的 290 張圖片當中，原本分類正確的圖片有 63 張，但在上面的實驗中，即使分割後選出正確的圖片，分類正確的結果僅有 16 張，推測可能的原因在於圖像分割的最後階段，將圖片中的物件分離時，對於被分離物件留空的地方，是以填上黑色做為處理的辦法，這點從 Figure 4.8或是前面章節的 Figure 2.13等便可看出，但在模型的訓練階段中，訓練資料幾乎沒有這種背景為黑的圖片，造成模型所學習、認得的特徵與測試資料的情況有一定的落差，進而造成辨識率的下降。為了試著處理這項問題，提出的解決方法如下：

4.2.1 實驗 — 新增訓練資料

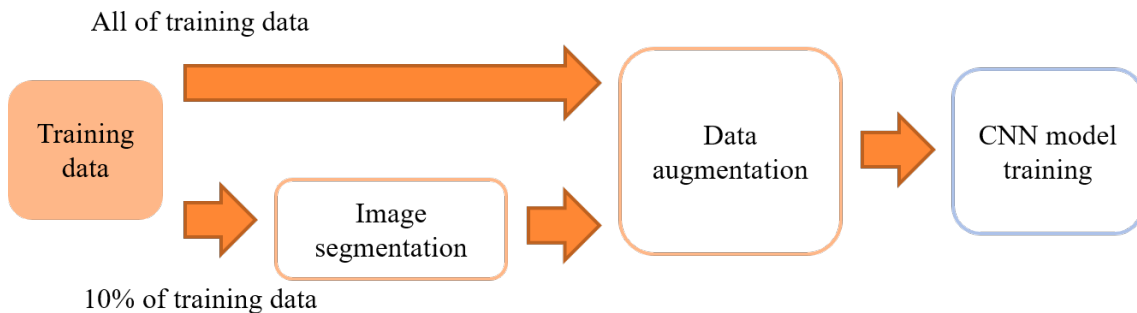


Figure 4.9: 加入額外的訓練資料

上述分析中，推斷造成辨識率下降的原因可能是在於，訓練資料缺乏分割之後背景為黑的圖片，因此在這項實驗中，會在訓練資料裡也加入以訓練圖片進行圖像分割之後的圖片，具體的實驗設定為：隨機選取 10% 的訓練資料進行圖像分割（具體的數量為 10202 張），並且一樣選出植物主體的圖片做為代表，並將這些圖片加入訓練過程中（Figure 4.9）。這裡所使用的圖像分割以及選取分割圖片的方法都與一開始的實驗中對測試資料所進行的處理一樣，與第 3.5 節所述相同，另外，進行圖像分割的 10% 訓練資料的原圖片在訓練過程中並不會被排除，依然會

輸入模型。

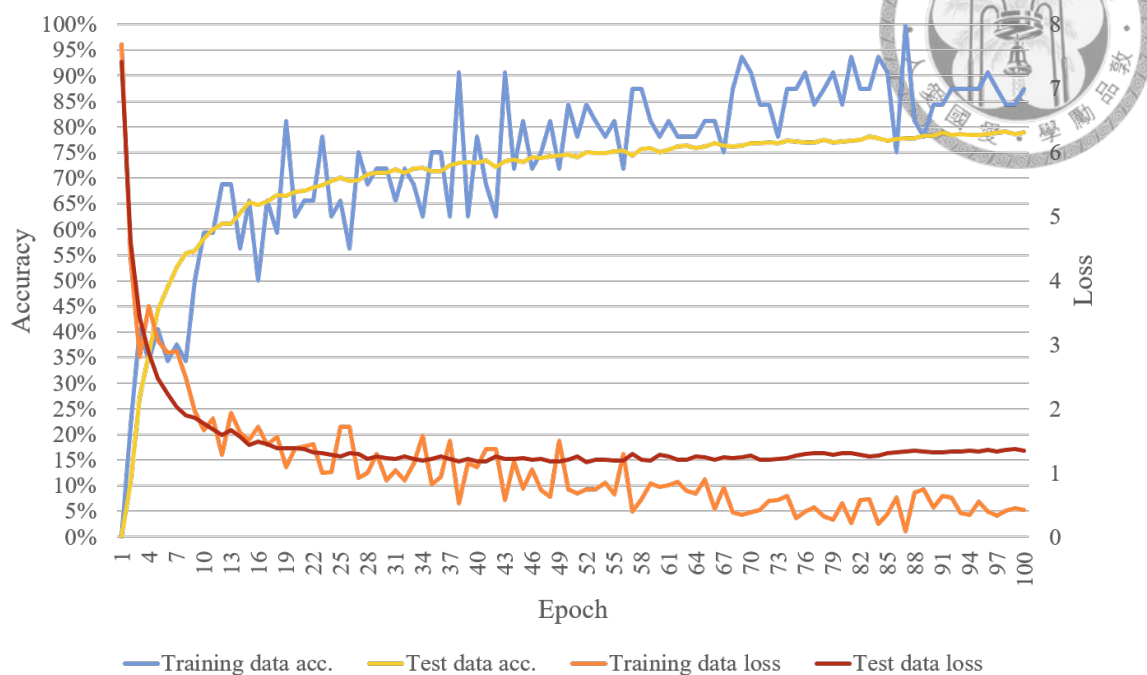


Figure 4.10: 加入額外的訓練資料後所進行的訓練記錄

Figure 4.10為加入額外的訓練資料後所進行的訓練過程折線圖，該模型最後的辨識率如 Table 4.5所示：

	Top-1 辨識率	Top-5 辨識率
VGG-16		
加入額外的訓練資料	78.81%	92.14%
VGG-16	79.14%	92.13%

與一般未使用圖像分割後的圖片做為訓練資料的模型比較，加入這些額外的訓練資料對於辨識率並沒有太大的影響。接著實作 UCQI 辨識以及進行再辨識後，得到的結果如下：

原辨識正確的圖片數量 8875 張

原 top-1 辨識率 78.81%

被選出的圖片數量 291 張



辨識正確的圖片數量（除去被選出的圖片） 8813 張

Top-1 辨識率（除去被選出的圖片） 80.34%

分割後的 3 張圖片當中有分類正確者 3 張

分割後選出的圖片為分類正確者 3 張

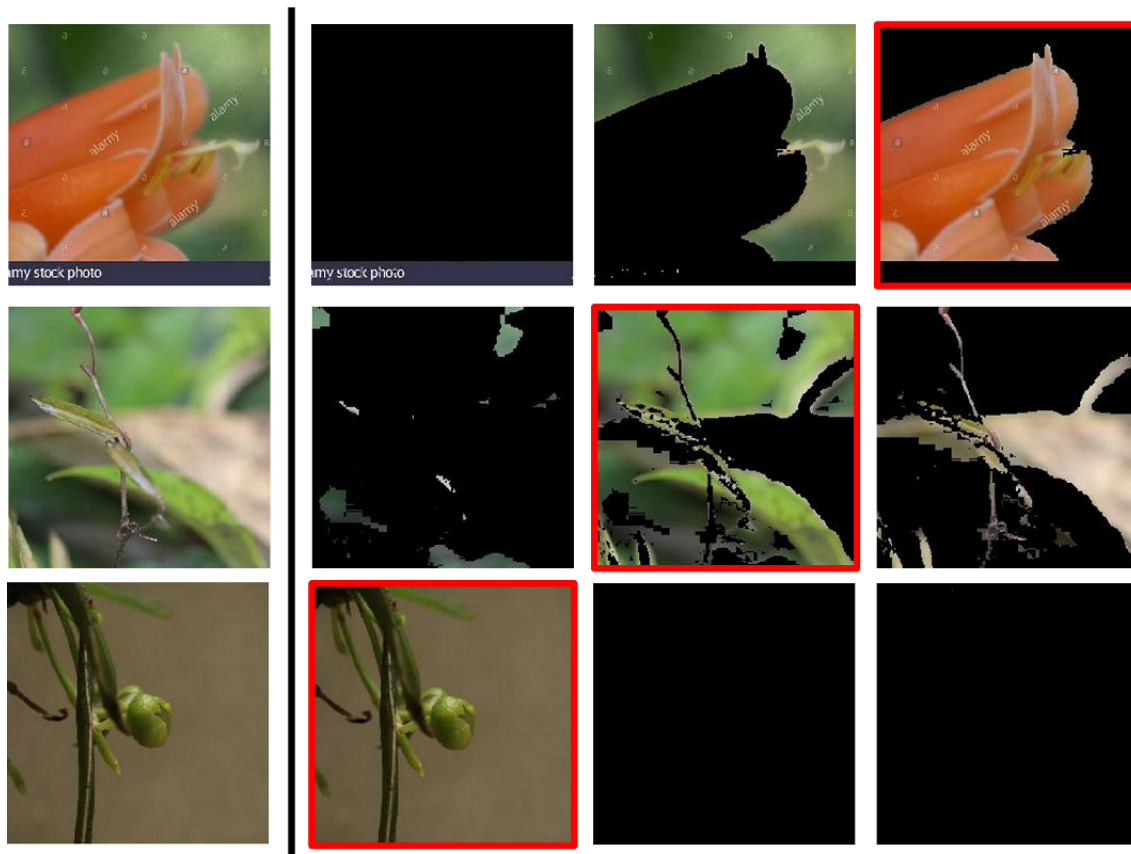


Figure 4.11: 加入分割圖片訓練模型後的再辨識結果。與 Figure 4.8 相同，紅框代表的是能夠辨識正確的分割圖片，同時也是依第 3.5 節所提出之方法選出的圖片

加入了額外的訓練資料後，雖然被選出的圖片數量並無太大差別，但再辨識的結果卻比原本的實驗結果還要不理想，能夠再辨識正確的圖片數量從原本的 16 張降為僅有 3 張，其中原辨識結果 top-5 分類正確的有 3 張，top-1 分類正確的有 0 張。

除了上述實驗，加入分割後圖片做為額外的訓練資料外，以下將實驗另外一種避免分割圖片與訓練圖片的差異造成的辨識率下降：



4.2.2 實驗一 背景補色

在先前的分析中推論，對於圖像分割時對於被分離的部分，若是以填黑做為補償辦法將會造成模型對於分割圖片的辨識率低落，因此在接下來的實驗中，將會嘗試不同於填黑的處理方法，去降低因為訓練資料與測試資料的差異造成的辨識率下降之影響，具體實作方法如下：

- (1) 依照前述 (第 3.5 節) 之方法，分割圖片後選出能代表植物主體的圖片
- (2) 計算另外 2 張未採用圖片 RGB 值的平均
- (3) 對於在 (1) 中被選出的圖片，其中 RGB 值為 (0,0,0) 的地方將 RGB 值改為 (2) 算出的平均值

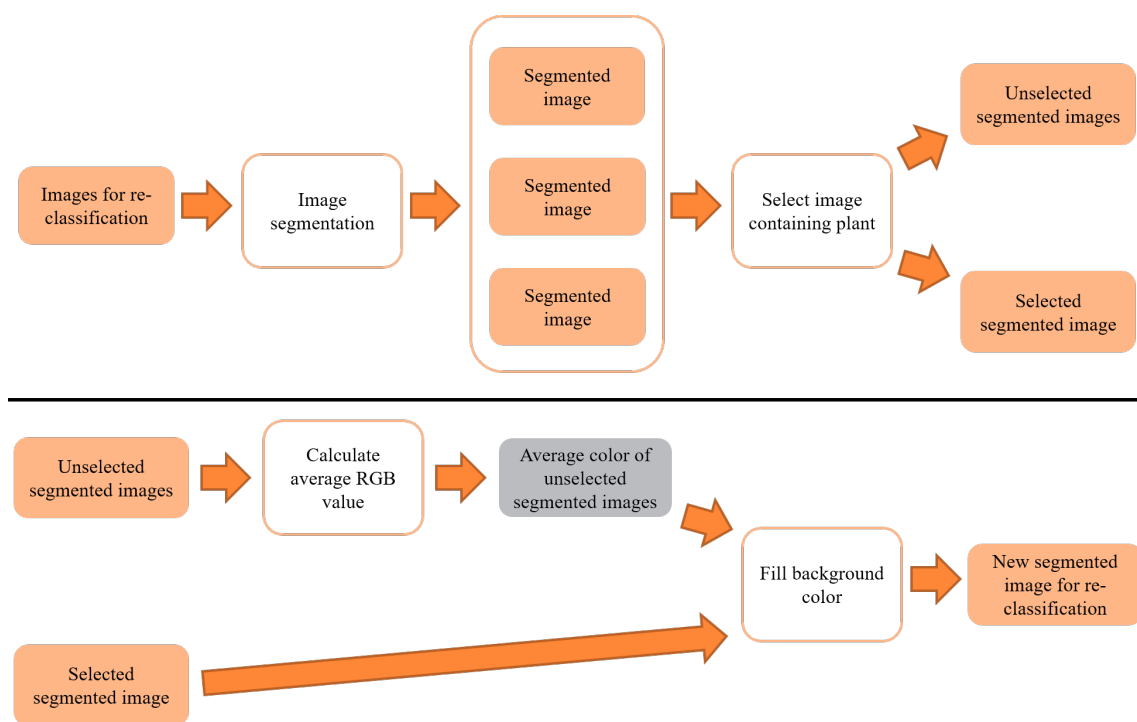


Figure 4.12: 背景補色的流程圖

在 (3) 提到的 RGB 值為 (0,0,0) 的地方，實際上就代表圖像分割時物件被分離時留空的地方，而對於這些因留空而填黑的地方，在這次實驗中會改為利用 2 張未被採用的圖片，求出 RGB 值平均後填入這些留空的地方，減少分割後的圖片與訓練資料的背景色差異 (Figure 4.13)。

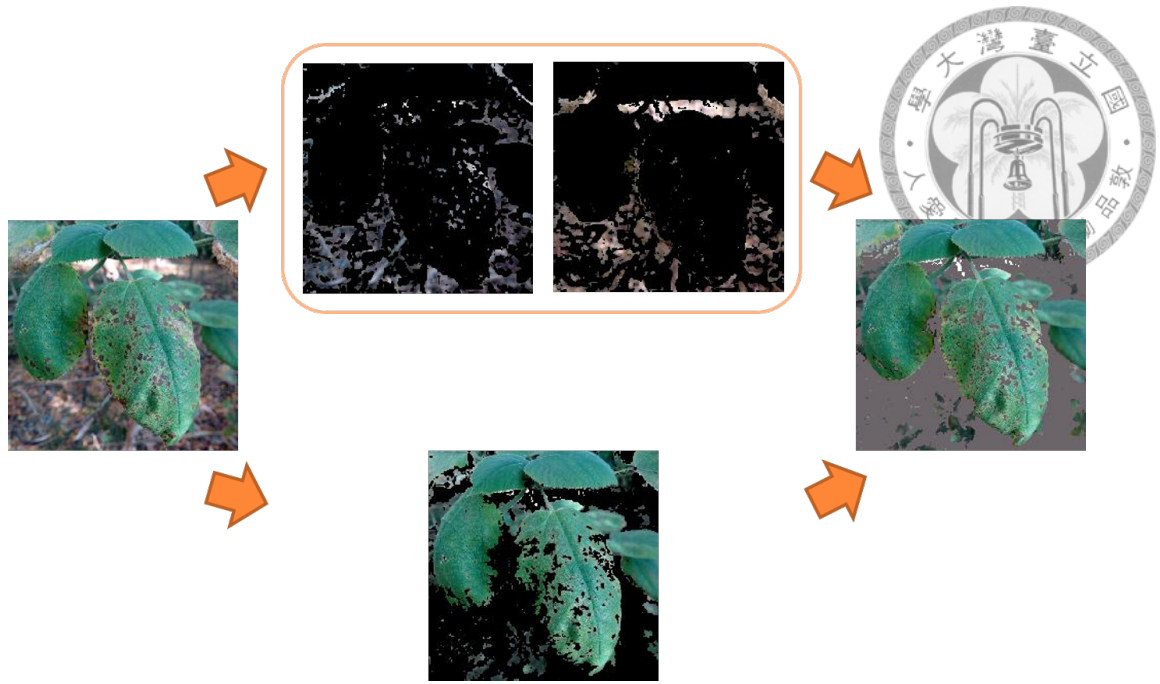


Figure 4.13: 背景補色

實驗的結果如下：

被選出的圖片數量 290 張

圖像分割+補色後分類正確者 23 張

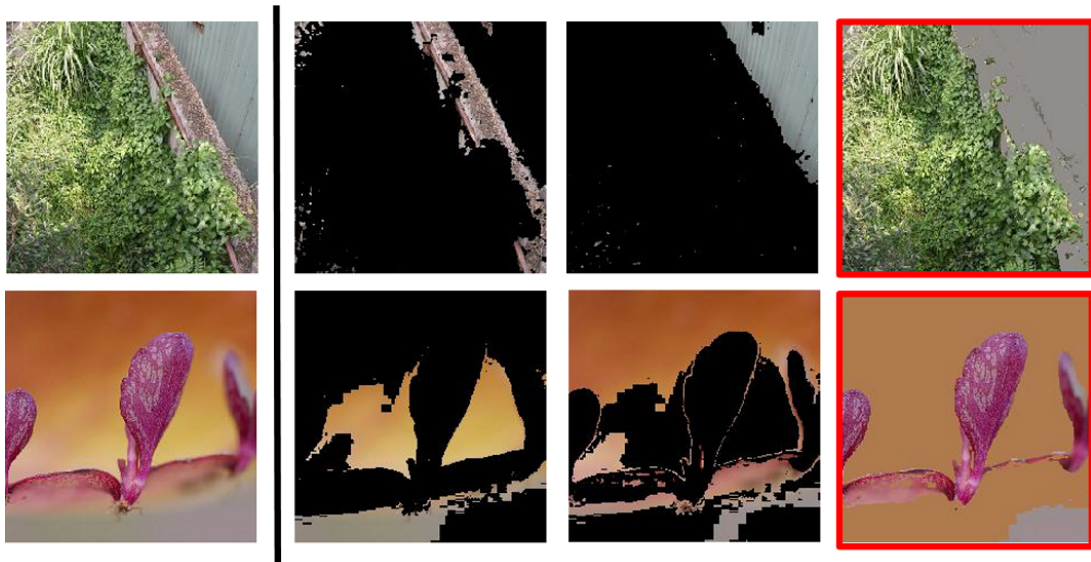


Figure 4.14: 經過圖像分割與背景補色後辨識正確的結果。紅框代表的是被選出的圖片經過背景補色後的成果



與本節一開始的實驗結果相比，經過背景補色後辨識正確的圖片數量從原本的16張增加到了23張，在這23張分類正確的圖片中，原辨識結果 top-5 分類錯誤者有4張，而在 top-5 分類正確的19張中，top-1 分類正確者有10張。

Figure 4.15為在原本實驗中辨識不正確，但在經過背景補色處理後成功辨識的圖片。比較兩張處理過後的圖片，經過背景補色的圖片與原圖的相似度較高，但仍然保有強調植物主體的特性。



Figure 4.15: 原圖、分割圖與補色圖。每組圖片的左、中、右圖各代表了原圖、圖像分割後選出的圖片、經過背景補色的分割圖片

Table 4.6為本節的實驗結果整理一覽表：

Table 4.6: 再辨識結果比較

	原實驗	新增訓練資料	背景補色
原辨識正確的圖片數量	8912	8875	
原 top-1 辨識率	79.14%	78.81%	
被選出的圖片數量	290	291	(與原實驗一致)
辨識正確的圖片數量 (除去被選出的圖片)	8849	8813	
Top-1 辨識率 (除去被選出的圖片)	80.66%	80.34%	
分割後的3張圖片當中有分類正確者	16	3	23
分割後選出的圖片為分類正確者	16	3	23



Chapter 5

結論與未來展望

5.1 回顧與結論

回顧本次研究，一開始（第 2 章）先從各個角度探討了以卷積神經網路模型進行圖片分析，判斷圖片中的植物種類時遇到的幾個議題，從基礎的模型架構探討、比較與選擇（第 2.1 節到第 2.3 節）、講述測試資料的類別若不屬於訓練資料所包含的類別會發生的問題與為了對應上述情況，由 Siang Thye Hang 等人提出的 UCQI 辨識法（第 2.4 節）、介紹用以強調圖像物件的其中一種方法與實作（第 2.5 節）。

接著在第 3 章說明本次實驗的相關事項，包括所使用的資料集（第 3.1 節）、模型架構與細部的參數相關設定（第 3.2 節）、圖片前處理與資料增強的過程（第 3.3 節）、以及如何修改 UCQI 辨識法以符合本次研究的需求（第 3.4 節）。

最後，第 4 章的內容為實作前述提到的各種方法並且分析結果，具體來說，實驗各種不同的模型架構與訓練方法，並且比較它們各自的結果，從中選擇辨識率最高者（第 4.1 節）、選擇完合適的模型與訓練方法後，實作 UCQI 辨識法對測試圖片進行選擇，並對被選出的圖片進行處理後輸入模型再辨識（第 4.2 節），同時也進行各種實驗，力求改善、增加辨識率。

經過實驗與結果比較，VGG-16 模型在本次實驗使用的 top-500 資料集中擁有的辨識率，並且加上多重尺度學習能夠使辨識率更加提升，另外，利用遷移式學習初始化模型權重能夠有效減少權重收斂的時間，加速訓練過程。在第 4.1 節實驗中得到最佳的結果為：經過約 20 小時（100 epoches）的訓練

後，得到 top-1 辨識率 79.14% 與 top-5 辨識率 92.13% 的成果。

第 4.2 節則是集中在圖片再辨識的階段，實作前述的 UCQI 辨識法與圖像分割，驗證再辨識的成效，並且實驗兩種改善方法，分別為在訓練階段時增加圖像分割的圖片，以及對分割後的圖片執行背景補色，減少測試圖片與訓練圖片的誤差。實驗結果顯示，實作背景補色能夠提升再辨識的成功率，成功辨識 290 張辨識度不足的圖片其中 23 張。



5.2 改進方向與未來展望

回顧以上實驗過程與結果，其中有幾個細節是可以修改或是留有改進的地方，以下將講述這些改進的方向：

5.2.1 模型架構

在本次研究中，採用了 VGG-16 模型做為主要的卷積神經網路模型架構，另外也實驗了 AlexNet 的效能，這兩者分別為 ILSVRC2014 年的亞軍 [1] 與 ILSVRC2012 年的冠軍，隨著時間推進與硬體效能的提升，眾多的研究與新的模型架構概念不斷被提出 [22]。

回顧 ILSVRC 的歷屆冠軍，2015 年由 Kaiming He 等人提出的 ResNet [23] 之中，為了解決模型架構深度增加時產生的梯度消失問題提出了殘差學習區塊 (residual learning block) 的架構 (Figure 5.1)，受惠於這項研究成果，ResNet 的架構可以高達 152 層，之後的研究甚至能夠達到 200 層 [24]。

2014 年，由 Christian Szegedy 等人提出的 Inception (GoogLeNet) [25] 拿下了該年 ILSVRC 的冠軍，他們提出了網路內網路 (network in network) 的架構 (Figure 5.2)，讓網路在同一層中便享有不同尺度的特徵，另外，還有採用 1×1 卷積核降低維度 (Figure 5.2 的灰色區塊)，在 1×1 卷積核後加入 ReLU 激勵函數使得模型更加非線性化等等新概念，之後也不斷有改良版的架構發布 [26,27]，其中還有加入前述 ResNet 架構的 Inception-ResNet [28]。

正如一開始所述，模型架構與最後的辨識率有著很大的關連性，因此需要不斷嘗試各種不同的模型，找出最適合實驗所使用之資料集的模型架構，再以此為基

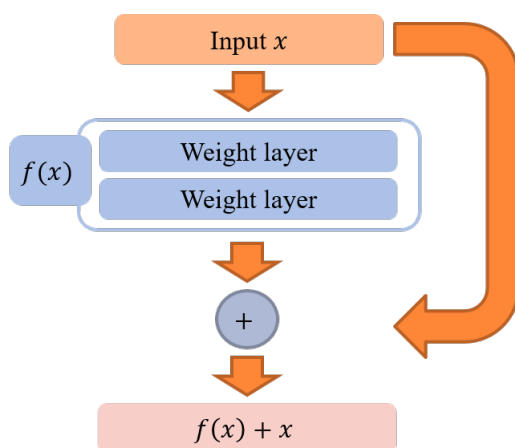


Figure 5.1: 殘差學習區塊

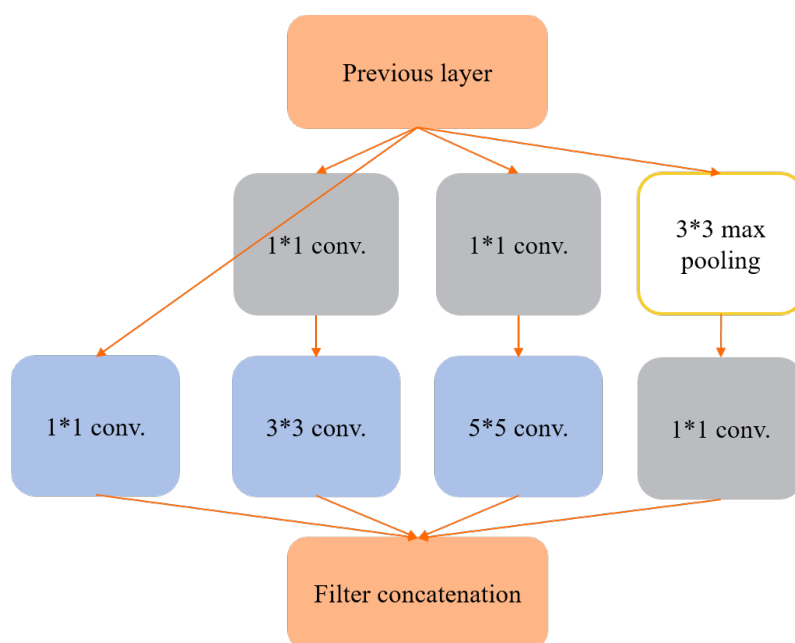


Figure 5.2: Inception 架構

底進行參數的微調。

5.2.2 測試方法

在第 3.3.1 節中曾介紹多重尺度學習的訓練方法，透過每次將訓練圖片縮放為不同大小的方式，讓模型能夠兼得全面與局部的特徵，而這種作法也能應用在測試階段上 [29]，在測試階段時，將測試圖片縮放成不同的大小 (Figure 5.3)，並將各個大小的圖片皆輸入模型辨識，將結果處理過後得到最後的辨識類別，而在過程中也會遇到模型輸入大小與縮放後的圖片大小不符的情況，除了與第 3.3.1 節相同，使用隨機裁切來處理以外，也能將卷積神經網路架構後半的全連接層以卷積

層取代 [30]，藉此解決卷積神經網路的輸入大小必須固定的問題。

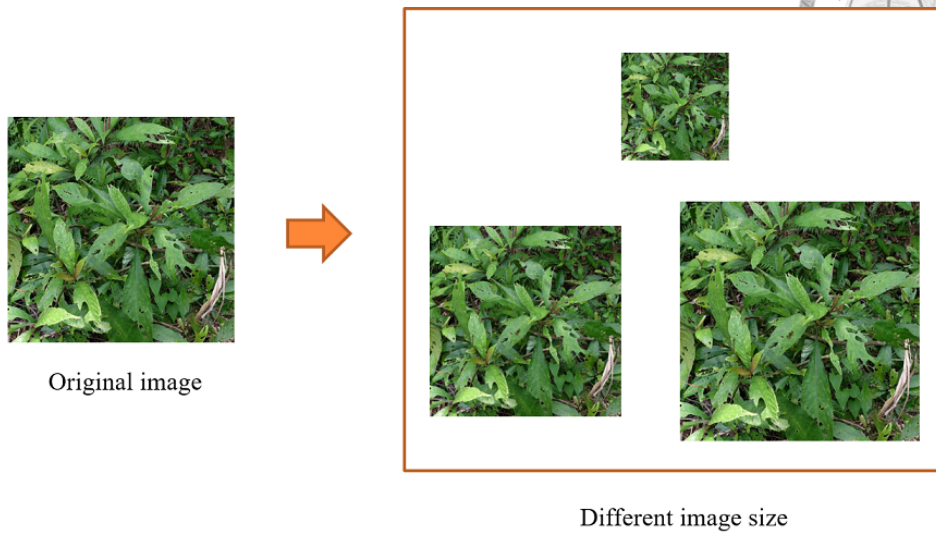


Figure 5.3: 多重尺度測試

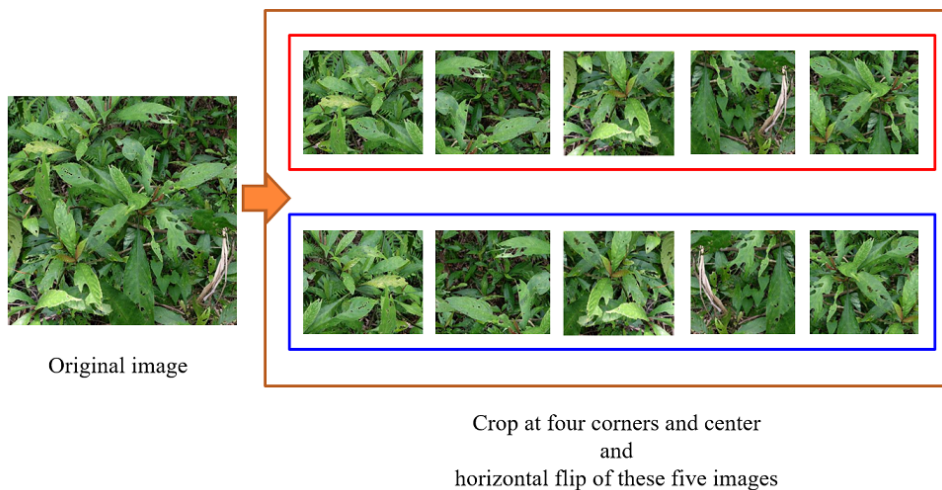


Figure 5.4: 多重圖片測試

5.2.3 UCQI 的門檻值

第 2.4 節曾提到由 Siang Thye Hang 等人提出的 UCQI 辨識法，並且在第 3.4 節說明如何修改以符合本次實驗中找出辨識率較低的測試圖片。其主要概念是以全連接層的輸出當作圖片對於每個類別的分數，分數愈高代表與該類別的關聯性愈高，愈有可能屬於該類別，而在訓練階段時會找出每一類別的圖片當中分數最小者，做為該類別的門檻值，接著在測試階段時，測試圖片在原本的分類過程中被

預測屬於某類別後，還需要與對應的門檻值相比較，若圖片輸出的分數不低於門檻值時，才會真正將它歸類在該類別。

從上述的步驟可以發現，門檻值的選出基本上是以訓練資料做為依據來決定的。而在 Siang Thye Hang 等人的文章 [8] 中，有提出如何利用驗證資料 (validation data) 來調整門檻值：

$$r_i = \frac{v_i}{t_i}, i = 1, \dots, N \quad (5.1)$$

(5.1) 中， t_i 代表基於訓練資料找出的各類別門檻值， v_i 則是基於驗證資料的門檻值，利用這兩者算出 r_i 後，再進行以下過程調整門檻值：

$$z_i = \begin{cases} r_i, & 0 < r_i < 1 \\ 0, & \text{otherwise} \end{cases}, i = 1, \dots, N \quad (5.2)$$

$$\bar{z} = \frac{1}{n} \sum_{i=1}^N z_i, n = \|z\|_0 \quad (5.3)$$

$$t' = \bar{z}t \quad (5.4)$$

(5.3) 中， $\|z\|_0$ 為 10 範數 (10-norm)，代表 z 向量中非零元素的數量。

以上式子的目的在於找出驗證資料門檻值比訓練資料門檻值還要小的類別，算出它們之間的比率，求得平均之後對整體門檻值進行調整。在 [8] 的實驗中，經過驗證資料調整門檻值過後，被選出的圖片數量減少了，而模型的辨識率則得到進一步的提升。

在本次實驗中，只有將資料集分為訓練資料與測試資料兩個子資料集，並沒有再額外分割出驗證資料，因此，使用驗證資料進行門檻值的調整將是未來實作的方向。另外，本次實驗與原文章中，門檻值的選出是以各類別當中分數最小者做為該類別的門檻值代表，而在原文章的最後也提出，是否能夠從這一點進行改進，譬如說以各類別分數的平均值或是中位數取代原本使用的最小值，藉此取得更合適的類別門檻值。



5.2.4 圖片後處理

在經過修改後的 UCQI 辨識之後，辨識度不足的图片會被選出以進行之後的圖片後處理與再辨識過程，其中，圖片後處理在本次的實驗係指對圖片以 K-平均分群法進行圖像分割，嘗試將植物主體與背景或其它不相關的雜訊分離後，找出含有植物主體的图片並以此進行再辨識。對於以上關於圖像後處理的步驟，未來可嘗試的地方如下：

K-平均分群法

在第 2.5.1 節中，講述了 K-平均分群法的演算法，其中在最後一個步驟是這樣敘述的：

(4) 重複步驟(2)(3)直到分群中心幾乎不再變動，或是已執行規定的迭代次數為止。

接著在第 3.5 節關於圖片後處理的章節中，提到了本次實驗中關於 K-平均分群法的參數設定，其中迭代次數固定為 5 次，而不是上述提到的「直到分群中心幾乎不再變動」，未來或許可以改進成，當分群中心的變動小於某個門檻值後再停止整個分群過程。與原本固定迭代次數的設定相比，這樣的作法可能會花費更多的時間進行分群，此外，如何找出合適的門檻值又會是另外的課題。

除了上述關於演算法迭代的部分，還可以考慮分群的數量與初始化的方法，這兩點是影響最後分群結果甚鉅的兩個要素，在原本的 K-平均分群法，分群的數量為固定數值，而分群中心的初始化則是隨機決定，因此可能會出現事先決定的分群數量不適合該圖片，或是因為初始化的中心不佳造成最後分群結果的不理想，目前也有許多研究 [31] 提出改良的方法來避免發上以上的情況。

其它圖像分割方法

除了 K-平均分群法以外，第 2.5 節中還提到其它的圖像分割演算法，除了分析灰階像素值、邊緣偵測、群聚分析等等較為經典的方法外，也有人嘗試利用深度網路進行圖像分割 [32]。



Figure 5.5: 不理想的圖像分割成果

選擇分割後圖片

執行完圖像分割之後，接著要從分割後的結果中選出最能代表植物主體者做為再辨識的圖片，在第 3.5 節中有提到，實驗中所使用的方法為：

- (1) 各別計算 3 張圖片的質心位置
- (2) 各別計算 3 張圖片的質心到圖片中央的距離
- (3) 選擇距離最小者做為再辨識的圖片

Figure 5.6 為使用上述選擇方法時，得到的結果較為不理想的圖片。在這些圖片當中，有些是因為植物的主體較小或是較遠離圖片中心，因此畫面比重較大的其它物件（背景或是其它雜訊等等），其質心和主體質心相比，與畫面中心的距離較短，造成圖片選取的錯誤，抑或是背景與植物主體一樣均勻分布在畫面之中，也有可能造成背景質心比植物主體質心更接近中心的情況。這些不理想的結果會造成原本可以辨識正確的圖片，因為選擇錯誤的關係而將較無關聯的圖片做為再測

試用圖片而得到不正確的結果。未來可嘗試考慮更多要素，或是實作其它選擇圖片的方法，譬如：利用訓練資料的圖像分割後成果，訓練出用於分割圖片的分類器等，以避免如 Figure 5.6 所示，較不理想的分割圖片選擇成果。



Figure 5.6: 不理想的分割圖片選擇成果。藍框表示依第 3.5 節所提出方法選擇的圖片

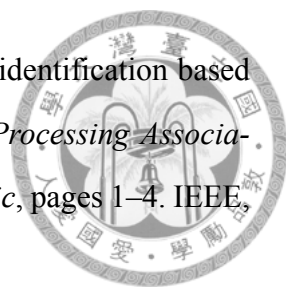
用於再辨識的模型


第 4.2.1 節的實驗內容為，加入額外的分割圖片做為訓練資料來嘗試訓練出對分割圖片辨識度較高的模型。上述實驗中是將分割圖片加入原模型的訓練過程，且原辨識過程與再辨識使用的是同一模型，未來可嘗試建立專門用於再辨識的模型，其訓練資料只由分割圖片所組成，並驗證該模型是否能夠提升再辨識的效果。




參考文獻

- [1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.
- [4] Sue Han Lee, Chee Seng Chan, Paul Wilkin, and Paolo Remagnino. Deep-plant: Plant identification with convolutional neural networks. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 452–456. IEEE, 2015.
- [5] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [6] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.
- [7] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

- 
- [8] Siang Thye Hang and Masaki Aono. Open world plant image identification based on convolutional neural network. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2016 Asia-Pacific*, pages 1–4. IEEE, 2016.
- [9] Hervé Goëau, Pierre Bonnet, and Alexis Joly. Plant identification in an open-world (lifeclef 2016). In *CLEF 2016-Conference and Labs of the Evaluation forum*, pages 428–439, 2016.
- [10] Song Yuheng and Yan Hao. Image segmentation algorithms overview. *arXiv preprint arXiv:1707.02051*, 2017.
- [11] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- [12] Salem Saleh Al-Amri, NV Kalyankar, and SD Khamitkar. Image segmentation by using edge detection. *International Journal on computer science and engineering*, 2(3):804–807, 2010.
- [13] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.
- [14] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.
- [15] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
- [16] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466, 1952.
- [17] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

- 
- [18] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.
- [19] SM Aqil Burney and Humera Tariq. K-means cluster analysis for image segmentation. *International Journal of Computer Applications*, 96(4), 2014.
- [20] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.
- [21] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
- [22] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.
- [23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.
- [25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

- 
- [26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.
- [28] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.
- [29] Andrew G Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013.
- [30] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.
- [31] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.
- [32] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.