

2022 年臺灣國際科學展覽會

研究報告書

編 號：190020-121

作 者：吳振榮

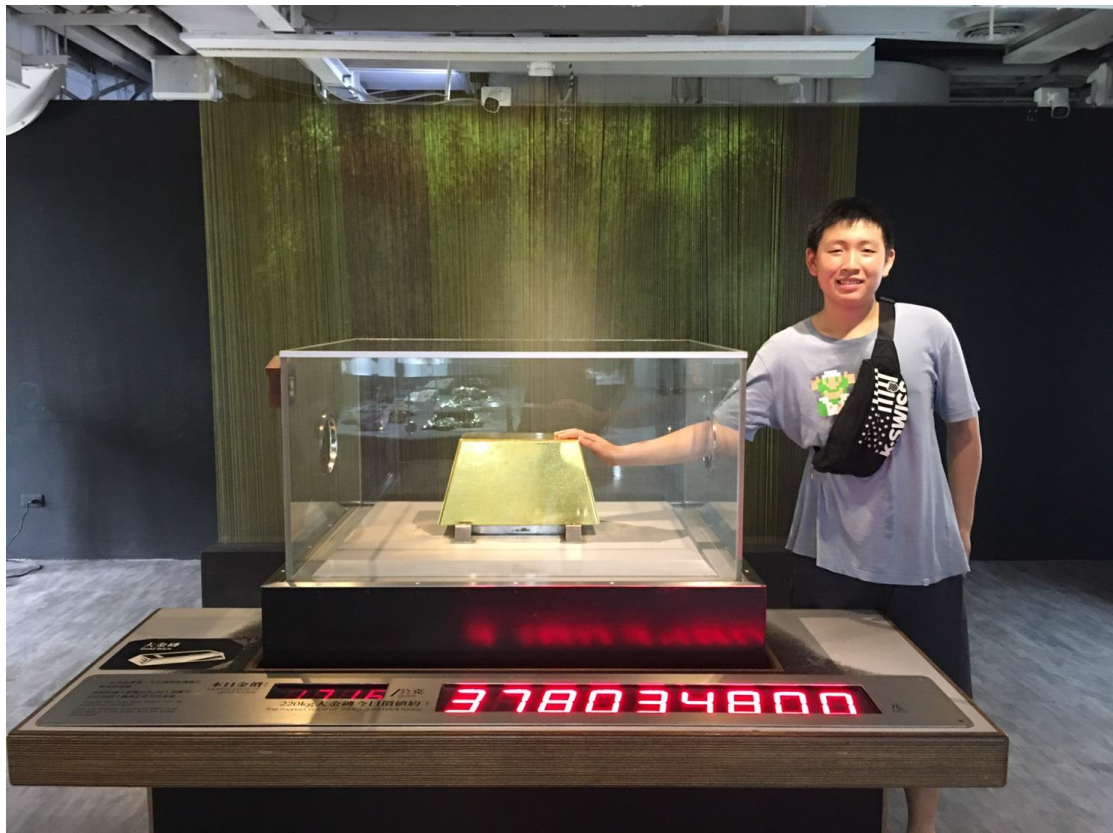
作品名稱：利用半監督式學習進行自動星系分類

學 校：新北市立中和高級中學

指導教師：王一哲、江振國

關 鍵 詞：人工智慧、深度學習、星系分類

作者簡介



我是就讀中和高中的吳振榮，目前就讀高二，高一的時候第一次接觸到程式語言，在高二的時候決定利用高一所學的機器學習知識來參加國際科展，在研究的過程中遇到許多的困難，時常會因為準確率太低而感到很氣餒，但隨著不斷的嘗試，許多問題也逐漸解決了。非常謝謝教授、老師及學長的幫助才能順利完成研究。

摘要

本研究使用半監督式機器學習搭配卷積神經網路來訓練核心模型，並將星系的圖片加入模型裡，讓電腦自動判斷出該星系的種類為何。我使用自行設計的 CNN 架構以及 VGG-16 當作我的卷積神經網路架構。資料集來源為 EFIGI 和 Galaxy Zoo 2。我分為兩種任務，第一種任務是讓模型能分辨橢圓 (E)、螺旋 (S)、不規則 (I) 這三種類別的星系，訓練資料共有 2,468 張照片，最後的正確率能達到 94%。第二個任務是將 8 種的星系照片 (E、S0、Sa、Sb、Sc、SBa、SBb、SBc) 進行分類，並使用自動編碼器作為預訓練，使用 1,923 張 EFIGI 的圖片以及 1,258 張 Galaxy Zoo 2 的照片當作訓練資料。由於各星系照片有許多外觀太過相似，測試準確度最高達到 54.12%，基於我的研究，星系自動化辨識於天文學上應該有相當大的運用空間。

Abstract

In this study, in order to classify the pictures of galaxies automatically, we have built a model of convolutional neural networks (CNN). With the optimized VGG-16 structure and semi-supervised machine learning strategy, the model has shown surprising accuracy and efficiency in the differentiation between galaxies. The datasets of galaxy pictures are from EFIGI and Galaxy Zoo 2 (GZ2). First, three major galaxy types: Ellipticals (E), Spirals (Sc) and Irregular (I) have been well classified. With 2,468 galaxy images as training data, the model can reach 94% classification accuracy. Then, we expanded classification to 8 types: E, S0, Sa, Sb, Sc, SBa, SBb, and SBc. With the help of semi-supervised strategy which using autoencoder as pretrained model, data for model training was successfully enlarged from 1,923 to 3,181 pictures, and the averaged classification accuracy finally increased from 12.87% to 54.12%. Despite the similarity among spiral galaxies, the model still can differentiate E (80%), Sb (80%), SBb (85%) and Sc (85%) patterns. In conclusion, machine learning with semi-supervised strategy has been shown a better solution when well-recognized galaxy patterns are not enough for model training, and we hope the method could paved a new way for automatic astronomy patterns recognition.

壹、前言

(一) 研究動機

近年來人工智慧 (artificial intelligence, AI)、機器學習 (machine learning)、深度學習 (deep learning) 這幾個名詞十分熱門，加上硬體設備的進步以及更多可用的深度學習框架，例如 TensorFlow 和 PyTorch，使得機器學習不再只是高科技公司的專利。近年來人工智慧被廣大運用在不同領域，例如：電腦視覺 (computer vision)、自然語言處理 (nature language processing)、語音辨識 (voice recognition).....等。而我對於圖形辨識的模型特別有興趣，試著將這樣的模型套用在其它的領域上。

某次我在地球科學課本上看到有許多不同種類的星系時，對於分類星系的方式感到很有興趣。目前最廣泛使用的分類方法為哈伯序列 (Hubble sequence)，這是依據星系的外觀進行分類，但有不少星系的外觀十分相似，也有的星系外觀介於兩個種類之間，利用人眼進行分類會非常耗時費力。於是我想要建立並訓練一個可以自動將星系分類的模型。

(二) 研究目的

本研究試圖建立基於人工神經網路的圖形辨識模型，並將模型應用於星系自動分類。為了達成這個目的，我提出的構想的如下：

- 1、建立並訓練簡易的卷積神經網路 (convolutional neural network, CNN) 模型。
- 2、建立並訓練基於 VGG-16 架構的卷積神經網路模型。
- 3、利用半監督式機器學習來增加模型判斷正確率。
- 4、將星系的照片輸入兩種模型內，檢測模型分類星系的準確率。

貳、研究設備及器材

(一) 訓練資料集

1、 EFIGI (Extraction de Formes Idealisées de Galaxies en Imagerie)

官方網站上提供了 4458 張與太陽系距離較近的星系照片，並且附上每個星系詳細的分類資料。(Bertin et al., 2011)

2、 Galaxy Zoo 2

Galaxy Zoo 2 (以下簡稱為 GZ2) 的作法有別於其它星系分類計畫，GZ2 是由大量的志願者依據圖 1 圖 1 的流程進行分類工作，已完成分類的星系共有 304,122 個，但分類的方式與哈伯序列不相同，所以我不使用他分類出來的標籤，只單純使用照片，我下載其中 243,434 張星系照片。(Willett et al., 2013)

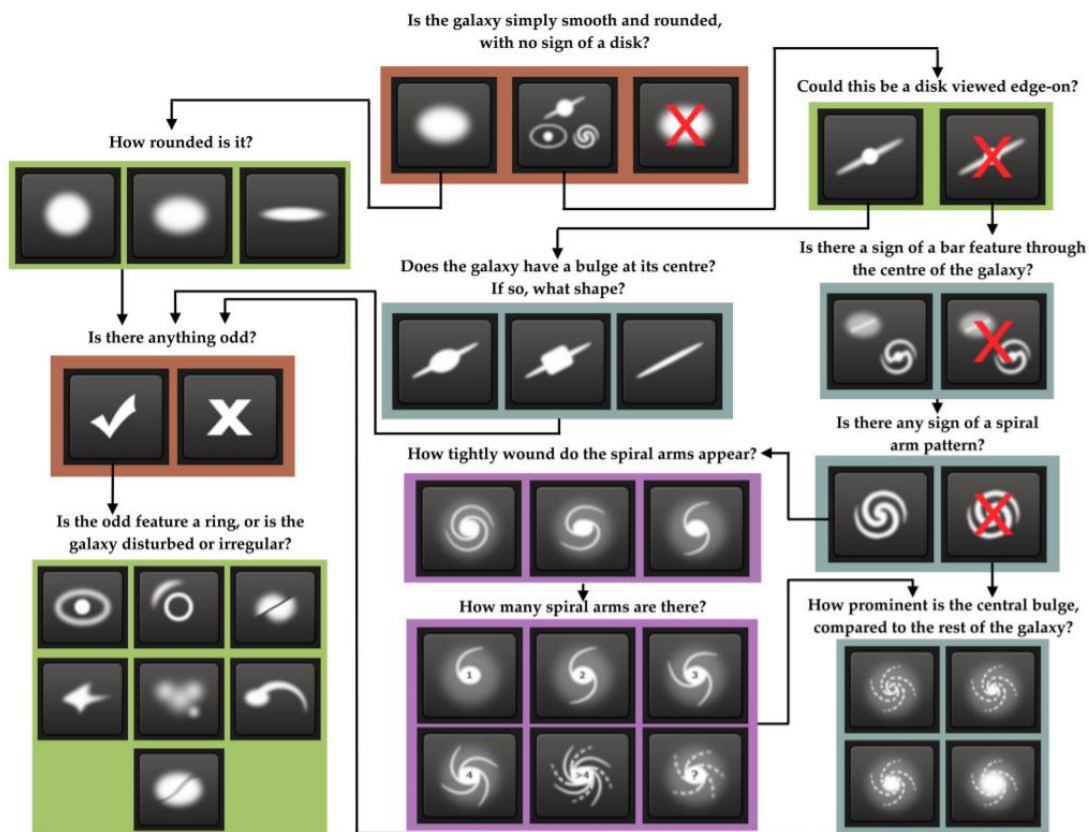


圖 1：Galaxy Zoo 2 星系分類流程 (Willett et al., 2013)

(二) 訓練設備及框架

1、 工作站電腦

(1) 顯示卡：GeForce® RTX 2070 SUPER™

(2) 記憶體：16 GB

2、 作業系統

為了得到更好的相容性以及效能，我使用 Ubuntu 18.04 作為我的作業系統。

3、 神經網路框架、程式語言

(1) 主要程式語言為 Python 3.8.11

(2) 深度學習框架為 PyTorch 1.9.0 + cu102

參、 研究方法或過程

(一) 星系分類介紹

目前最廣泛使用的星系分類方法為哈伯序列 (Hubble sequence)，這是由天文學家愛德溫鮑威爾·哈伯 (Edwin Powell Hubble) 於 1926 年提出，依據星系的型態外觀分為以下 4 類 (Seeds, 2005)：

1、 橢圓星系 (ellipticals)

代號為 E，外形呈正圓形或橢圓形，中心亮，邊緣漸暗。可以再依據

橢圓的橢率細分為 E0 到 E7，橢率的定義為 $e = 1 - \frac{b}{a}$ ，其中 a 為橢圓

的半長軸， b 為橢圓的半短軸，例如 $e = 0.7$ 的橢圓星系標示為 E7。

2、 螺旋星系 (spirals)

這種星系的中央為恆星聚集的核心，外圍則是平坦的盤面以及旋臂。

若核心部分沒有短棒結構為正規的螺旋星系，代號為 S；若核心部分有短棒結構則稱為棒旋星系，代號為 SB。此外，可以再依據旋臂緊密的程度細分為 Sa (SBa)、Sb (SBb)、Sc (SBc)、Sd (SBd)。如果該星系的形狀介於兩種分類之間，可以用 2 個小寫字母註記，例如 Sab，代表該星系的形狀介於 Sa 和 Sb 之間。

3、 透鏡星系 (lenticulars)

於哈伯序列中位於橢圓星系和螺旋星系的交界處，代號為 S0。

4、 不規則星系 (irregular)

這種星系的外形沒有特別規律的結構，可以再細分為兩種，第一種是 Irr I，星系的外觀不對稱，缺乏中央核心或明顯螺旋結構；第二種是 Irr II，星系是平滑的，但是外觀不對稱。

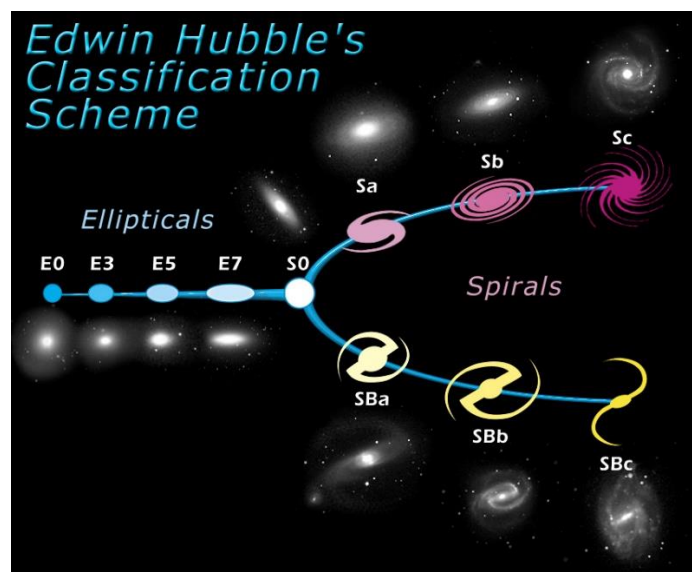


圖 2：哈伯序列示意圖，未包含不規則星系。(圖片來源: NASA/Hubble)

表 1：各種類別的星系照片，標示方法為「星系編號或名稱(類別)」
(圖片來源：EFIGI 及 GZ2)







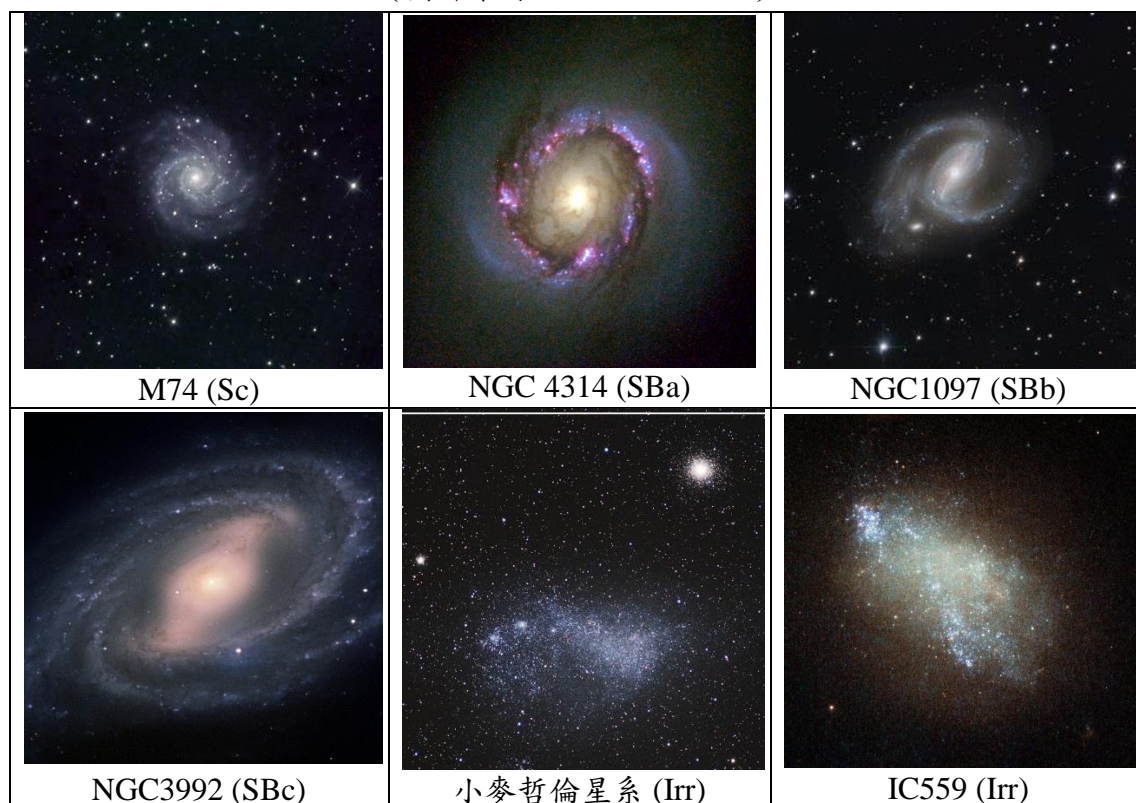
		
M87 (E0)	M49 (E2)	M59 (E5)
		
M85 (S0)	NGC3623 (Sa)	M31 (Sb)

表 1 (續)：各種類別的星系照片，標示方法為「星系編號或名稱(類別)」
(圖片來源：EFIGI 及 GZ2)



(二) 尋找解決方案

1、尋找深度學習框架

目前有許多公司推出不同的深度學習框架，而在眾多框架中我最終選擇了 PyTorch。PyTorch 的前身為 Torch，由於 Torch 是以 Lua 開發而成的，使用者較少，但在 2016 年改用 Python 開發，PyTorch 的使用者急遽增加。PyTorch 的優點在於它十分簡潔，同時支持動態計算圖，使 PyTorch 在撰寫上的靈活度很高。而且 PyTorch 在除錯上就和 Python 一樣簡單，基於以上的原因，我選擇使用 PyTorch。

2、決定使用的神經網路架構

目前最常被使用在圖形辨識的神經網路架構為 CNN，搭配使用的激勵函數 (activation function) 為線性整流函式 (rectified linear unit, ReLU)，損失函數 (loss function) 為交叉熵 (cross entropy)。我決定採用這樣的架構，再試著使用不同的優化器 (optimizer)，例如隨機梯度下降 (stochastic gradient descent, SGD)、自適應矩估計 (adaptive moment estimation, Adam)，並且調整每次最佳化的步幅，也就是學習率 (learning rate)，試著找出辨識效果最好的組合。除此之外，我也試著自行設計輕量的 CNN 模型，並加入自動編碼器來達成半監督式學習 (semi-supervised learning)。

(三) 系統架構與演算流程

1、半監督式機器學習

我使用兩個資料集進行訓練，第一個是 EFIGI，此資料集的標籤是依據哈伯序列而定的，而另一個資料集是 GZ2，此資料集的標籤與哈伯序列不合的，所以我不採納 GZ2 資料的標籤，只使用 GZ2 的照片。EFIGI 資料集的照片只有 4,458 張，而 GZ2 有 243,434 張，此時有標籤的資料是少數，而沒有標籤的資料佔多數，所以我打算使用半監督式機器學習 (Vashishth, Yadav, Bhandari, & Talukdar, 2019)，以下是我採用的兩種方式：

- (1) 先使用 EFIGI 的少量資料訓練一個分類器，接著將 GZ2 的資料加入此分類器，再依據模型對照片做出的自信分數，來判斷是否將此照片加入我的訓練資料，藉此增加訓練資料，以提升模型的正確率。
- (2) 利用自動編碼器 (AutoEncoder) 進行模型的預訓練，並將訓練結束後的參數傳到 CNN 模型裡，因此 CNN 模型不採用隨機初始化參數，希望能使 CNN 有更好的表現。

2、神經網路架構

(1) 簡易神經網路

目前所常用的 CNN 模型是 Yann Lecun 教授於 1998 年提出的 LeNet-5 模型，若以 LeNet-5 模型處理 MNIST 手寫數字影像為例，其模型示意圖如圖 3 所示。主要共有 5 種不同功能的層：

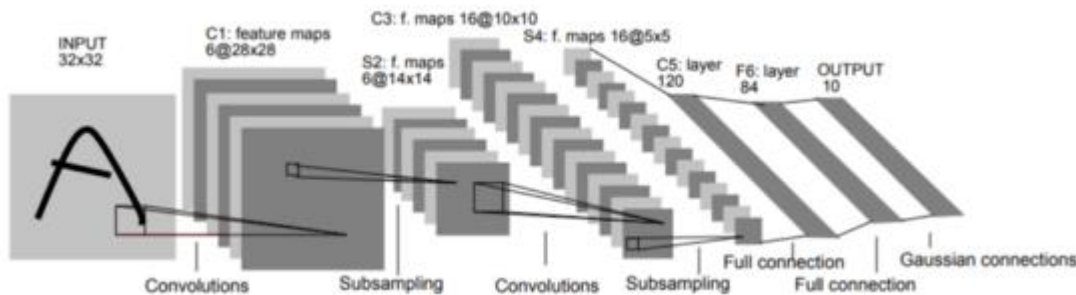


圖 3：LeNet-5 模型結構圖 (Lecun, 1998)

- a. 輸入層 (input layer)：輸入資料。
- b. 卷積層 (convolutional layers)：將輸入資料分成一小塊，常用的大小為 3×3 或 5×5 ，從每一小塊中提取圖形的特徵。
- c. 池化層 (pooling layers)：池化層會將圖片的解析度降低，留下圖形中特徵最明顯的部分，常用的池化有最大池化 (max pooling)、平均池化 (average Pooling)。
- d. 全連接層 (fully connected layers)：將池化層輸出的資料改變維度，接著將每個神經元彼此連接，再將資料傳給 softmax 層。

e. Softmax 層：softmax 又稱為歸一化指數函數，目的是將有限項的離散機率分布梯度歸一化，樣本向量 x 屬於第 j 個分類的機率為

$$P(y = j | x) = \frac{e^{x^T \cdot w_j}}{\sum_1^K e^{x^T \cdot w_k}}$$

表 2 是我使用的卷積神經網路架構，輸入資料的大小是 256 像素 × 256 像素的星系照片，同時使用的星系照片皆包含紅、藍、綠三種顏色，因此輸入層的維度大小是 $255 \times 255 \times 3$ ，卷積核的大小皆為 3×3 ，採用最大池化，池化大小 (pooling size) 為 2×2 ，步幅 (strides) 為 2，同時為了避免過擬合 (overfitting) 的情況發生，我在全連結層加入 dropout 來隨機忽略部分神經元。

表 2：簡易神經網路架構

編號	種類	維度	編號	種類	維度
1	輸入層	$256 \times 256 \times 3$	8	卷積層	$32 \times 32 \times 512$
2	卷積層	$256 \times 256 \times 64$	9	池化層	$16 \times 16 \times 512$
3	池化層	$128 \times 128 \times 64$	10	卷積層	$16 \times 16 \times 512$
4	卷積層	$128 \times 128 \times 128$	11	池化層	$1 \times 1 \times 256$
5	池化層	$64 \times 64 \times 128$	12	全連結層	$1 \times 1 \times 256$
6	卷積層	$64 \times 64 \times 256$	13	全連結層	$1 \times 1 \times 128$
7	池化層	$32 \times 32 \times 256$	14	輸出層	$1 \times 1 \times \text{種類}$

(2) 卷積神經網路 VGG-16

VGG-16 是 2014 年的 ImageNet 圖像分類與定位挑戰賽的亞軍，共有 16 個隱藏層，我將使用 VGG-16 進行以下的部分實驗。(Chen, Papandreou, Kokkinos, Murphy & Yuille, 2014)

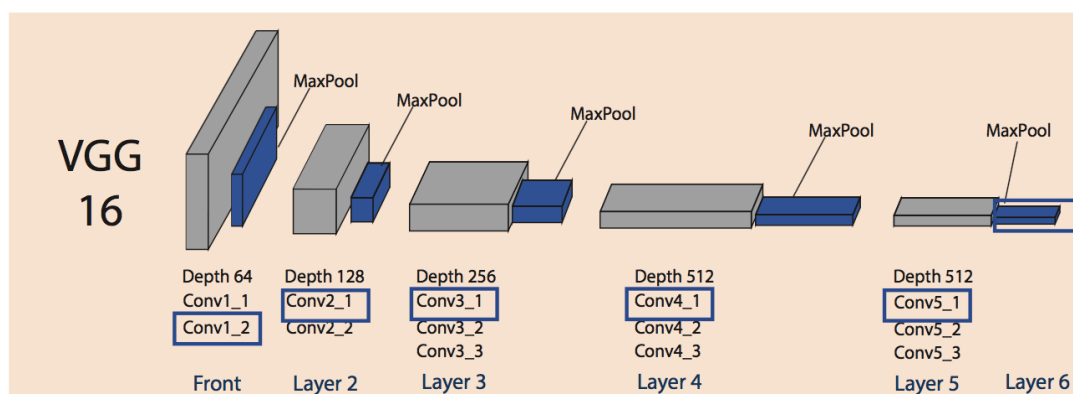


圖 4：VGG-16 主要架構圖 (Berardino, Ballé, Laparra & Simoncelli, 2017)

表 3 是我使用的 VGG-16 架構，我沒有載入 VGG-16 預訓練的權重，只單純使用 VGG-16 的架構，因為原始 VGG-16 的辨別物與星系的外觀相差太大，所以預訓練的權重反而會使我的模型表現不好。輸入資料的大小是 256×256 像素，同時使用的星系照片皆包含紅、藍、綠三種顏色，因此輸入層的維度大小是 $256 \times 256 \times 3$ ，卷積核的大小皆為 3×3 ，採用最大池化，池化大小為 2×2 ，步幅為 2，同時為了避免過擬合的情況發生，我在全連結層加入 dropout 來隨機忽略部分神經元。

表 3：卷積神經網路 VGG-16 架構

編號	種類	維度	編號	種類	維度
1	輸入層	$256 \times 256 \times 3$	12	卷積層	$32 \times 32 \times 512$
2	卷積層	$256 \times 256 \times 64$	13	卷積層	$32 \times 32 \times 512$
3	卷積層	$256 \times 256 \times 64$	14	卷積層	$32 \times 32 \times 512$
4	池化層	$128 \times 128 \times 64$	15	池化層	$16 \times 16 \times 512$
5	卷積層	$128 \times 128 \times 128$	16	卷積層	$16 \times 16 \times 512$
6	卷積層	$128 \times 128 \times 128$	17	卷積層	$16 \times 16 \times 512$
7	池化層	$64 \times 64 \times 128$	18	卷積層	$16 \times 16 \times 512$
8	卷積層	$64 \times 64 \times 256$	19	池化層	$8 \times 8 \times 512$
9	卷積層	$64 \times 64 \times 256$	20	全連結層	$1 \times 1 \times 4096$
10	卷積層	$64 \times 64 \times 256$	21	全連結層	$1 \times 1 \times 4096$
11	池化層	$32 \times 32 \times 256$	22	全連結層	$1 \times 1 \times \text{類別數}$

(3) 自動編碼器 (AutoEncoder)

Hinton 與 Salakhutdinov 於 2006 提出自動編碼器的技術。我使用的自動編碼器分為兩個部分，第一個部分是編碼器 (encoder)，第二個部分是解碼器 (decoder)，表 4 是我使用的編碼器架構，表 5 是我使用的解碼器架構，編碼器會利用卷積層和池化層將原本資料的大小轉成 2 維，解碼器會使解卷積層將 2 維的資料轉成原本圖像的大小。我使用自動編碼器做預訓練，做完自動編碼器後我會建立一個 CNN 模型，這個模型和第一個模型不同，它是以編碼器為起始的網路，最後再加上全連結層。我會將自動編碼器的權重加入到新的 CNN 模型中，這樣可讓模型的初始權重對於星系圖片有較好的判斷能力。

表 4：編碼器架構

編號	種類	維度	編號	種類	維度
1	輸入層	256×256×3	9	池化層	16×16×256
2	卷積層	256×256×32	10	卷積層	16×16×512
3	池化層	128×128×32	11	池化層	8×8×512
4	卷積層	128×128×64	12	卷積層	8×8×1024
5	池化層	64×64×64	13	池化層	4×4×1024
6	卷積層	64×64×128	14	卷積層	4×4×1024
7	池化層	32×32×128	15	輸出層	4×4×1024
8	卷積層	32×32×256			

表 5：解碼器架構

編號	種類	維度
1	輸入層	4×4×1024
2	解卷積層	16×16×512
3	解卷積層	64×64×128
4	解卷積層	128×128×32
5	輸出層	256×256×3

(4) F1-score

在進行機器學習效能的計算的時候不能只使用準確率(Accuracy)做計算，因為只使用準確率當作指標的話可能無法有效的對模型做出衡量，例如:如果有 100 個樣本數，其中有 90 個為正例，有 10 個為反例，模型預測結果判斷出 90 個正確的，其中包含正例與反例，因此此時正確率為 90/100=90%，雖然有 90%的正確率，但是如果今天在樣本數目中全部都預測正例，卻依然得到 90%的正確率，所以今天如果樣本數目不平衡，使用正確率當指標可能沒辦法有效的評估模型。

因此我們使用了 F1-score 當作第二個指標，F1-score 是由 Precision 和 Recall 值計算出來的，其中

$$\text{Precision} = \frac{TP}{TP + FN}$$

$$\text{Recall} = \frac{TP}{TP + FP}$$

由表 6 可以看到 TP 就是答案為真，且預測為真，代表此次預測是對的，FP 就是答案為假，但預測為真，代表此次預測是錯的，FN 就是答案為真，但預測為假，代表此次預測是錯的，TN 就是答案為假，且預測為假，代表此次預測是對的。

$$\text{F1-score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

經由混淆矩陣可以計算出 F1-score。而在我們的研究中會使用多個類別 (multi class) 的判斷，我們使用 Macro-average 的方式將計算出來的結果取平均當作我們最終的指標，藉由 F1-score 可以更有效的判斷模型的效能。(Chicco & Jurman, 2020)

表 6：真假混淆矩陣

	Positive(True answer)	Negative(True answer)
Positive(Predicted)	TP	FP
Negative(Predicted)	FN	TN

(四) 實驗設計

1. 使用簡易卷積神經網路

- (1) 實驗 1：為了初步驗證我的想法，我使用 EFIGI 資料集中數量較多的橢圓 (E)、螺旋 (Sc)、不規則 (I) 3 類照片進行實驗，訓練資料有 2,468 張，測試資料 100 張，學習率為 0.0003，優化器是 Adam，激勵函數是 ReLU，批量大小 (batch size) 為 32。
- (2) 實驗 2：原始 EFIGI 資料集有 4,458 張照片，但有些照片的分類不在哈伯序列中，因此我只挑選標籤為哈伯序列裡的資料，挑選出來的訓練資料有 1,923 張，測試資料有 145 張。將星系分為橢圓 (E)、透鏡 (S0)、螺旋 (Sa、Sb、Sc、SBa、SBb、SBc)，共 8 類，學習率為 0.0003，優化器是 Adam，激勵函數是 ReLU，批量大小為 32。

2. 使用卷積神經網路 VGG-16

- (1) 實驗 3：使用 EFIGI 資料集中數量較多的橢圓 (E)、螺旋 (Sc)、不規則 (I) 3 類照片進行實驗，訓練資料有 2,468 張，測試資料 100 張，學習率為 0.0003，優化器是 Adam，此處使用權值衰減 (weight decay) 來防止過擬合，權值衰減值為 0.0008，激勵函數是 ReLU，批量大小為 32。
- (2) 實驗 4：原始 EFIGI 資料集有 4,458 張照片，但有些照片的分類不在哈伯序列中，因此我只挑選標籤為哈伯序列裡的資料，挑選出來的訓練資料有 1,923 張，測試資料有 145 張。將星系分為橢圓 (E)、透鏡 (S0)、螺旋 (Sa、Sb、Sc、SBa、SBb、SBc)，共 8 類，學習率為 0.0003，權值衰減值為 0.0008，優化器是 Adam，激勵函數是 ReLU，批量大小為 32。

3. 利用無監督式學習進行星系分類
 - (1) 實驗 5：首先分為橢圓星系(E)和螺旋星系(S)以及不規則星系(I) 3 個種類，將 3 個類別的資料輸入訓練好的簡易卷積神經網路，並在維度為 $1 \times 1 \times 128$ 時將參數回傳，傳入 t-distributed stochastic neighbor embedding (t-SNE) 模型裡降維 (dimension reduction)，將它降到 2 維之後用 K-means 演算法分群 (clustering)。(Van der Maaten & Hinton, 2008; Le Cam & Neyman, 1967; Krishna & Murty, 1999))
 - (2) 實驗 6：分為橢圓 (E)、透鏡 (S0)、螺旋 (Sa、Sb、Sc、SBa、SBb、SBc) 8 個種類，將 8 個類別的資料輸入訓練好的簡易卷積神經網路，並在維度為 $1 \times 1 \times 128$ 時將參數回傳，傳入 t-SNE 模型裡降至 2 維，再使用 K-means 演算法來分群。
4. 使用半監督式學習進行 8 種型態的星系分類
 - (1) 實驗 7：我將 GZ2 的大批資料輸入實驗 4 訓練出來的模型裡，此時用此模型對每張的自信機率來判斷是否將此資料加入我的訓練資料裡，假設有一張星系照片輸入進去此模型，而此模型對此照片的判斷是 E 這個類別，接著判斷此模型對此張照片的自信機率，假設此自信機率高過某一個定值，我就將此張照片加入到我訓練 8 種型態的訓練資料集裡。然後用新加入和原本 EFIGI 的照片接著繼續 VGG-16 模型，一直重複此動作，直到此模型無法再從 GZ2 的資料集裡找出自信機率比較高的照片。
 - (2) 實驗 8：在前幾個實驗中，簡易卷積神經網路和 VGG-16 的初始化參數皆是使用隨機初始化的方式。我在這個實驗使用自動編碼器做模型的預訓練，我使用 GZ2 的 243,434 張照片進行自動編碼器的訓練，將預訓練結束後的參數傳到一個新的 CNN 架構裡，而訓練自動編碼器的學習率為 0.0003，優化器是 Adam，激勵函數是 ReLU，批量大小為 32。而這個 CNN 的架構和簡易卷積神經網路不同，是將編碼器的架構移植到這個 CNN 架構上，並在後方加入全連結神經網路。在訓練完自動編碼器之後，將參數傳給 CNN，再使用 EFIGI 和 GZ2 的資料集，訓練資料有 3,186 張(此訓練資料為實驗 7 增加訓練數據後的資料)，測試資料有 145 張，分類橢圓 (E)、透鏡 (S0)、螺旋 (Sa、Sb、Sc、SBa、SBb、SBc) 等 8 類星系，訓練 CNN 的學習率為 0.0003，優化器是 Adam，激勵函數是 ReLU，權值衰減值為 0.01，批量大小為 32。

5. 使用 Resnet18 和 Inception v3 模型進行星系分類

- (1) 實驗 9：Resnet18 和 Inception v3 都是複雜且很深的模型，Resnet18 有 68 層隱藏層，而 Inception v3 有 306 層隱藏層，我們自行設計的 CNN 模型只有 29 層的隱藏層而已，而在 Deep Learning 的研究中某些任務通常模型越深越有比較好的表現，因此我們想要試試這兩種模型來進行星系分類，希望能得到更好的結果，但模型越深代表模型越複雜，因而會加大 overfitting 的機會，我們在 Resnet18 的訓練中的學習率為 0.0003，優化器是 Adam，激勵函數是 ReLU，批量大小為 32。在 Inception v3 的訓練中的學習率為 0.0003，優化器是 Adam，激勵函數是 ReLU，批量大小為 32，兩種模型的訓練資料有 1,923 張，測試資料有 145 張。將星系分為橢圓 (E)、透鏡 (S0)、螺旋 (Sa、Sb、Sc、SBa、SBb、SBc)，共 8 類。

肆、研究結果與討論

(一) 使用簡易卷積神經網路

- 1、**實驗 1**：當只分為 3 種類別時，模型很快就能收斂，測試的平均準確度為 92.8%，橢圓星系 (E) 和螺旋星系 (S) 的測試準確率皆可以達到 95% 以上，F1 score 約為 0.92，不規則星系 (I) 的測試準確率比較低，大概在 83% 上下，我們推斷這是因為不規則星系缺乏較為一致的特徵，所以訓練後的模型無法很有效地辨認出不規則星系。

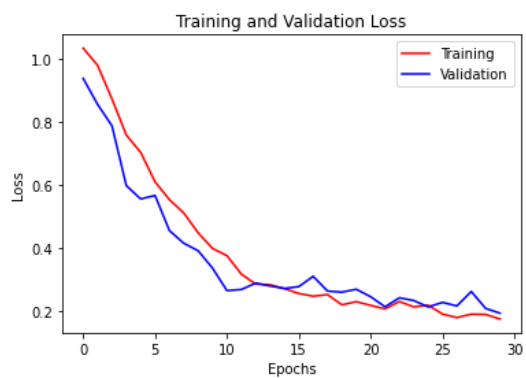


圖 5：實驗 1 損失-epoch 關係圖

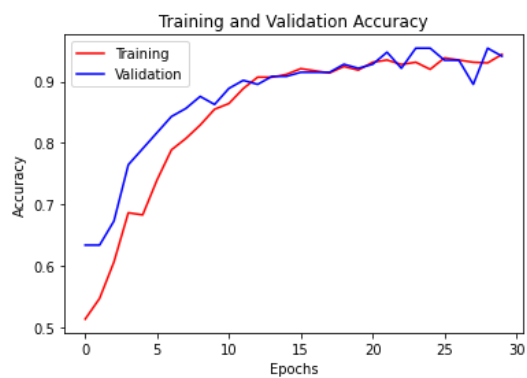


圖 6：實驗 1 準確率-epoch 關係圖

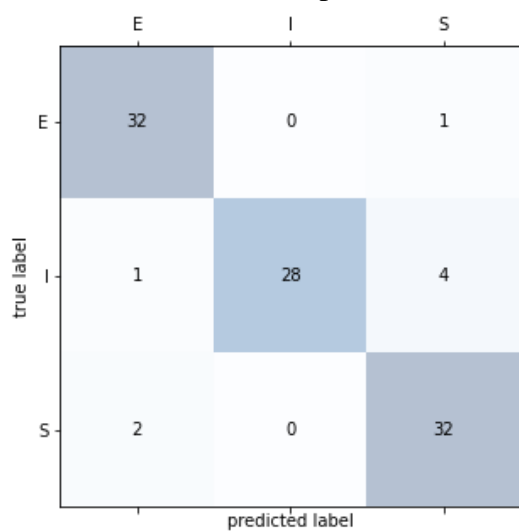


圖 7：實驗 1 混淆矩陣

表 7：實驗 1 各種類星系測試數量及正確率

種類	正確數	總數	準確率
S	32	34	96.9%
E	32	33	96.9%
I	28	33	84.8%

2、**實驗 2**：分為 8 個類別的時候，在第 22 個 epoch 時發過擬合，而且普遍的正確率都不高，平均測試正確率為 36%，F1 score 約為 0.34。我們推測正確率不高的原因，是因為我們的簡易卷積神經網路只有 6 層的卷積層及 2 層的全連結層，整體結構還算太淺，不夠深，使模型無法學習到更複雜的規律，導致正確率偏低。

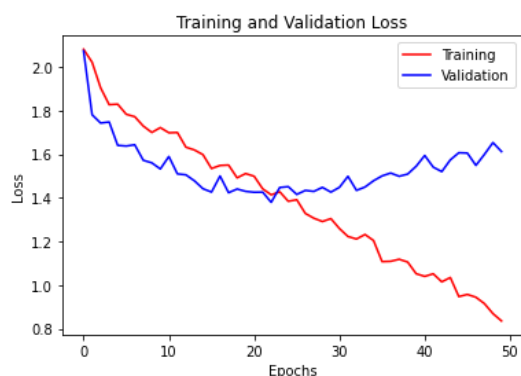


圖 8：實驗 2 損失-epoch 關係圖



圖 9：實驗 2 準確率-epoch 關係圖

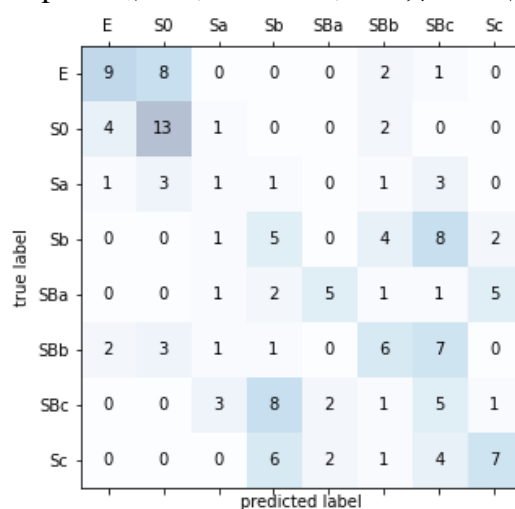


圖 10：實驗 2 混淆矩陣

表 8：實驗 2 各種類星系測試數量及正確率

種類	正確數	總數	準確率	種類	正確數	總數	準確率
E	9	20	45%	SBa	5	10	50%
S0	13	20	65%	SBb	6	20	30%
Sa	1	20	5%	SBc	5	15	33%
Sb	5	20	25%	Sc	7	20	35%

- 3、**實驗 1、2 總結：**使用卷積神經網路建立並訓練模型，如果只分成 3 個種類時效果很好，但當種類較多時分類效果不太理想。由於我使用的是簡易卷積神經網路，在分 8 類的時候可能會因為深度不夠而造成模型學習效率不佳。

(二) 使用卷積神經網路 VGG-16

- 1、**實驗 3：**當我們只分 3 個種類時，模型很快就能收斂，測試的平均準確率為 87.9%，F1 score 約為 0.87，橢圓星系 (E) 和螺旋星系 (S) 的測試準確度比較高，能達到 90% 以上，而不規則星系 (I) 的測試準確率比較低，大約只能有 75% 的正確率，由於不規則星系並無固定型態，所以模型在判斷時可能會有些疑慮，因此造成正確率比較低的現象。



圖 11：實驗 3 損失-epoch 關係圖

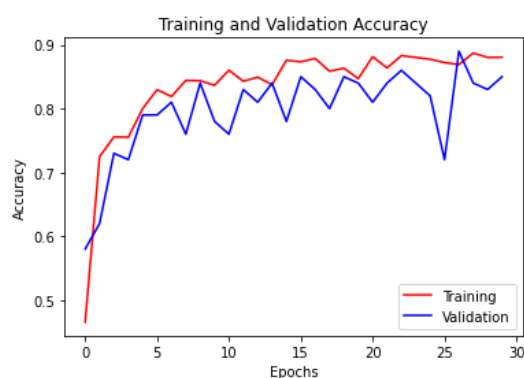


圖 12：實驗 3 準確率-epoch 關係圖

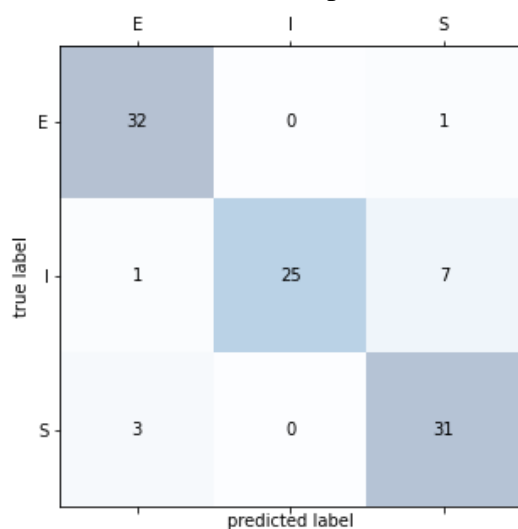


圖 13：實驗 3 混淆矩陣

表 9：實驗 3 各種類星系測試數量及正確率

種類	正確數	總數	準確率
S	31	34	91.1%
E	32	33	96.9%
I	25	33	75.7%

2、**實驗 4**：分為 8 個種類時，我們嘗試增加權值衰減值來避免過擬合，測試平均準確率為 41.25%，F1 score 約為 0.37，橢圓 (E)、透鏡 (S0)、螺旋 (Sc) 這三種的正確率都比較高，因為這三種的外觀型態比較好辨認，而 Sa、Sb、SBa、SBb、SBc 這五種星系的外觀太過於相似，以至於模型無法正確的從樣貌斷出他們的型態。

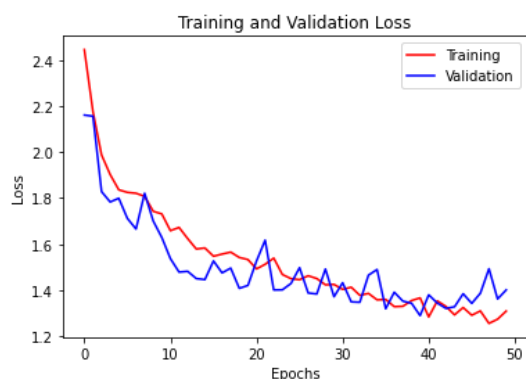


圖 14：實驗 4 損失-epoch 關係圖



圖 15：實驗 4 準確率-epoch 關係圖

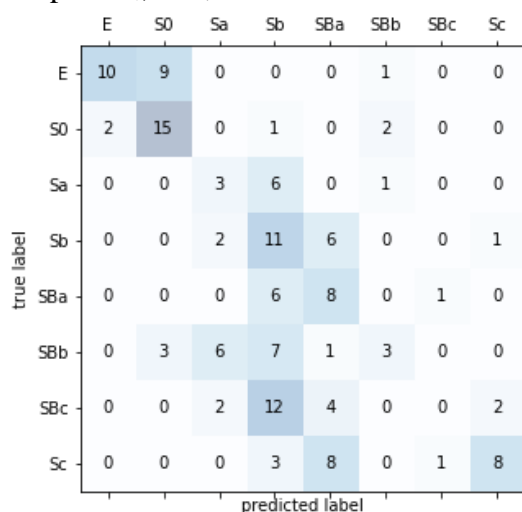


圖 16：實驗 4 混淆矩陣

表 10：實驗 4 各種類星系測試數量及正確率


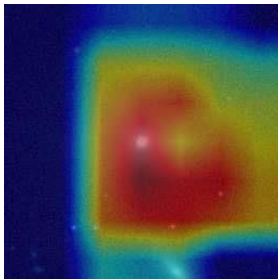

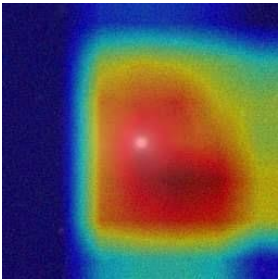
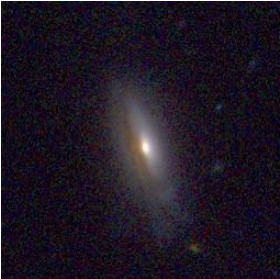
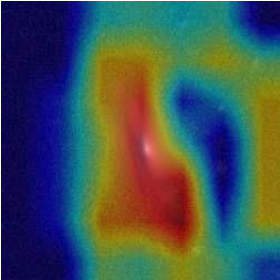

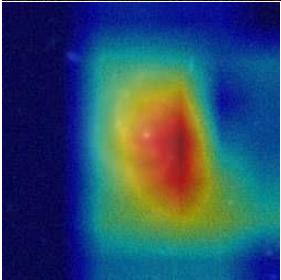
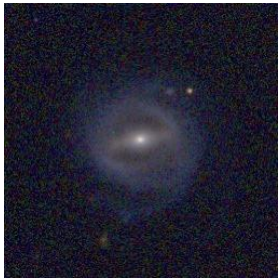
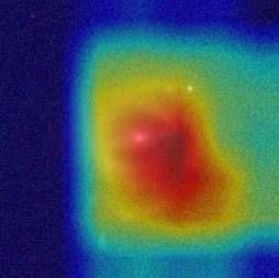

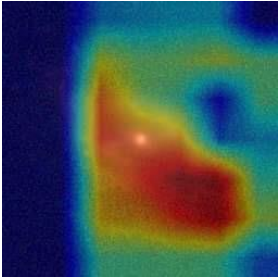
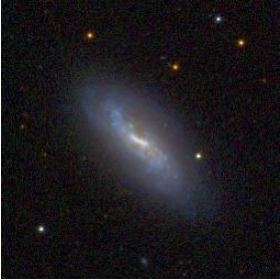
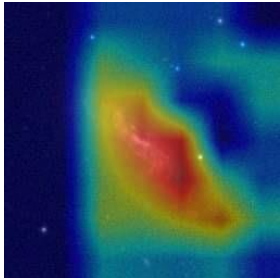

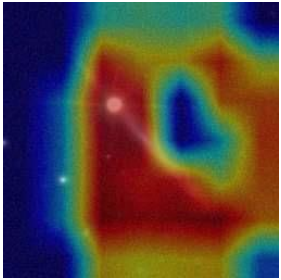
種類	正確數	總數	準確率	種類	正確數	總數	準確率
E	10	20	50%	SBa	8	10	80%
S0	15	20	75%	SBb	3	20	15%
Sa	3	20	15%	SBc	0	15	0%
Sb	11	20	55%	Sc	8	20	40%

同時我為了查看模型對於各種型態的權重分佈圖，這裡使用 Grad-CAM 來進行可解釋人工智慧，Grad-CAM 會依照權重的不同給予不同的顏色分佈，其顏色對上權重大小的對應圖如，越左邊權重越小，越右邊權重越大。(Selvaraju, Cogswell, Das, Vedantam, Parikh & Batra, 2017) 接著我將星系圖片放進 Grad-CAM 模型裡，並將圖片輸出，我將每種類別 (E、S0、Sa、Sb、Sc、SBa、SBb、SBc) 各一張照片進行判斷，輸出照片如表 10。輸出的結果可以看到模型都能準確抓到星系的位置，但對於其輪廓的偵測還有些進步空間。



圖 17：權重-顏色對應圖

表 11：各種類別與 Grad-CAM 輸出對應圖

E  	S0  	Sa  	Sb  
SBa  	SBb  	SBc  	Sc  

3、**實驗 3、4 總結：**在分成三類的時候橢圓 (E)、透鏡 (S0) 的分辨正確率較高，但對於不規則星系 (I) 正確率卻降到 75%。而分成 8 類的時候，整體的正確率相對於實驗 2 增加了許多，同時藉由 Grad-CAM 可以更清楚的了解到我的模型對於星系輪廓的範圍判斷是有一定的能力，且權重最強的紅色正好能包含住各個型態重要的部位，代表我的模型能了解在分辨星系時需要觀察的點是他的外觀型態。由此可以得知在分辨 3 類的時候適合使用簡易卷積神經網路，分成 8 類的時候適合使用卷積神經網路 VGG-16。

(三) 利用無監督式學習進行星系分類

此處使用無監督式學習主要不是希望模型能協助做分類，而是經由 T-SNE 將高維度度的資料降維到低維度，方便我做分析，此處是想分析星系之間的差異，藉由低維度的資料分布可以看出不同星系間的差異。

1、**實驗 5：**此處我將 70 張的 FIGI 的星系圖片做降維，由圖 18 可以看到 t-SNE 清楚的將橢圓 (E)、不規則星系 (I)、螺旋 (S) 做出區分，紅色點為 E，藍色點為 I，黃色點為 S。為了驗證我降維的成效，我使用 K-means 演算法做出分群。由圖 19 可以看到 K-means 演算法清楚的將三類做出分群，圖中的灰點為 K-means 分群的中心點，平均測試正確率為 92.86%。由此可以看出 E、I、S 三種星系圖片中的差異是比較大的，因此驗證使用簡易卷積神經網路以及卷積神經網路 VGG-16 在分類三種類別的正确率比較高，且收斂的比較快速的因素。

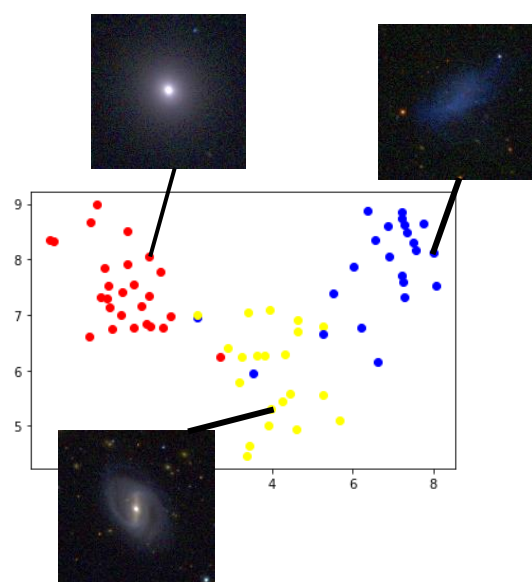


圖 18：三種類別星系圖經由 t-SNE 降維

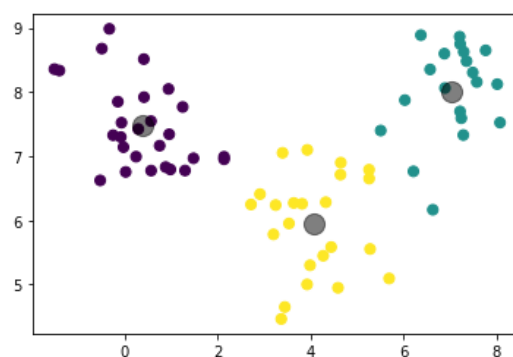


圖 19：三種類別星系圖經由 K-means 分群

2、**實驗 6**：將 80 張 EFIGI 的照片做降維，這 80 張照片包含 (E、S0、Sa、Sb、Sc、SBa、SBb、SBc) 這 8 種。圖 20 的地方我使用 t-SNE 降維之後顯示在二維空間裡，E 為紅色，S0 為藍色，Sa 為黑色，Sb 為橘色，SBa 為綠色，SBb 為粉色，SBc 為紫色，Sc 為灰色。E 和 S0 的外觀較為相似，所以在二維空間的點分布大多在左下方，這也是為何在實驗 2 和實驗 4 中的混淆矩陣裡，E 和 S0 之間容易分辨錯誤。而其他的螺旋星系 (S) 點的分布在中間及右上方，但都應為外觀過於相似，所以點與點之間並無法做出區分，每個點都交錯在一起。我使用 K-means 演算法更能驗證這一點，圖 21 中為 K-means 演算法進行分群的結果，由於點和點之間都相互交錯，所以分群之後的效果非常差，平均測試正確率只有 11.43%，由此處可以看出造成實驗 2 和實驗 4 的最終結果正確率比較很可能就是星系間外觀太過相似，導致模型在進行學習時無法準確抓住特徵點。

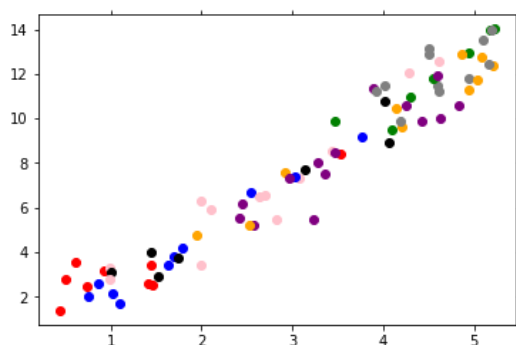


圖 20：8 類別星系圖經由 t-SNE 降維

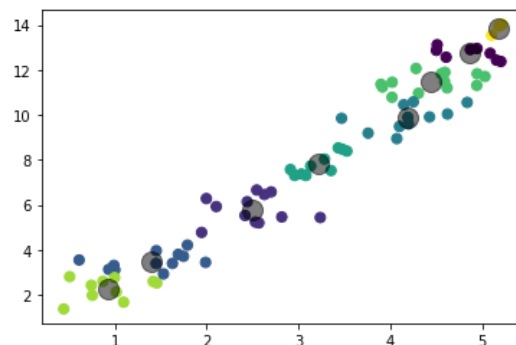


圖 21：8 別星系圖經由 K-means 分群

3、**實驗 5、6 總結**：在此部分利用 t-SNE 降維以及 K-means 分群演算法進行分析，分析在三種類別的分類以及 8 種類別的分類正確率落差極大的原因，在進行分類問題時如果外觀過於相似，且無較具代表性的特徵點的話，會讓模型不知道如何區別不同種的類別。

(四) 使用半監督式學習進行 8 種型態的星系分類

1、**實驗 7**：我們在此實驗中使用半監督學習，並成功將訓練資料數量從 1,923 張成長到 3,186 張，原始各類別資料數量如表 12，經由半監督學習後增加後的資料數量如表 13。圖 22 是我們利用半監督式學習所增加訓練資料的訓練流程，這邊我們使用的半監督學習方法是將 GZ2 的照片丟入訓練好的 VGG-16 架構的 CNN 模型，再利用模型對每張圖片的自信分數來決定是否將此照片加入訓練資料裡，表 14 是利用自信分數篩選出來的照片。訂定自信分數的方式是使用實驗 4 訓練好的模型，將 EFIGI 的 100 張資料輸入，輸入之後查看當模型判斷正確時它的自信分數是多少，經由觀察發現當自信分數高於 30% 的時候模型幾乎判

斷正確，以防萬一我們設 35% 當作我們的門檻，當模型對 GZ2 的照片自信分數高過 35%，我們就將此張照片歸類為正確的判別，並將此照片加入訓練資料集，同時我們會將此照片從 GZ2 中移除。加入原來的訓練資料集後，我們又繼續訓練剛才 VGG-16 架構的 CNN 模型，訓練完之後再將 GZ2 的資料傳入並判斷自信分數，我們重複此動作 5 次，因為在做第二次的判斷之後增加的照片都低於 10 張，我們猜測這是因為其他的照片的相似度太高，導致我們的模型無法有足夠的自信分數來做判斷。

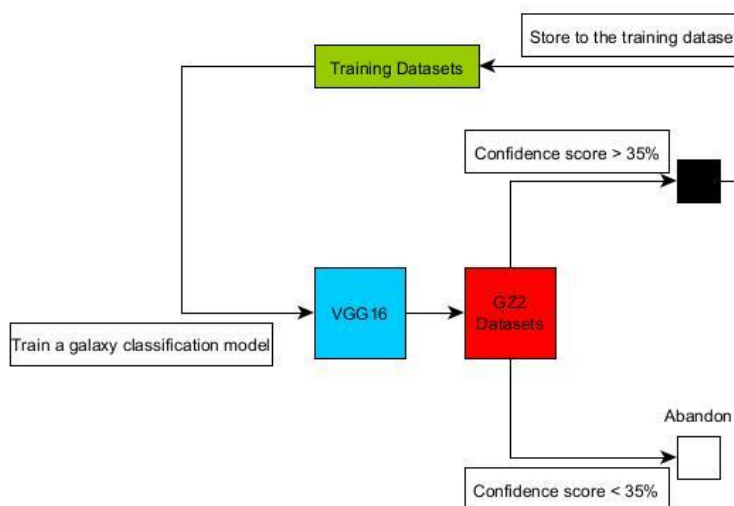


圖 22: 資料迭代增加流程圖

表 12：實驗 8 各種類別原始訓練資料數量

E	S0	Sa	Sb
243	336	161	257
Sc	SBa	SBb	SBc
563	84	155	124

表 13：實驗 8 各種類別增加資料後訓練資料數量

E	S0	Sa	Sb
246	342	162	314
Sc	SBa	SBb	SBc
1167	94	173	688

表 14：實驗 8 透過自信分數篩選出來的照片範例

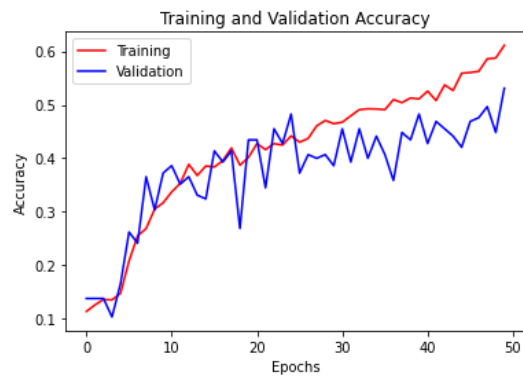
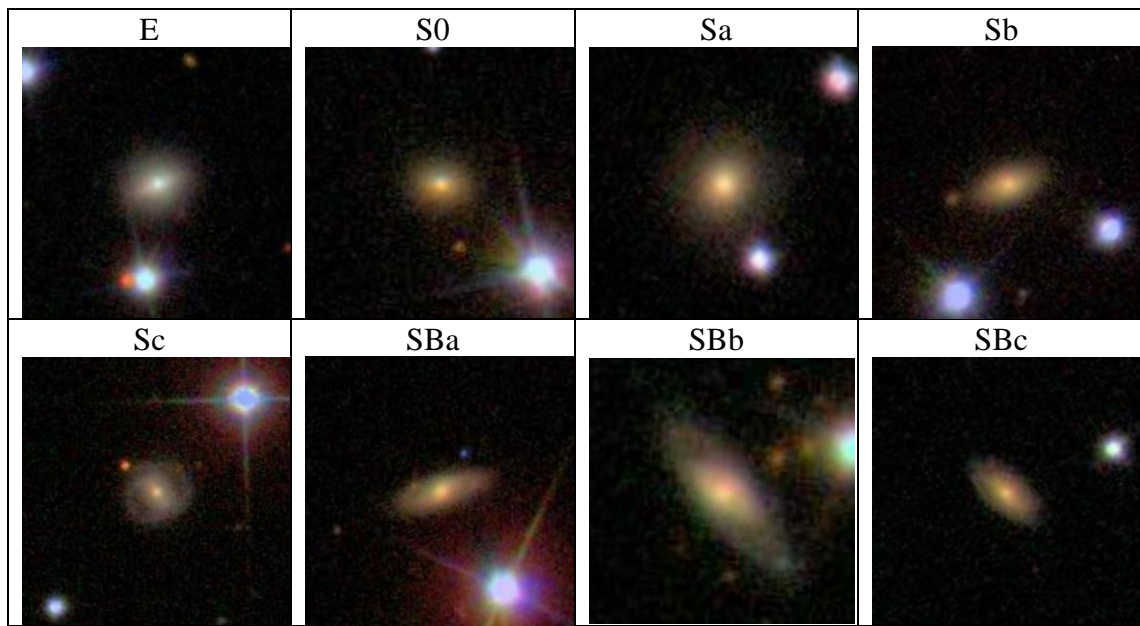


圖 22：實驗 7 損失-epoch 關係圖

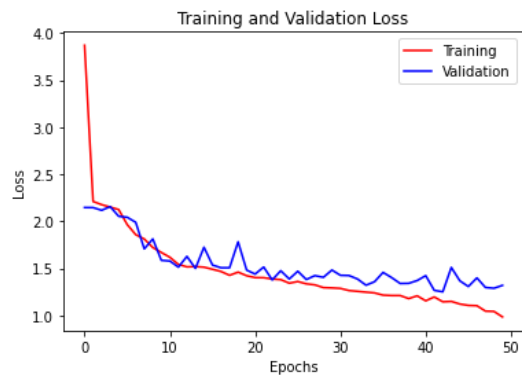


圖 23：實驗 7 正確率-epoch 關係圖

	E	S0	Sa	Sb	SBa	SBb	SBc	Sc
E	18	2	0	0	0	0	0	0
S0	3	10	0	1	0	6	0	0
Sa	0	2	2	3	1	1	1	0
Sb	0	0	0	11	2	0	2	5
SBa	0	0	0	0	1	1	0	13
SBb	0	1	0	0	2	14	2	1
SBc	0	0	3	3	3	2	4	5
Sc	0	0	0	0	1	0	2	17
	predicted label							

圖 24：實驗 7 混淆矩陣

表 15：實驗 7 各種類星系測試數量及正確率

種類	正確數	總數	準確率	種類	正確數	總數	準確率
E	18	20	90%	SBa	1	10	10%
S0	10	20	50%	SBb	14	20	70%
Sa	2	20	10%	SBc	4	15	26%
Sb	11	20	55%	Sc	17	20	85%

做完半監督學習後我們的平均測試正確率可以達到 49.5%，F1 score 約為 0.48，在橢圓星系 (E) 的型態部分甚至可以達到 90% 的正確率，同時相似度極高的螺旋星系的 Sb、SBb、Sc 的正確率都大幅提升，除了數據上的增加，為了避免過擬和，我們同時將權值衰減增加到 0.01，使用 Adam 當優化器，學習率為 0.0003，批量大小為 32。

- 2、**實驗 8：**我們創建兩個神經網路，一個是自動編碼器，架構如表 4 和表 5，另一個是基於編碼器架構的 CNN 模型，接著再接全連結層，架構如表 16，圖 26 是我們的訓練流程，首先我們將 GZ2 的資料傳入自動編碼器裡，訓練完此模型之後我們使用 deepcopy 將 Encoder 最後一層的參數傳到 CNN 模型的編碼器裡，這麼一來 CNN 模型就不用採用權重隨機初始化，同時我們沿用實驗 7 以增加資料的訓練資料集來進行訓練。運用訓練資料數量地增加，且利用自動編碼器的預訓練 (pretrain)，並進行微調 (Fine-tuning)，我們最終可以得到目前最佳的結果，平均測試正確率可以達到 54.12%，F1-score 可以到達 0.51。

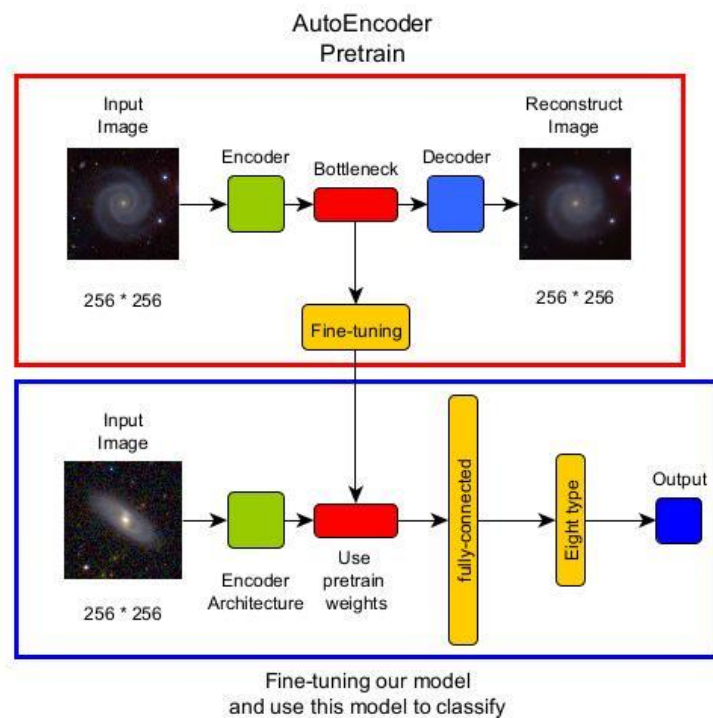


圖 26: 利用自動編碼器當預訓練並微調之訓練流程圖

表 16：接取自動編碼器權重的 CNN 模型架構

編號	種類	維度	編號	種類	維度
1	輸入層	256×256×3	10	卷積層	16×16×512
2	卷積層	256×256×32	11	池化層	8×8×512
3	池化層	128×128×32	12	卷積層	8×8×1024
4	卷積層	128×128×64	13	池化層	4×4×1024
5	池化層	64×64×64	14	卷積層	4×4×1024
6	卷積層	64×64×128	15	全連結層	1×1×1024
7	池化層	32×32×128	16	全連結層	1×1×256
8	卷積層	32×32×256	17	輸出層	1×1×8
9	池化層	16×16×256			

表 17：實驗 8 各種類星系測試數量及正確率

種類	正確數	總數	準確率	種類	正確數	總數	準確率
E	16	20	80%	SBa	3	10	30%
S0	8	20	40%	SBb	17	20	85%
Sa	0	20	0%	SBc	5	15	33%
Sb	16	20	80%	Sc	17	20	85%

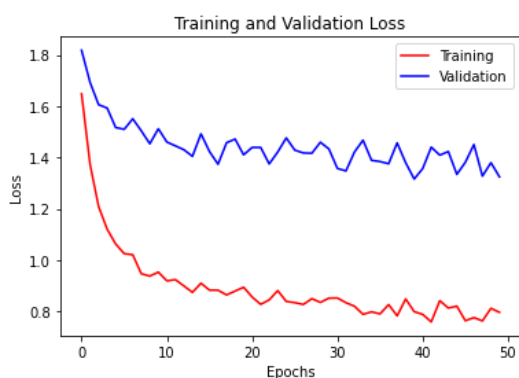


圖 25：實驗 8 損失-epoch 關係圖



圖 26：實驗 8 正確率-epoch 關係圖

	E	S0	Sa	Sb	SBa	SBb	SBc	Sc
true label \ predicted label	E	S0	Sa	Sb	SBa	SBb	SBc	Sc
E	16	4	0	0	0	0	0	0
S0	4	8	0	0	0	7	1	0
Sa	0	3	0	5	0	1	0	1
Sb	0	0	0	16	1	0	1	2
SBa	0	0	0	2	3	0	0	10
SBb	0	1	0	1	1	17	0	0
SBc	0	0	0	7	1	2	5	5
Sc	0	0	0	2	1	0	0	17

圖 27：實驗 8 混淆矩陣

在圖 27 和圖 28 中可以看到經過預訓練的模型相對於未進行預訓練的模型可以在一開始有較高的正確率以及較低的 loss，且模型的收斂速度也相對較快，但在這一個 CNN 模型中的參數非常多，所以發生了 overfitting 的狀況，我們也嘗試利用此 CNN 模型進行實驗 7 的訓練數據增加的半監督式方法，但是當我們拿增加過後的數據再下去訓練竟發生更嚴重的 overfitting，因此最後我們就不再使用實驗 7 的方法繼續迴圈式的訓練下去了。我們對於利用自動編碼器會有更好的表現提出一個解釋，在圖 20 裡我們利用 CNN 最後一層的輸出對圖片進行 t-SNE 降維，可以看到所有的數據點都擠在一起，未能有較好的分布，且利用 K-means 演算法僅能得到 11.43% 的正確率，但我們將自動編碼器的編碼器最後一層的參數做扁平(flatten)並對它進行降維，可以得到圖 30 的結果，E 為紅色，S0 為藍色，Sa 為黑色，Sb 為橘色，SBa 為綠色，SBb 為粉色，SBc 為紫色，Sc 為灰色。可以看到數據點並不是全部都擠在同一塊，但因為星系圖片實在過於相近，數據點還是呈現較無規律的分布，但是我們利用 K-means 演算法可以得到 22.86% 的正確率，相對於利用 CNN 最後一層的輸出對圖片進行降維，利用已經訓練過的編碼器我們可以增加 11% 的正確率，因此可以判斷自動編碼器可以對星系有較好的區分能力，若將此能力複製到我們的 CNN 模型，就可以得到較高的正確率。

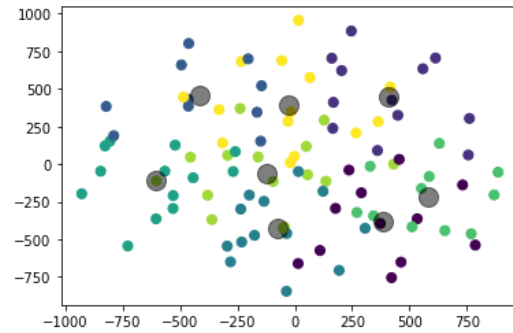
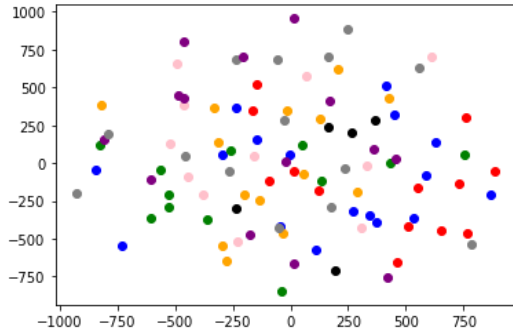


圖 30: 利用編碼器最後一層權重做降維

圖 31: 將圖 30 的結果用 K-means 計算

3、實驗 7、8 總結

- (1) 在做分類問題時，如果訓練資料越多分類的效果就會越好，但是 FIGI 的資料集給予每個類別的星系照片都小於 600 張，甚至有星系的圖片只有 80 多張，像這樣訓練資料太少會導致模型很難學到星系間的規律。因此我們使用半監督學習的方式增加訓練數據量，藉由模型的自信分數來直接斷定此判斷是否正確，當判斷完的 GZ2 資料全部加入訓練資料後再繼續訓練模型，訓練完模型再繼續使用自信分數增加訓練數據，經由這樣的迴圈式訓練最終我們增加了 1,263 張訓練資料，同時因為數據量增加，我們的正確率也提升到了 49.5%。
- (2) 利用圖片的增加、以及自動編碼器的預訓練我們可以得到最佳的結果，不僅模型收斂迅速，同時又有較高的正確率，平均測試正確率可以達到 54.12%，F1-score 可以達到 0.51。

(五) 使用 Resnet18 和 Inception v3 模型進行星系分類

- 1、**實驗 9：**我們使用 Resnet18 和 Inception v3 進行星系分類，我們自行設計的模型都是比較淺且簡易的模型，而 Resnet18 和 Inception v3 都是很深的模型，因此我們想測試當模型變深時，是否可以增加辨識星系的正確率，但因為這兩種模型太深，因此很容易造成 overfitting，我們嘗試解決問題，試著增加權值衰減值，但又會造成 underfitting，而導致模型辨識率更低，因此我們最終並未加入權值衰減，我們的 Resnet18 的平均測試正確率為 24.5，F1 score 為 0.22，且在第 3 個 epoch 發生了 overfitting，Inception v3 平均正確率為 33.25，F1 score 為 0.31，且在地 6 個 epoch 發生了 overfitting。而我們也想把實驗 7 的半監督式方法增加訓練資料來應用在這兩種模型上，但是實驗 7 利用自信分數來增加訓練資料的方法需要建立在能訂出自信分數標準的情況下，但我們訓練的 Resnet18 和 Inception v3 不管有沒有判斷正確，每種星系的判斷自信分數都不高於 20%，且就算有接近 20% 的正確率都

有可能判斷錯誤，因此我們沒辦法將此 Resnet18 和 Inception v3 應用在星系分類上。

種類	正確數	總數	準確率	種類	正確數	總數	準確率
E	7	20	35%	SBa	1	10	10%
S0	9	20	45%	SBb	2	20	10%
Sa	1	20	5%	SBc	4	15	26%
Sb	1	20	5%	Sc	12	20	60%

表 18：實驗 9 的 Resnet 各種類星系測試數量及正確率

種類	正確數	總數	準確率	種類	正確數	總數	準確率
E	13	20	65%	SBa	2	10	20%
S0	2	20	10%	SBb	6	20	30%
Sa	2	20	10%	SBc	7	15	46%
Sb	5	20	25%	Sc	12	20	60%

表 19：實驗 9 的 Inception v3 各種類星系測試數量及正確率



圖 32：實驗 9 的 Resnet 損失-epoch 關係圖

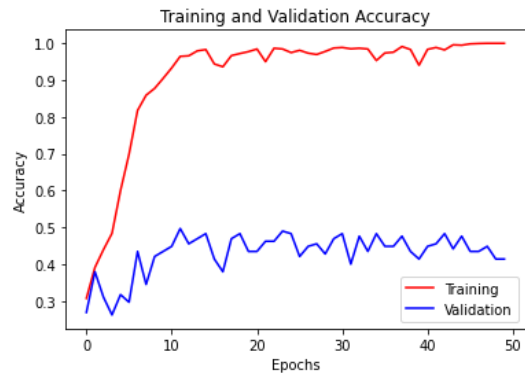


圖 33：實驗 9 的 Resnet 正確率-epoch 關係圖

	E	S0	Sa	Sb	SBa	SBb	SBc	Sc
E	7	10	1	0	0	2	0	0
S0	7	9	1	0	0	2	1	0
Sa	1	1	1	1	0	3	2	1
Sb	0	0	0	1	0	5	5	9
SBa	0	1	1	3	1	0	0	9
SBb	3	7	0	2	1	2	3	2
SBc	0	2	0	5	0	4	4	5
Sc	0	0	0	2	0	0	6	12

圖 34：實驗 9 的 Resnet 混淆矩陣

	E	S0	Sa	Sb	SBa	SBb	SBc	Sc
E	13	5	1	0	0	1	0	0
S0	14	2	0	0	0	3	1	0
Sa	1	0	2	1	0	2	3	1
Sb	0	0	0	5	1	0	10	4
SBa	0	0	0	0	2	3	3	7
SBb	2	2	1	1	0	6	7	1
SBc	0	0	4	2	0	1	7	6
Sc	0	0	0	2	1	1	4	12

圖 35：實驗 9 的 Inception v3 混淆矩陣

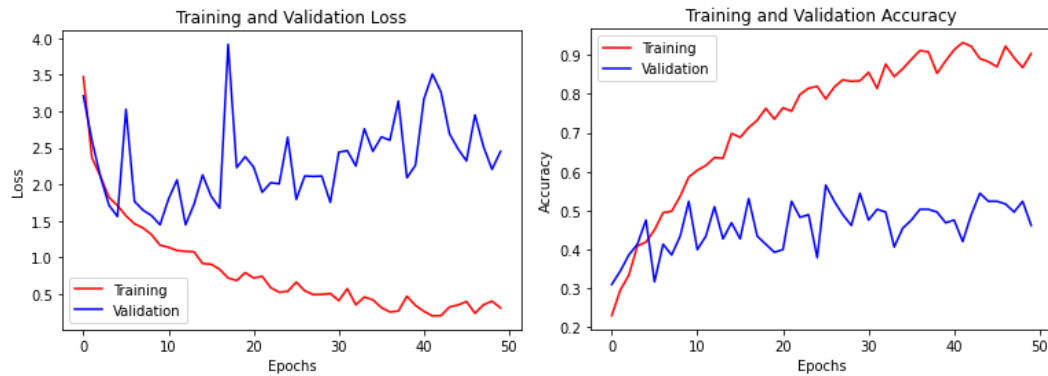


圖 36: 實驗 9 的 Inception v3 損失-epoch 關係圖 圖 37: 實驗 9 的 Inception v3 正確率-epoch 關係圖

伍、結論與未來展望

(一) 結論

根據我的研究結果得到以下的結論：

- 1、如果只是要分辨橢圓 (E)、螺旋 (S)、不規則星系 (I)，不論是使用簡易卷積神經網路或是 VGG-16，模型的正確率都能到達 80% 以上，而使用簡易卷積神經網路的效果會略優於 VGG-16。
- 2、最適合用於 8 類星系分類的模型為：採用基於自動編碼器的編碼器並接上全連結層的架構，且此模型需經過自動編碼器的預訓練，自動編碼器的學習率為 0.0003，使用 Adam 當作優化器，批量大小為 32，訓練時使用 GZ2 的 243,434 張照片，完成預訓練之後將參數傳給 CNN 模型，訓練 CNN 模型的學習率為 0.0003，使用 Adam 當作優化器，權值衰減值為 0.01，批量大小為 32，訓練時使用 3,186 張照片，包含 EFIGI 資料集以及 GZ2 資料集，雖然訓練資料並不多，但是最後訓練出來的模型能有 54.12% 的正確率和 0.51 的 F1 score，比沒做半監督式機器學習的模型高出 12.87% 和 0.14 的 F1-score，同時橢圓 (E)、Sb、SBb、Sc 都有高於 80% 的正確率。

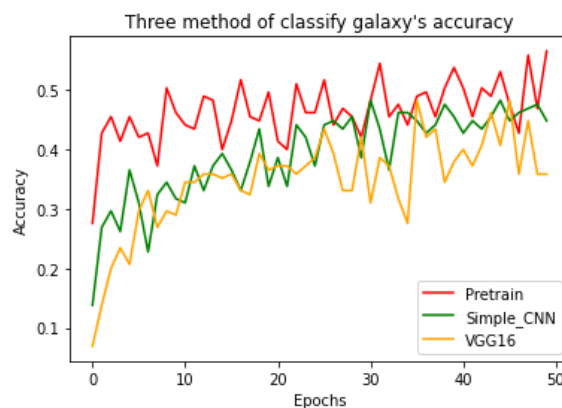


圖 32: 比對使用已做預訓練和資料增加的模型及

使用簡易 CNN 模型和使用 VGG-16 在各 Epoch 的表現

- 3、在實驗 9 中我們可以看到模型架構越複雜且越深不一定會在星系辨識上得到較高的正確率，模型架構越深反而會很容易造成 overfitting，且在這樣的情況下，模型無法對於判斷星系時給予較高的自信分數，因此在增加訓練資料的半監督學習方法就不適用於這樣的情況。

(二) 未來展望

1、短期目標

- (1) 能為目標判斷星系照片做出更準確的範圍，減少背景星系的干擾，以及做出星系的特徵點提取。
- (2) 找出提高螺旋星系 (Sa、Sb、Sc、SBa、SBb、SBc)判斷的正確率。

2、長期目標

- (1) 增加圖形化使用者介面，使用者只需要將想要判斷種類的星系照片輸入，並按下執行之後即可輸出星系類別。
- (2) 能大量的輸入星系的照片進入圖形化使用者介面，軟體自動使用表格的方式將星系類別的判讀結果列成表格，同時表格裡附上模型對此星系照片的自信機率，此自信機率可以給使用者另一種的參考，藉由自信機率使用者可以知道那些判斷的可信度較高，且自信機率低的照片可以當作需要人工篩選的指標。

陸、 參考文獻

1. Baillard, A., Bertin, E., De Lapparent, V., Fouqué, P., Arnouts, S., Mellier, Y., ... & Tasca, L. (2011). The EFIGI catalogue of 4458 nearby galaxies with detailed morphology. *Astronomy & Astrophysics*, 532, A74.
2. Willett, K. W., Lintott, C. J., Bamford, S. P., Masters, K. L., Simmons, B. D., Casteels, K. R., ... & Thomas, D. (2013). Galaxy Zoo 2: detailed morphological classifications for 304 122 galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, 435(4), 2835-2860.
3. Seeds, M. A. (2005). *Foundations of astronomy*. Belmont, CA: Brooks/Cole-Thomson Learning.
4. Vashishth, S., Yadav, P., Bhandari, M., & Talukdar, P. (2019, April). Confidence-based graph convolutional networks for semi-supervised learning. In *The 22nd International Conference on Artificial Intelligence and Statistics* (pp. 1792-1801). PMLR.
5. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4), 541-551.
6. Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.
7. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786), 504-507.
8. Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1), 1-13.
9. Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).
10. Le Cam, L. M., & Neyman, J. (Eds.). (1967). *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 3). Univ of California Press.
11. Krishna, K., & Murty, M. N. (1999). Genetic K-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3), 433-439.
12. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based

- localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618-626).
13. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).
 14. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).