東海大學資訊工程學系研究所

碩士論文

指導教授：陳隆彬博士

適用於行動裝置的
深度學習影像辨識應用

Applicable of the mobile device with deep learning of image recognition applications

研究生：粘昱薪

中華民國 一〇六 年 七 月

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 粘 昱 薪 所提之論文

適用於行動裝置的深度學習影像辨識應用

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

召 集 人 _____ 簽章

委 員 _____

_____

_____

指 導 教 授 _____ 簽章

中華民國 106 年 6 月 28 日

摘　　要

由於行動裝置的普遍，具備高維度訊息處理能力的手機端就成為智能服務的第一線。並且近幾年來深度學習處理高維度訊息應用在圖像識別上有重大的進步，本文以醫院為背景，以電腦視覺領域為研究，欲發展一套通用於各個場景的地點辨識系統，對於佔地不小以及多樓層的醫院，佈署各式定位設備需要不少成本，因此本文提出以深度學習訓練圖片資料庫建立場景地圖，研究使用神經網路及卷積神經網路的分類模型，再透過行動裝置影像進行辨識地點，能節省設備成本，並能方便使用者得知自己目前的所在地。

關鍵詞：機器學習、影像辨識、卷積神經網路。

ABSTRACT

As the mobile device is universal, with high-dimensional information processing capabilities of the mobile phone side has become the first line of intelligent services. And in recent years the Deep learning to deal with high-dimensional information applications in image recognition has made huge progress.In this paper, the hospital as the background, to the field of computer vision for the study. To use the Image recognition to develop a location identification system which can be applicable to any scene.For the area is not small and multi-level hospital. The deployment of various positioning devices requires a lot of cost. Therefore, this paper proposes to use the Deep learnign training the picture database. Research the classification model using the neural network and the convolution neural network, and then identify the location through the mobile device. Can save equipment costs, and can facilitate the user to know their current location.

Keywords: Machine learning, Image recognition, Convolution neural network.

目錄

圖目錄

表目錄

1. 介紹

自從 2016 年 AlphaGo 電腦圍棋程式擊敗職業棋士，AlphaGo 所使用的深度學習技術引起了各界的關注。事實上，深度學習技術已廣泛應用在語音助理和人臉辨識等各個領域。iPhone 的 Siri 語音助理可以將聲音訊號辨識成文字；Facebook 的相片好友人臉辨識等等，用的都是深度學習的技術。

深度學習是機器學習 (machine learning) 的一種方法，1950 年代，當時科學家仿造人類大腦的運作方式，提出神經元模型的感知機，這是最簡單也是最早的類神經模型，感知機通常被拿來做分類器（Classifier）使用。早期類神經網路的理論仍不成熟，因此類神經網路並沒有受到很大的重視。下圖為一個類神經元的模型：

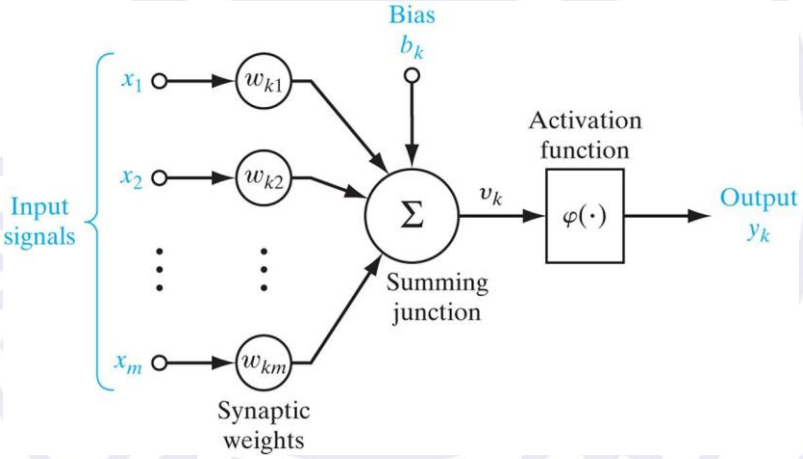

圖 1-1 類神經元模型

- x：稱為神經元的輸入（input）
- W：稱為鍵結值（weights）
- b：稱為閥值（bias），有偏移的效果
- S：稱為加法單元（summation），此部分是將每一個輸入與鍵結值相乘後做一加總的動作。
- φ(。)：稱之為活化函數（activation function），通常是非線性函數，有數種不同的型式，其目的是將 S 的值做映射得到所需要的輸出。
- Y：稱之為輸出（output），亦即我們所需要的結果。

深度學習採用的類神經網路稱為卷積網路 (convolutional neural networks，

CNN)。卷積網路是一種多層級的神經網路，能藉由許多不同層級的特徵處理，有效的辨別語音和圖像特徵，讓使用者能迅速取得辨識後的結果，由實驗結果顯示，CNN 的深層架構確實能夠學習出影像的關鍵特徵，影像分類的準確度高於一般神經網路。由於 CNN 卷積網路的參數(parameters)數量高達數萬至數十萬個，要對這麼多的參數來求出最佳解，需要花很多時間。因此，高速深度學習的領域現在受到很大關注。例如 NVIDIA 公司運用 GPU 加速運算來訓練 CNN 的影像、筆跡和聲音辨識。

由於手持裝置的普遍，具備高維度訊息處理的手機端就成為智能服務的第一線。使用者運用手機的麥克風和攝像頭來取得即時的語音和影像，再利用深度學習技術，進行影像和語音資料的分析，最後再透過雲端服務，進行智能服務，例如醫療資訊服務。然而巨量的訊息串流以及較長的計算時間和電量的限制，深度學習人工智能是耗能的。本文的目標是在盡可能少的硬體設備下，使用一般現有的裝置功能，利用影像辨識，來達到在室內確認地點的目的，並且能通用於各個場景的地點辨識系統。行動裝置的硬體一直在進步，愈來愈快的處理速度及更好的裝置效能，致使我們能在行動裝置上作業的功能愈來愈多，如果能使用行動裝置，加上一般就有的 PC 做為主機，就能利用深度學習結合影像辨識，不用加裝多餘的感測器，進而節省設備成本，只需有一台行動裝置，和一台可以進行機器學習訓練的主機，就能由個人獨自完成場景辨識系統。本文將以醫院地點辨識為應用，在兼顧準確度的情形下，研究不同圖像解析度，和研究使用神經網路或卷積神經經網路，對於分析圖像的精確度的差異，以及在這個訓練模型下，訓練次數是否會影響圖像分析的精確度。

2. 背景知識與技術

為什麼計算機可以學會圖像識別？圖像識別是開發機器學習的一項重要任務，因為視覺對於我們是最重要的感知外界的能力，這個能力的重要性還在其它感覺之上。

對於我們來說簡單到是一項本能的能力，對於計算機卻是困難重重，圖像識別就是讓計算機自己去完成這樣的過程，我們給計算機提供演算法模型，讓計算機從經驗中去學習，就像我們人類做的那樣。

2.1.機器學習

機器學習是一門研究人工智慧的科學領域，研究計算機如何在經驗學習中改善演算法的效能。機器學習是對能通過經驗自動改進的電腦演算法的研究。

機器學習是用資料或以往的經驗，做為最佳化電腦程式的效能標準。實現機器學習的演算方法有多種，可以分成下面幾種類別：

(1) 監督學習[2](supervised learning)從給定的訓練資料集中學習出一個函式，當新的資料到來時，可以根據這個函式預測結果。我們給計算機提供貓和狗的大量圖片，並告訴計算機哪些是狗，哪些是貓，然後再讓計算機學習辨識狗和貓，計算機透過我們預先給定的標籤來學習。

本文用到的辨識也是屬於監督學習。

圖 2-1 .監督學習

(2) 非監督學習(unsupervised learning)，訓練集沒有人為標註的結果，我們一樣提供大量圖片給計算機，但是沒有給定哪些圖片是狗，哪些是貓，讓計

算機自行學習如何判斷 2 種圖片的不同，進而總結出一種規律，來判斷 2 種圖的不同點。

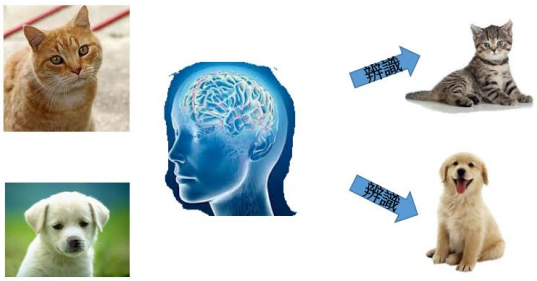

圖 2-2　非監督學習

(3)　半監督學習(semi-supervised learning)介於監督學習與無監督學習之間。主要是提供給計算機少量有標籤的圖，和大量無標籤的圖，來進行訓練和分類。

(4)　增強學習[1](reinforcement learning)通過觀察來學習做成如何的動作。每個動作都會對環境有所影響，agent 根據觀察到的周圍環境的反饋來做出判斷。比如讓計算機在玩遊戲，並告訴它什麼情況下可以拿到較高的分數，開始的幾局，計算機可能分數不高，但是透過回饋的分數，它能自己總結一個成功和失敗的方法，並愈玩愈好，ALPHAGO 也是應用了這種學習方式。

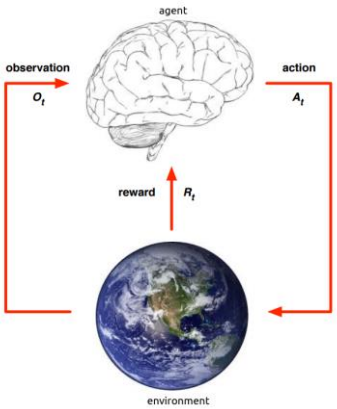

圖 2-3　強化學習

2.2.神經元

神經網路是基於生物大腦的工作原理設計的，由許多人工神經元組成，每個神經元處理多個輸入信號並傳回單個輸出信號，然後輸出信號可以用作其他神經元的輸入信號。一個人工神經元：其輸出是其輸入加權和的 ReLU 函數值。

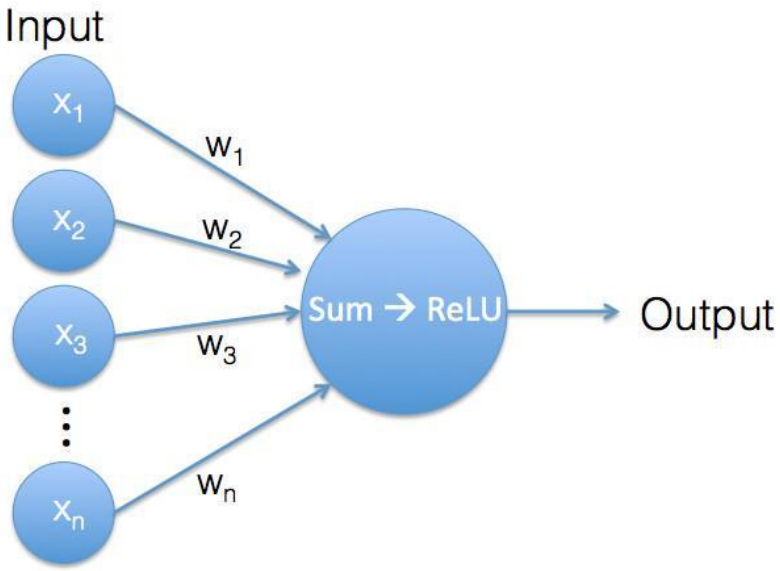

圖 2-4 神經元示意圖

在單個神經元中發生的情況與在 softmax 分類器中發生的情況非常相似。一個神經元有一個輸入值的向量和一個權重值的向量，權重值是神經元的內部參數。輸入向量和權重值向量包含相同數量的值，因此可以使用它們來計算加權和。

$$\text{Weighted Sum}=\text{input1}\times w1+\text{input2}\times w2+...$$

只要加權和的結果是正值，就做為神經元的輸出；但是如果加權和是負值，就忽略該值，神經元產的輸出為 0。此操作稱為整流線性單元（ReLU）。如圖 2-5。

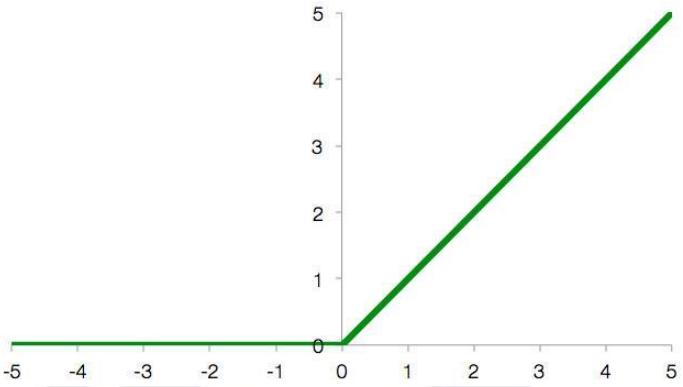

圖 2-5 Relu

2.3.卷積神經網路（Convolutional Neural Network，CNN）

卷積神經網路是一種神經網路結構，它利用卷積和降低維度取樣的運算來取得輸入資料的特徵，卷積神經網路在圖像及語音上有很高的適用性，因為降維取樣的運算可以有效減少網路的學習參數，所以對於輸入多維的圖像時的表現更為優異。

卷積神經網路訓練的目的，是學習大量輸入的映射，學習的模型是由多個卷積層(convolutional Layer)和降維取樣層(subsample layer)交替構成，最後再由完全連接層(fully-connected layer)輸出提取到的高階特徵，做為辨識影像的依據。

卷積神經網路，簡稱 Convnets。當我們提供一個圖片給計算機，這張圖片是有厚度的，所以厚度為 3，他有 RGB 三個原色，然後，我們從圖片取出一小塊，運行 K 個輸出的神網絡。

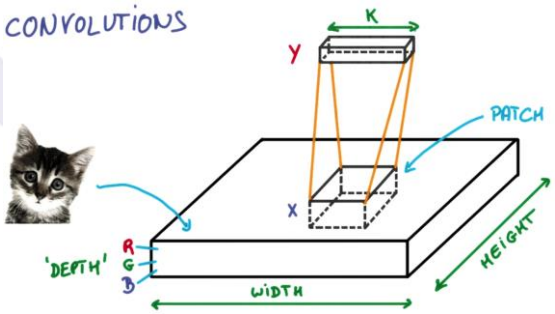

圖 2-6 圖示

資料來源：UDACITY(https://www.udacity.com/)

把這 K 個輸出的神網絡，表示成垂直的一小段，並在所有參數不變的情況下，對整個圖片由上到下，由左到右的掃過一遍，然後我們取得的輸出，是一個比原圖的寬高都更小的圖像，例如在圖 2-7 中的橘色方塊，但是這張圖像比原圖來的厚，這一系列的操作就是卷積**(Convolutional)**，我們會增加數層這樣的操作，這個卷積網絡是組成深度網絡的基礎。

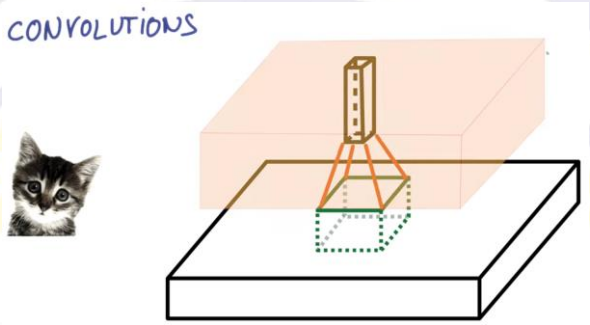

圖 2-7　圖示

資料來源：UDACITY(https://www.udacity.com/)

經過數層的卷積(Convolutional)操作，每一層的操作都在壓縮圖像的維度，同時增加圖像的深度，最後得到的很厚的小圖像，我們可以在圖 2-8 中最右邊的這個方塊，得到一個分類器，因為它包含了整個圖像的特徵訊息。

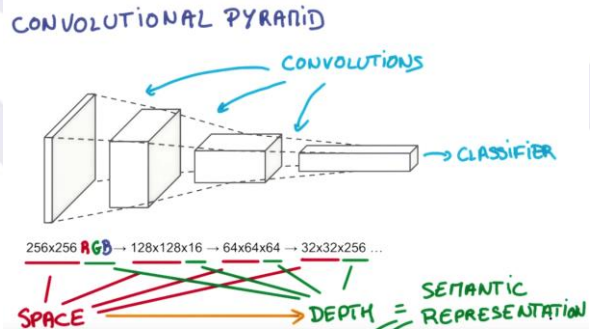

圖 2-8　圖示

資料來源：UDACITY(https://www.udacity.com/)

到這裡，我們接觸到幾個觀念，如圖 2-9 最下面的圖像是一個特徵圖(Feature Map)，從特徵圖取出的 K 個方塊，稱為方塊(patch)或稱為(kernel)

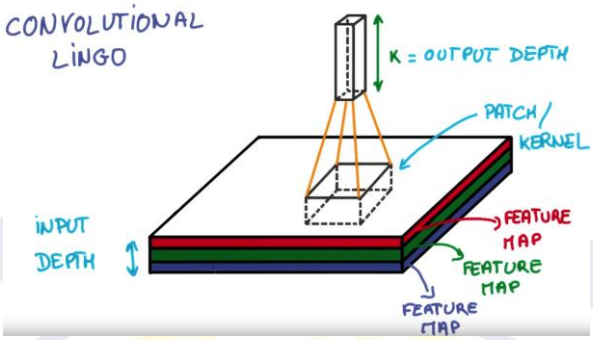

圖 2-9　圖示

資料來源：UDACITY(https://www.udacity.com/)

步幅(stride)是在卷積時，取得壓縮特徵圖時移動的像素，步幅(stride)為 1 時，在卷積取得壓縮特徵圖時，我們不需要填充邊界，稱為有效填充(valid padding)，而大於 1 時，卷積可能會超出邊界，這時需要填充，然後會取到和原圖一樣長寬的特徵圖，稱之為相同填充(same padding)。

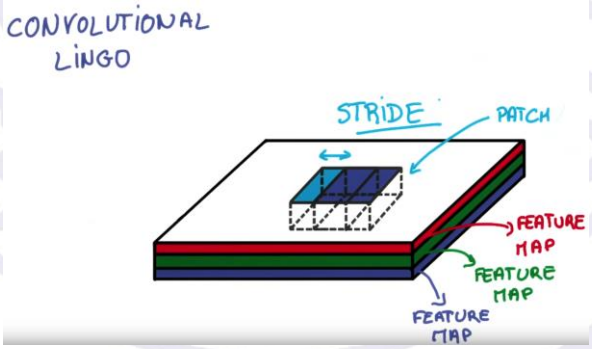

圖 2-10　圖示

資料來源：UDACITY(https://www.udacity.com/)

池化(pooling)，例如要把圖像的特徵抽離成更小的特徵圖，我們把步幅(stride)調成 2，因為步幅(stride)大，在抽離特徵時，可能會丟失掉能辨識圖像的重要特徵，為了解決這種問題，在中間加入了一層步幅(stride)為 1 的中間層，來作為連繫，以達成步幅(stride)2 的跨度。步幅(stride)為 1 時的優點是能取得更精確的特徵訊息，

但相對的缺點是，因為跨度為 1，所以它的計算量更大。

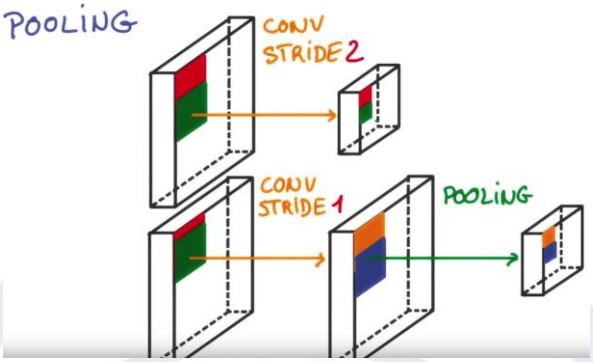

圖 2-11　圖示

資料來源：UDACITY(https://www.udacity.com/)

圖 2-12 是一個典型的卷積神經網路結構，由底層的圖像，經由幾層的卷積(Convolutional)層，卷積(Convolutional)層和池化(Pooling)層共用，可以更好的保存圖像的特徵訊息，然後往上的 Fully Connected 是神經網路的隱藏層，最後就是分辨器(Classifier)

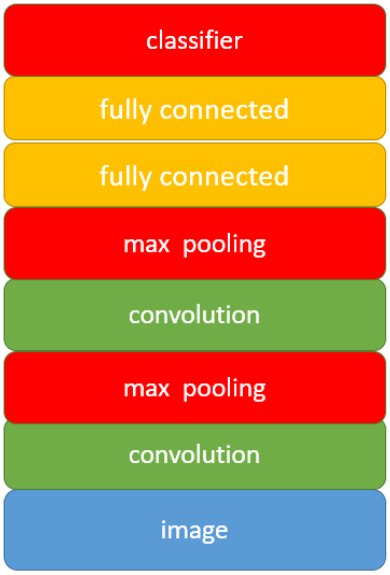

圖 2-12　Layer

2.4.池化(pooling)

在卷積神經網路中，我們經常會碰到池化操作，而池化層往往在卷積層後面，通過池化來降低卷積層輸出的特徵向量，同時不易出現過擬合。

為什麼可以降低維度？因為圖像具有一種「靜態性」的屬性，這也就意味著在一個圖像區域有用的特徵極有可能在另一個區域同樣適用。因此，為了描述大的圖像，一個很自然的想法就是對不同位置的特徵進行聚合統計，例如，人們可以計算圖像一個區域上的某個特定特徵的平均值 (或最大值)來代表這個區域的特徵。

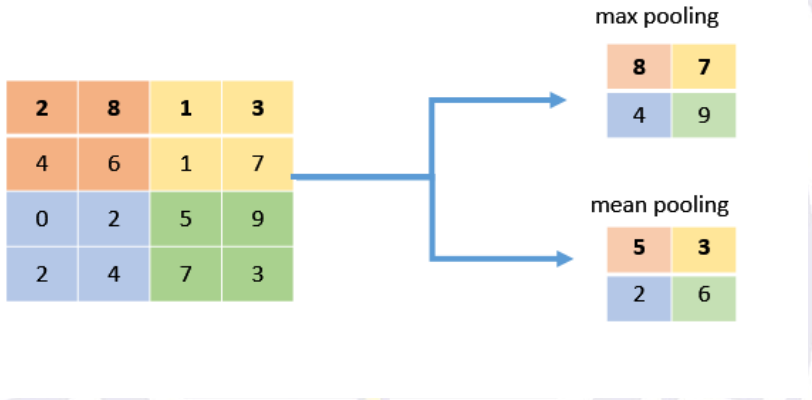

圖 2-13　最大池化或平均池化

2.5.過擬合(overfitting)

隱藏層的數量會影響神經網路最後預測的結果，隱藏層的神經元個數會決定神經元的數量，在訓練過程中，網路參數不足，導致不能進行預測，程為擬合不足(underfitting)，相對地隱藏層數量太多，使得網路參數太多，則會造成過擬合。

簡單來說過擬合是機器學習對於資料模型的訓練過於追求完美，以下圖 2-14 的藍線來說它的誤差率可能為 10，而黃線的誤差率為 1，因為它完美的經過了每一個點，但是藍色線(斜直線)可以拿來做為預測的模型，而黃色線(不規則曲線)則不行。另外一種情況是，解決過擬合的方法有:

方法一：增加數據量，大部分過擬合產生的原因是因為數據量太少了，如果我們有成千上萬的數據，黃線也會慢慢被拉直，變得沒那麼彎曲。

方法二:dropout：在訓練的時候，我們隨機忽略掉一些神經元和神經聯結，使這個神經網路變得不完整，用一個不完整的神經網路訓練一次。

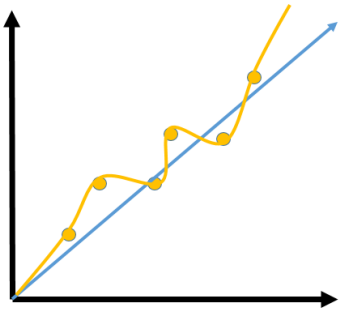

圖 2-14 過擬合(overfitting)

2.6. Tensorflow

TensorFlow[3]是 Google 開發的一款應用神經網路，使用 Python 語法的套件包，也是一個採用數據流圖來進行數值計算的開源軟件庫，Tensorflow 在 2016/11 開放了支援 windows 系統，所以本文可以在視窗環境下進行實驗。

利用 TensorFlow 我們可以快速的入門訓練深度神經網路， 大大降低深度學習的開發成本和開發難度，TensorFlow 的開源性， 讓所有人都能使用並且維護它，使用 Tensorflow 訓練辨識模型的過程，分為四个主要步驟：

(1).收集訓練用的圖像像。

(2).使用 TensorFlow 和 Inception 模型來訓練一個計算圖（模型）。

(3).編寫 script 來使用計算圖進行圖像分類。

(4).通過對新圖像進行分類來辨識圖片。

2.6.1.數據流圖（Data Flow Graph）

Tensorflow 利用數據流圖（Data Flow Graph）用“節點”(nodes)和“線”（edges）的有向圖來描述數學計算。“節點”一般用來表示施加的數學操作，但也可以表示數據輸入（feed in）的起點/輸出（push out）的終點，或者是讀取/寫入持久變量（persistent variable）的終點。“線”表示“節點”之間的輸入/輸出關係。這些數據“線”可以運輸“size 可動態調整”的多維數組，即“張量”（tensor）。

以下是各名詞的解釋：

- 圖（Graph）：圖描述了計算的過程，TensorFlow 使用圖來表示計算任務。
- 張量（Tensor）：TensorFlow 使用 tensor 表示數據。每個 Tensor 是一個類型化的多維數組。
- 操作（op）：圖中的節點被稱為 op（operation 的縮寫），一個 op 獲得 0 個或多個 Tensor，執行計算，產生 0 個或多個 Tensor。
- 會話（Session）：圖必須在稱之為“會話”的上下文中執行。會話將圖的 op 分發到諸如 CPU 或 GPU 之類的設備上執行。
- 變量（Variable）：運行過程中可以被改變，用於維護狀態。

2.6.2. Tensorflow 深度學習引擎

TensorFlow 是 Google 所開發的深度學習引擎的開源軟體 (open source software library)。Google 公司在設計 Tensorflow 架構時便考慮到靈活性 (flexible)，它能部署在桌面，雲端服務器，車輛，或移動設備。TensorFlow 最初是由研究人員和谷歌的機器智能研究機構，為了進行機器學習和神經網路研究的目的而開發。利用 TensorFlow 我們可以快速的入門訓練深度神經網路， 大大降低深度學習的開發成本和開發難度，TensorFlow 的開源性， 讓所有人都能使用並且維護它。

Tensorflow 支援 C++和 Python 語法的套件，Tensorflow 以數據流圖形 (data flow)表示深度學習計算過程。如圖 2-14 所示。圖中的 node 表示的數學運算，而圖的 edge 代表它們之間傳送的多維數據陣列稱為 tensor。Node 可以被分配到多個計算設備上，可以非同步和並行地執行操作。因為是有向圖，所以只有等到之前的入度節點們的計算狀態完成後，當前節點才能執行操作。

TensorFlow graph 基本名詞：

- graph ： 表示一整個計算任務
- op.： graph 中的節點，也就是對資料的操作
- Session ： 在 Session 中才會執行 graph

- tensor ： 表示數據資料，a multidimensional array of numbers
- variable ： 表示變數資料，負責維護狀態
- feed 和 fetch：賦值或者從其中獲取數據

3. 實作圖像辨識模型

3.1. 常用深度學習模型

1985 年，Rumelhart 和 Hinton 提出了後向傳播（Back Propagation）演算法，使得神經網路的訓練變得簡單可行，這篇文章光是在 Google Scholar 上的引用次數就接近 2 萬次。幾年後，LeCun 利用後向傳播來訓練多層神經網路用於識別手寫郵政編碼[2]，這就是後來 CNN 的開山之作。

CNN 在 AlexNet [12]被發表了後變得越來越強大。AlexNet 有 6000 萬個參數和 65 萬個神經元，由五個卷積層組成。那些五個卷積層連著最大池化層，三個全連接層，最終再連一個 1000 way 的 softmax 層[12]。AlexNet 在百萬量級的 ImageNet 數據集合上，效果大幅度超過傳統的方法，從傳統的 70%多提升到 80%多。

GoogLeNet 是 ILSVRC 2014 年的獲勝者。它大大減少了 ImageNet 的 TOP-5 錯誤率，從 16.4% 降到 6.7% [15]。GoogLeNet 開始使用深卷積結構，擁有參數較少的優點（400 萬個，與 AlexNet 的 6000 萬個相比少了很多）[14]。後來有幾個更高級版本 Inception V2-V4，後來的架構得到了改善。

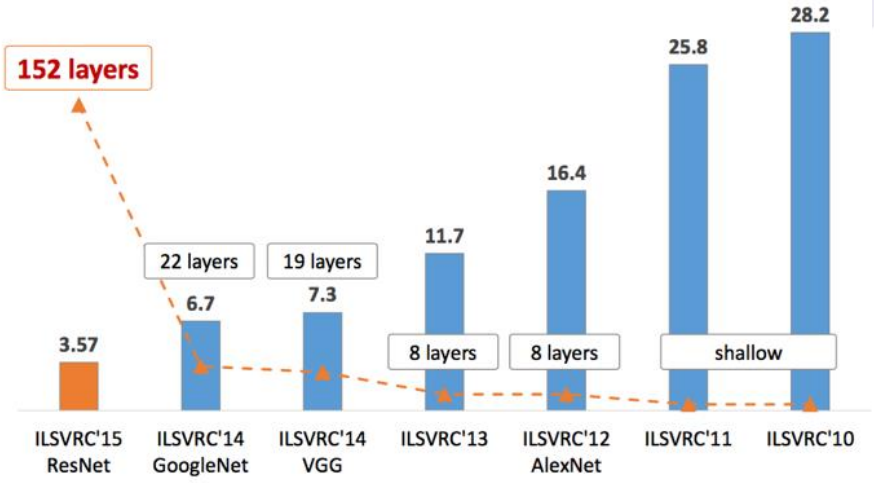

圖 3-1 比較圖

模型名	AlexNet	VGG	GoogLeNet	ResNet
年份	2012	2014	2014	2015
層數	8	19	22	152
Top-5 錯誤	16.4%	7.3%	6.7%	3.57%
數據擴張	+	+	+	+
Inception(NIN)	-	-	+	-
卷積層數	5	16	21	151
卷積核大小	11,5,3	3	7,1,3,5	7,1,3,5
全連接層數	3	3	1	1
全連接層大小	4096,4096,1000	4096,4096,1000	1000	1000
退出	+	+	+	+
當地回應正常化	+	-	+	-
批標準化	-	-	-	+

表格 3-1　AlexNet、VGG、GoogLeNet、ResNet 對比

GoogleLeNet 主要是用於識別 10 個手寫數字的，當然，只要稍加改造也能用在 ImageNet 數據集上，但效果較差。而本文要介紹的後續模型都是 ILSVRC 競賽歷年的佼佼者，這裡具體比較 AlexNet、VGG、GoogLeNet、ResNet 四個模型。

3.2. 優化器選擇

優化器的選擇對損耗下降過程有很大的影響。我們比較 3 個最常用的優化器來看他們的表現。我們利用一個簡單的線性函式，來測試這 3 個優化器。

(1) GradientDescentOptimizer

SGD 就是每一次迭代計算 mini-batch 的梯度，然後對參數進行更新，是最常見的優化方法。SGD 有許多缺點，例如選擇合適的 learning rate 比較困難 - 對所有的參數更新使用同樣的 learning rate。對於稀疏數據或者特徵，有時我們可能想更新快一些對於不經常出現的特徵，對於常出現的特徵更新慢一些，這時候 SGD 就不太能滿足要求了。SGD 容易收斂到局部最優，並且在某些情況下可能被困在鞍點。正因為這些缺點後續才會發展出各種算法。

(2) AdamOptimizer

Adam(Adaptive Moment Estimation)本質上是帶有動量項的 RMSprop，它利用梯度的一階矩估計和二階矩估計動態調整每個參數的學習率。 Adam 的優點主要在於經過偏置校正後，每一次迭代學習率都有個確定範圍，使得參數比較平穩。

(3) RMSPropOptimizer

RMSprop 可以算作 Adadelta 的一個特例，RMSprop 算是 Adagrad 的一種發展，它被提出來解決 Adagrad 的過早收斂，和 Adadelta 的變體，效果趨於二者之間，適合處理非平穩目標 - 對於 RNN 效果很好。

由圖 3-2 我們比較這幾個最常用的優化器來看他們的表現。以第一行的圖表的 loss 來看，這是比較優化器的預測值距真正的值的差距，3 個優化器的差異不大。以第二層訓練來看，可以看出，RMSprop 在開始時下降得更快，更快的收斂。而下降較慢但更穩定。

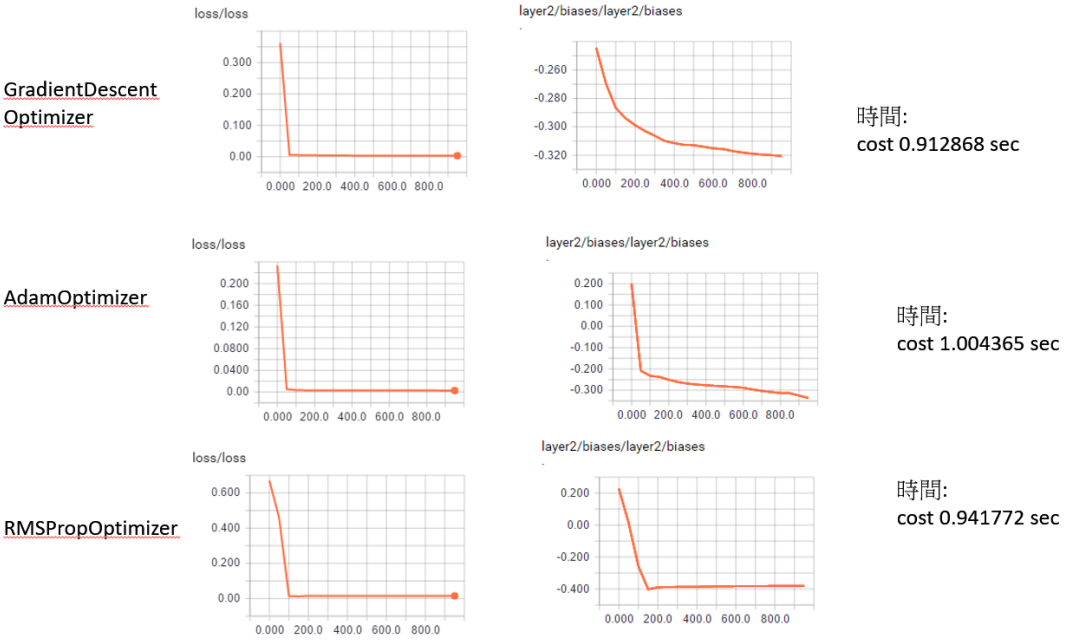

圖 3-2 優化器比較

3.3.圖像識別模型的訓練

Tensorflow 的圖像識別模型訓練，會使用 Inception v3 模型訓練一個新的頂層，可以識別其他類圖片，使用 Incenption V3 模型，不會有大的捲積核，模型會把 5x5 的卷積用兩個 3x3 代替，兩個 3x1 代替 3x3 卷積核，v3 在 raw 的 v2 上做瞭如下變化：RMSProp 替代 SGD，採用 Inception v3 架構模型進行培訓 ImageNet 圖像，頂層作為輸入接收每個圖像的 2048 維向量。 我們在該表示之上訓練一個 softmax 層。假設 softmax 層包含 N 個標籤。

圖像識別模型訓練，一開始會下載 google 的圖庫.放至 inception 目錄，接著會準備要訓練的圖片訓練標籤，建立分類文字檔，接著會修剪圖像，並增加一層訓練層以激勵函數及優化器，最後完成了所有的訓練，並以沒有使用的一些新圖像進行最終的測試，來驗証模型。

3.2 使用 CIFAR-10 數據集進行圖像分類

這裡我們使用標準的 CIFAR-10 數據集。CIFAR-10 包含了 60000 幅圖片。有 10 個不同的分類，每類包含 6000 幅圖片。每幅圖片的規格是 32x32 像素。這麼小尺寸的圖片對我們人類來說有時很難進行正確的分類，但它卻簡化了計算機模型的任務，並降低了圖片分析時的計算量。

圖 3-3 CIFAR-10 數據集中 10 個分類中的隨機圖片

資料來源：readhouse(http://www.readhouse.net/articles/189539568/)

3.2.1 以 softmax 實作一個簡單的圖像分類模型

在程式實做一開始，先加載 CIFAR-10 數據集，將 60000 幅圖像的數據集分為兩部分。大的一部分包含 50000 幅圖像。這些數據集用於訓練我們的模型。另外的 10000 幅圖像被稱作測試集。在訓練結束之前，我們的模型將不會看到這些圖像。直到模型中的參數不再變換，我們使用測試集作為模型輸入來檢驗模型的性能。

然後開始建立模型，如圖 3-4，我們的目標是如何把每張圖像對應到 10 個分類中，我們所採取的一種簡單的方法是單獨查詢每個像素。檢查這個像素的顏色增加或減少了它屬於某個種類的可能性。比如說像素顏色是紅色。如果汽車圖片的像素通常是紅色，我們希望增加“汽車”這一種類的得分。我們將像素是紅色通道的值乘以一個正數加到“汽車”這一類的的得分裡。對所有的 10 個分類我們都重複這樣的操作，對每一個像素重複計算，3072 個值進行相加得到一個總和。3072 個像素的值乘以 3072 個加權參數值得到這一分類的得分。最後我們得到 10 個分類的 10 個分數。然後我們挑選出得分最高的，將圖像打上這一類型的標籤。

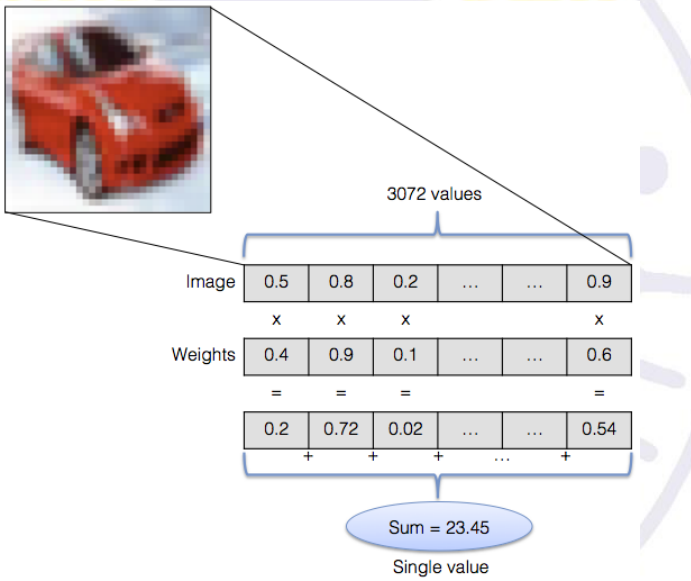

圖 3-4

資料來源：kknews(https://kknews.cc/zh-mo/news/kxgk3kq.html)

我們可以用矩陣的方法，這樣使像素值乘以加權值再相加的過程大大簡化。我們的圖像通過一個 3072 維向量表示。如果我們將這個向量乘以一個 3072x10 的加權矩陣，結果就是一個 10 維向量。它包括了我們需要的加權和一個圖像在所有 10 個類別中的分數。

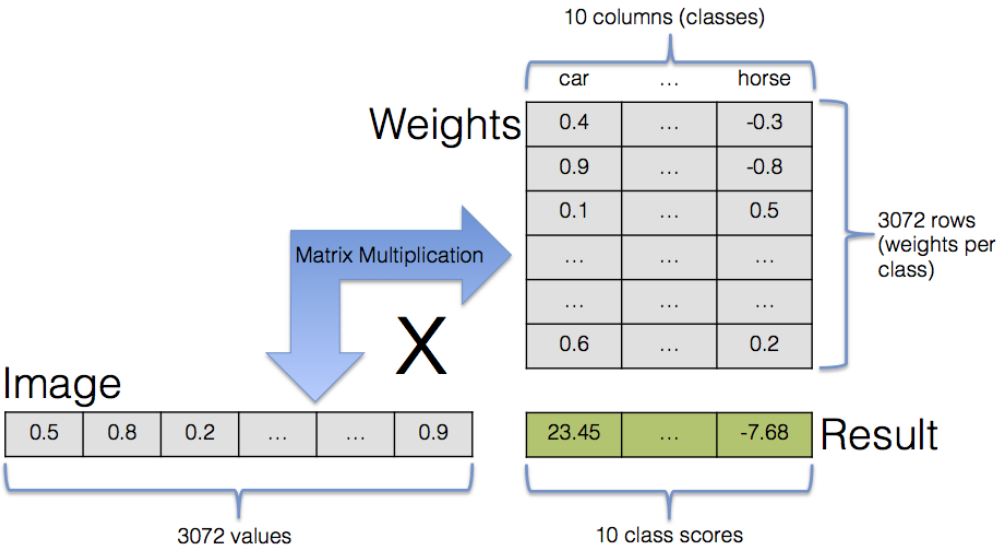

圖 3-5

資料來源：kknews(https://kknews.cc/zh-mo/news/kxgk3kq.html)

如圖 4-7，我們運行這個簡單的模型，在訓練一段時間後結果精確度大約落在 25-35%。這個模型其實還比較原始。它對具體圖像的比如線和形狀等特徵毫無概念。它只是單獨檢測每個像素的顏色，完全不考慮與其他像素的關聯。對一幅圖像某一個像素的修改對模型來說意味著完全不同的輸入。所以準確率看起來沒有很高。

如果我們多進行幾次訓練，發現訓練的準確率並不是穩定上升的，而是在 0.23 至 0.44 之間波動。看起來已經到達模型的極限，再進行更多的訓練也於事無補。這個模型不能再提供更好的結果。事實上，比起進行 1000 次的訓練，我們進行較少的訓練次數也能得到相似的準確率。

最後一個數字為測試精確度，如果大大低於訓練的精確度，並且差距非常大，這也意味著過度擬合。模型針對已經見過的訓練數據進行了精細的調整，而對於以前從未見過的數據則無法做到這點。

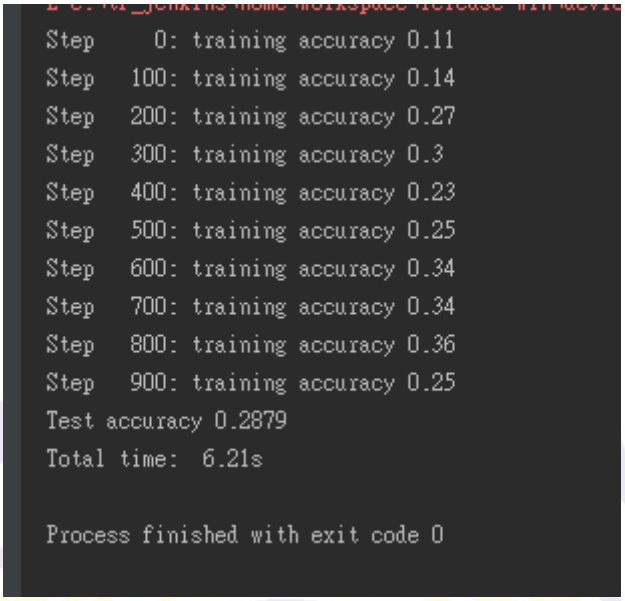

圖 3-6 訓練結果

3.2.2 以神經網路實作一個圖像分類模型

我們使用 ReLU 的原因是其具備非線性的特點，因而現在神經元的輸出並不是嚴格的輸入線性組合。人工神經網路中的神經元通常不是彼此隨機連接的，大多數時候是分層排列的，如圖 3-7，人工神經網路具有隱藏層和輸出層 2 個層。輸入並不被當作一層，因為它只是將數據傳送到第一層。

輸入圖像的像素值是第 1 層網路中的神經元的輸入。第 1 層中的神經元的輸出是第 2 層網路的神經元的輸入，後面的層之間以此類推。如果沒有每層的 ReLU，我們只是得到一個加權和的序列；並且堆積的加權和可以被合併成單個加權和，這樣一來，多個層並沒有比單層網絡有任何改進之處。這就是為什麼要具有非線性的重要原因。 ReLU 非線性解決了上述問題，它使每個附加層的確給網路添加了一些改進。

我們所關注的是圖像類別的分數，它是網路的最後一層的輸出。在這個網路架構中，每個神經元連接到前一層的所有神經元，因此這種網路被稱為完全連接的網路。

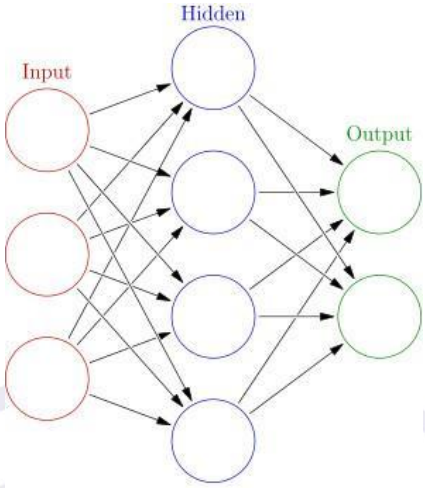

圖 3-7　神經網路

如圖 3-8 利用 tensorboard，在「事件」標籤中，我們可以看到網路的損失是如何減少的，以及其精度是如何隨時間增加而增加的。

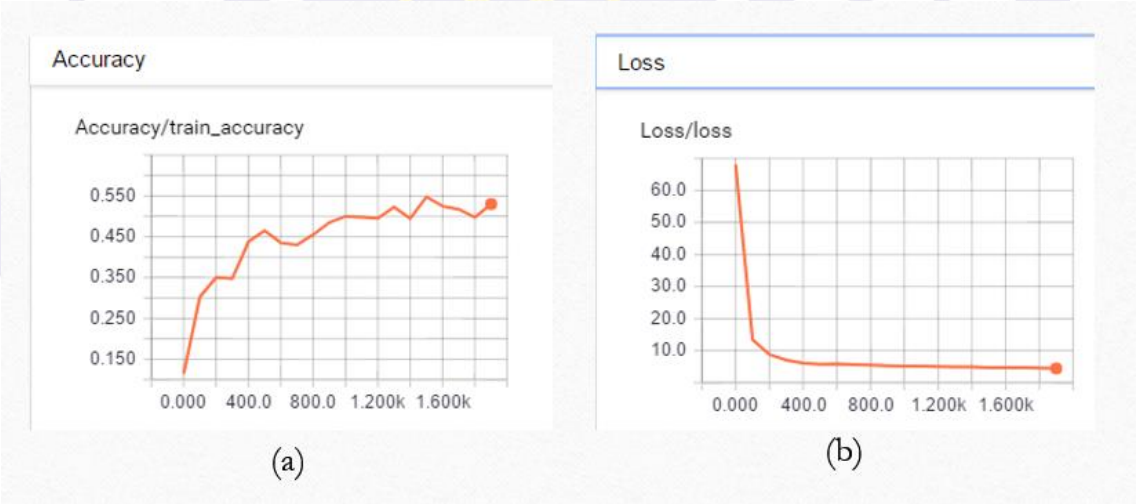

圖 3-8　(a) 精度　(b) 網路的損失

如圖 3-9 tensorboard，在「Graphs」標籤中，顯示一個已經定義的可視化的 tensorflow 圖。

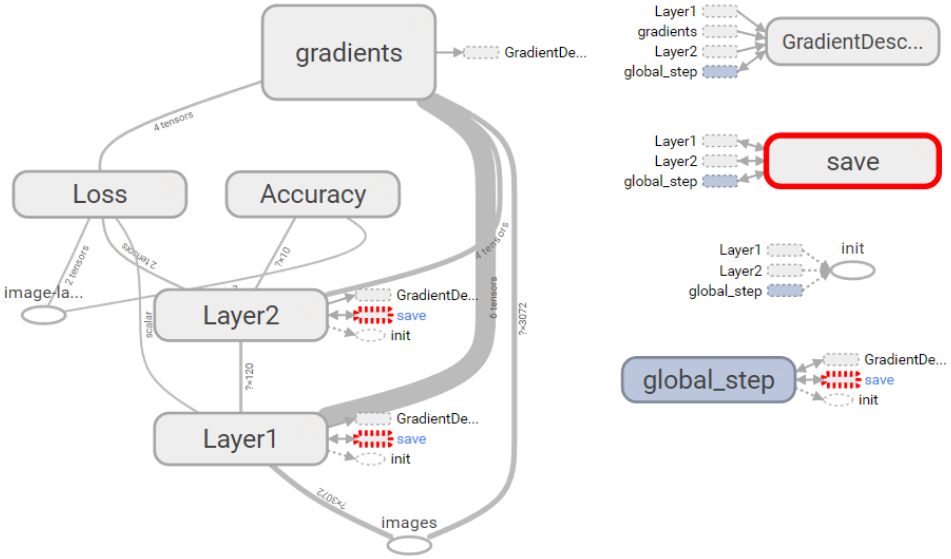

圖 3-9　tensorflow Graph 圖

如圖 3-10 可以看到訓練的準確性在第一次約 1000 次迴圈中，精度增加到約 50%，最後的測試精度為 46%。這表明我們的模型沒有顯著過度擬合。softmax 分級器的性能約為 30%，因此 46%的改進約為 50%。這一次使用神經網路的訓練比 4.1.1 節的簡單模型，有顯著的提升。

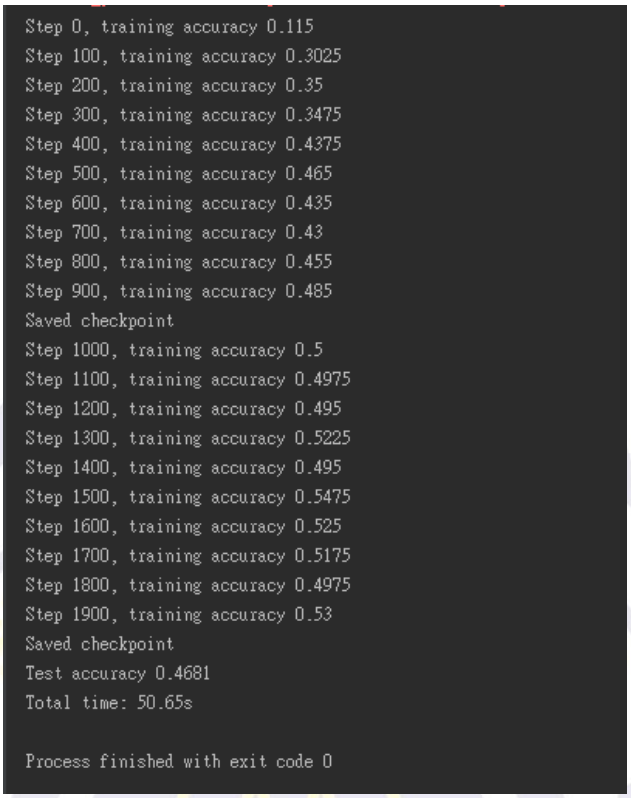

圖 3-10　神經網路訓練結果

在 3.2.1 節中，我們可以發現 softmax 分類器的計算時間比神經網路少很多。但即使把訓練 softmax 分類器的時間增加到和神經網路來訓練所用的時間一樣長，前者也不會達到和神經網路相同的性能，前者訓練時間再長，額外的收益和一定程度的性能改進幾乎是微乎其微的。

本文也已經在神經網路中也驗證也這點，額外的訓練時間不會顯著提高準確性。在神經網路中，我們可以通過改變參數，如隱藏層中的神經元的數目或學習率，這樣便能夠提高模型的準確性，模型的參數優化可以使測試精度可能大於 50%。但還有另一種類型的網絡結構能夠比較輕易實現這一點，就是本文將用於 4.3 節實驗的卷積神經網路，這是一類不完全連通的神經網路，相反，卷積神經網路嘗試在其輸入中理解局部特徵，這對於分析圖像非常有用。

4. 實驗結果

4.1 硬體設備

CPU	Intel I7
顯示卡	NVIDIA GTX1060
記憶體	32G
OS	Win10
開發 IDE	PyCharm

表格 4-1 主機規格

Samsung tab s	**Model-SM-T705Y**
系統	安卓 5.0.2

表格 4-2 行動裝置規格

4.2.系統架構

如圖 4-1 本文的目標是實作一個用於大型醫療中心的室內導航系統，我們希望能有一個不用大量設備的室內地點辨識系統。我們由醫院中的各個走道，掛號處，樓梯，電梯等處，拍攝圖像，做為系統訓練辨識圖像的來源。我們採用已經預先透過 ImageNet 訓練好的 InceptionV3 作為 CNN 模型，ImageNet 圖片庫龐大，因此預先訓練好的 InceptionV3 從中已經學習大量特徵，也就是說卷積核的參數已訓練完成，接著藉由我們處理過並分類的圖片訓練最後一層神經網路，這樣一來便能辨識與分類地點圖片。

圖 4-1　系統架構

4.2.TF-Slim 的實驗數據

Slim 它是一個用於在 TensorFlow 中定義、訓練和評估模型的輕量軟體包。可以讓我們快速地定義模型，同時還能維持模型架構透明。我們以榮總拍攝的圖片進行模型的完整訓練，使用建立在 TenorFlow　TF- Slim 實驗庫下的 Inception-V3 神經網路模型，進行模型的訓練。

4.2.3. Tfrecord

從宏觀來講，tfrecord 其實是一種數據存儲形式，Tensorflow 有和 tfrecord 配套的一些函數，可以加快數據的處理。

圖片數	訓練集張數	驗證集張數	分類	TFRecord檔大小	訓練次數	訓練時間	模型大小	loss
100	80	20	9	2.92 MB	2000	29:53	278 MB	NaN
300	260	40	9	9.21 M	2000	30:02	278 MB	2.3024
500	400	100	9	16.6 MB	2000	30:03	278 MB	2.6156
700	560	140	9	22.9 MB	2000	30:03	278 MB	3.5714
900	720	180	9	30.3 MB	2000	29:58	278 MB	1.8385

表格 4-3

4.3.圖像辨識於經過壓縮的圖

壓縮圖片，調整像素為原本的三分之一，整體圖檔的大小縮減至原本的四分之一，將近 33Mb 目前在本機實驗，使用 Tensorflow 的 CPU 模式來運算計算圖，

通過對新圖像進行分類來辨識圖片。給定一個花的圖片，透過已訓練好的計算圖模型，計算機能自己辨識出是哪一種花卉。例如我們提供一個玫瑰的圖片，經由計算機判斷後，它給出這張圖有 0.8 的機率是玫瑰。

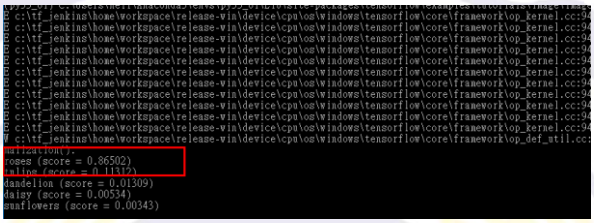

圖 4-2 辨識圖片

我們針對提供給計算機訓練用的圖片，進行壓縮，在減小訓練用圖片的總體體積時，也同時減小圖片的尺寸及像素，並以這些資料重新訓練計算圖模型，觀察壓縮後減少的像素對辨識率的影響，我們使用相同的 3670 張花卉的訓練圖片資料，辨識相同的一張玫瑰花的圖片，由表 4-3 來看，壓縮後的訓練資料，對訓練時間，沒有顯著的影響，在辨識率的數值來看，也沒有明顯的影響。

圖像縮放	圖片容量	訓練時間	辨識率
50%	49Mb	1m11 s	0.83977
30%	33Mb	1m23 s	0.88887
15%	22.8Mb	1m15 s	0.87385
10%	49Mb	1m30 s	0.86162

表格 4-4 比較表

4.4.影像辨識的各項實驗數值

在開放的公共空間，當在進行地點辨識時，應該會遇到有行人遮擋的情況，我們模擬行人遮擋的情況，在圖像特徵被遮擋的情況下，查看系統對於地點辨識的正確率，在一般情況下都能有很高的辨識率。

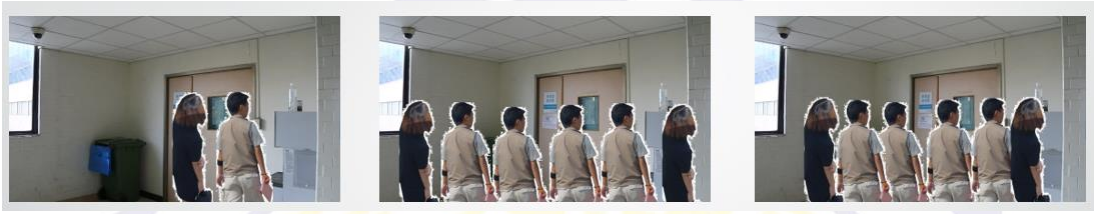

圖 4-3　手術室衛材庫地點圖片在行人遮蔽模擬

如圖 4-4，在某些燈光較陰暗的地點如停車場，或採光不足的停車場，這時取到的圖像可能是模糊失真的，以下測試在不同光線強度下，影像辨識的正確率。在光線對比降低 50%的影像中平均正確率為 0.9314，降低至 100%後平均正確率為 0.8287，如圖 4-4 為光線對比降低 100%之醫院走廊影像，在圖片相當昏暗的情況下此圖片也有高達 0.993 之正確率，依實驗結果分析低光源對於深度學習影像辨識影響差異不大(降低 0.005%)。在表格 4-4 中，我們也觀察到影像辨識的執行時間平均約 4.8 秒，這個平均值對於系統在行動裝置上的應用會有很大影響，我們在使用行動裝置做影像辨識時，需要給系統反應的時間，才能有好的圖像辨識正確率。

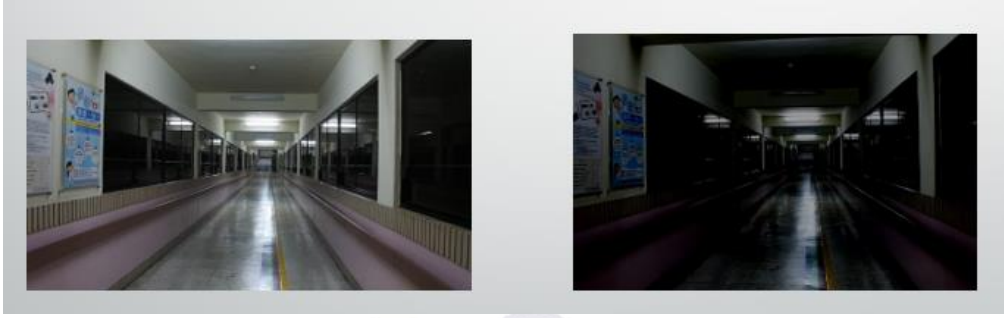

圖 4-4　醫療大樓走廊地點圖片的明暗度模擬

地點	圖片數量	平均正確率	執行時間	遮蔽10%	遮蔽30% 分散	遮蔽30%	亮度比例 -100	亮度比例 -50
手術室衛材庫	79	0.977518	4.71925	0.96797	0.93797	0.93364	0.98509	0.99649
2F電梯口	90	0.987625	4.773	0.90950	0.90767	0.93517	0.50195	0.98585
資訊室門口	37	0.98701	4.7653	0.95470	0.96995	0.99033	0.54272	0.64026
醫療大樓走廊	161	0.996063	4.87525	0.94518	0.81162	0.95327	0.97743	0.99945
急診藥局	110	0.995223	4.8315	0.96446	0.95711	0.95327	0.58785	0.99319
急診走廊	191	0.986603	4.78875	0.93707	0.92390	0.98670	0.99745	0.99261
3F電梯口	72	0.991705	4.7075	0.96420	0.91293	0.90963	0.86831	0.7756
停車場	216	0.986334	4.8406	0.92675	0.97747	0.99731	0.98509	0.99988
資訊室內部	38	0.971816	4.7204	0.95309	0.93759	0.92537	0.50195	0.99649

表格 4-5 測試數值

4.5 於行動裝置上的場景辨識

如圖 4-3 我們將訓練好的 Inception 分類器，裝載在有攝像頭的行動裝置上運行，並觀察實際運作的情況，實際運行下來，都能準確的辨識出場景位置，但在一些穩定度上會受到影响，例如鏡頭的搖晃，辨識時系統需要平均約 4.8 秒的反應時間，如果在這期間有搖晃，就會影響到辨識，至於在辨識準確度上，在某些特徵不明顯的場景，例如空白的牆面，因為特徵不明顯，系統可能會有誤判。

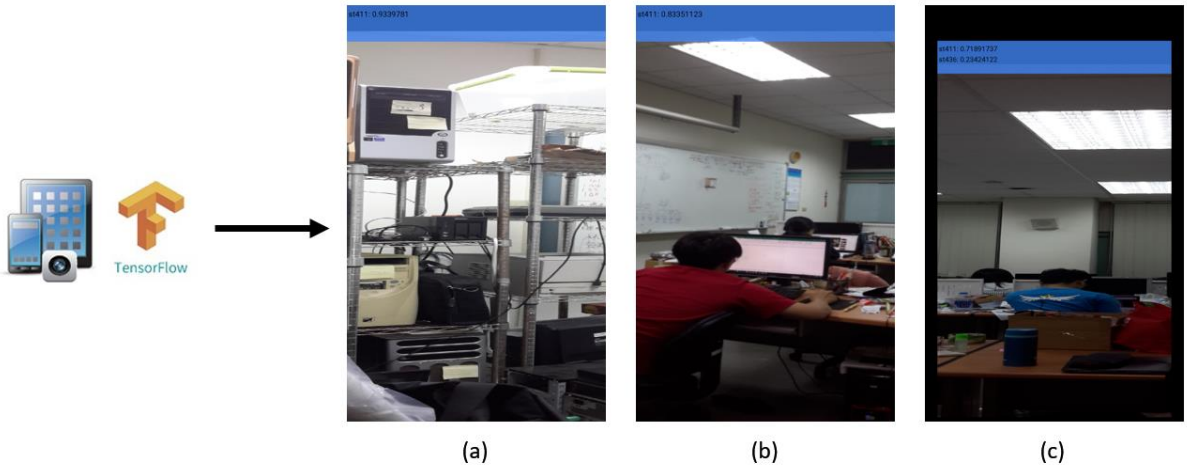

圖 4-3 於行動裝置上進行 ST411 室的場景辨識，圖(ａ)的辨識率為 0.97，圖(b)的辨識率為 0.83，圖(c)的辨識率為 0.71

5. 結論

圖像辨識是機器學習的一項重要任務，因為視覺是最重要的一項感知能力。對於人類來說，看見東西，並且在看的同時，由大腦進行分辨判斷，是本能一樣的能力，這一系列的處理，包含了太多的工作，所以對於計算機來說，可以說是困難無比，本文在手持裝置上實現基於深度學習的場景辨識系統，而卷積神經網路對圖像的辨識，即使在遮蔽或圖像有失真的情況下，也能有很高的辨識度，我們對系統未來的使用有一些期望能加強的方向，例如加入室內導航的功能，或者是進一步結合 AR 擴增實境，讓使用者操作上變得更便利更直覺。

6. 參考文獻

[1] Martin, C., Correa, J., Han, H., Allen, H., Rood, J., Champagne, C., Gunturk, B., Bray, G.: Validity of the remote food photography method (RFPM) for estimating energy and nutrient intake in near real-time. Obesity (2011)

[2] Noronha, J., Hysen, E., Zhang, H., Gajos, K.Z.: Platemate: crowdsourcing nutritional analysis from food photographs. In: ACM Symposium on UI Software and Technology (2011)

[3] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010.

[4] Y. Matsuda and K. Yanai, "Multiple-food recognition considering co-occurrence employing manifold ranking," in Pattern Recognition (ICPR), 2012 21st International Conference on, 2012.

[5] TADA: Technology Assisted Dietary Assessment at Purdue University, West Lafayette, Indiana, USA, available at http://www.tadaproject.org/.

[6] Bossard, L., Guillaumin, M., & Van Gool, L. (2014, September). Food-101–mining discriminative components with random forests. In European Conference on Computer Vision (pp. 446-461). Springer International Publishing.

[7] Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features:

Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)

[8] S ánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image Classification with the Fisher Vector: Theory and Practice. IJCV (2013)

[9] Bosch, A., Zisserman, A., Munoz, X.: Image Classification using Random Forests and Ferns. In: ICCV (2007)

[10] Moosmann, F., Nowak, E., Jurie, F.: Randomized clustering forests for image classification. PAMI (2008)

[11] Singh, S., Gupta, A., Efros, A.A.: Unsupervised discovery of mid-level discriminative patches. In: ECCV (2012)

[12] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)

[13] Convolutional Neural Networks (CNNs / ConvNets), http://cs231n.github.io/convolutional-networks/

[14] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-9).

[15] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.