

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

基於深度學習之場景辨識與路徑決策

Scene Recognition Based on
Deep Learning and Path Navigation

研究生：林庭緯

中華民國一百零六年七月

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 林庭緯 所提之論文

基於深度學習之場景辨識與路徑決策

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

黃國辰

簽章

委

員

陳隆彬

焦信達

指導教授

陳隆彬

簽章

中華民國 106 年 6 月 28 日

摘要

本文提出方法並實現一套能透過影像辨識完成定位與導航的系統。由於 GPS 定位在室內並不能夠完全發揮其功效，而其他各種定位技術也各有其優缺點。本文研究以電腦視覺領域透過深度學習達成場景辨識，使用馬可夫決策過程(MDP)紀錄並統計使用者路線，大量資料透過 Q-Learning 學習，能夠訓練出最佳路徑。

關鍵詞：機器學習、深度學習、卷積神經網路、Q-Learning



Abstract

In this paper, we propose a method and implement a system that can locate user by computer vision. And the system also can navigate the users to destination. As GPS can not play utility in indoor place, and other positioning technology also has its advantages and disadvantages. In this paper, we research the application of Deep Learning in scene recognition system, and use Markov Decision Process (MDP) to record and count user routes, then find the best path. by Q-Learning.

Keywords : Machine Learning 、 Deep Learning 、 Convolutional Neural Network 、 Q-Learning



目錄

| | |
|-----------------------------------|----|
| 摘要..... | 3 |
| Abstract..... | 4 |
| 圖表目錄..... | 6 |
| 1. 研究目的..... | 7 |
| 2. 背景與相關知識..... | 8 |
| 2.1. 深度學習..... | 8 |
| 2.1.1 類神經網路..... | 8 |
| 2.1.2 激活函數..... | 9 |
| 2.1.3 Softmax 回歸..... | 10 |
| 2.1.4 反向傳播算法..... | 10 |
| 2.1.5 卷積神經網路..... | 11 |
| 2.2 Q-Learning..... | 13 |
| 3. 系統架構與方法..... | 15 |
| 3.1 圖片資料集..... | 16 |
| 3.2 CNN 架構..... | 16 |
| 3.3 CNN 訓練程序..... | 17 |
| 3.4 路徑資料收集..... | 17 |
| 3.5 Q-Learning 設計..... | 18 |
| 4. 系統實作與實驗結果..... | 19 |
| 4.1 深度學習影像辨識地點之辨識正確率..... | 20 |
| 4.2 深度學習影像辨識各影像大小與解析度之辨識時間..... | 21 |
| 4.3 深度學習影像辨識對開方式式地點日夜間之辨識正確率..... | 22 |
| 5. 結論與未來展望..... | 27 |
| 參考文獻..... | 28 |

圖表目錄

| | |
|---|----|
| 圖表 1 各項定位技術比較..... | 7 |
| 圖表 2 神經網路..... | 8 |
| 圖表 3 神經元..... | 9 |
| 圖表 4 ReLU 函數，優點為速度快，減輕梯度消失問題還有稀疏性..... | 9 |
| 圖表 5 上圖為單一神經元連結所有像素，下圖使用局部連結，每神經元作為 2*2 的 Filter..... | 11 |
| 圖表 6 藍色為輸入，綠色為卷積核..... | 12 |
| 圖表 7 特徵卷積計算..... | 12 |
| 圖表 8 採樣計算..... | 13 |
| 圖表 9 強化式學習概念..... | 13 |
| 圖表 10 馬可夫決策過程..... | 14 |
| 圖表 11 系統架構..... | 15 |
| 圖表 12 圖片辨識分類與模型訓練..... | 15 |
| 圖表 13 Inception V3 模組..... | 16 |
| 圖表 14 CNN 訓練流程..... | 17 |
| 圖表 15 路徑辨識流程..... | 17 |
| 圖表 16..... | 18 |
| 圖表 17 R5 矩陣，Q5 矩陣..... | 18 |
| 圖表 18 實驗設備與開發環境..... | 19 |
| 圖表 19 網頁使用介面..... | 19 |
| 圖表 20 及時辨識 APP..... | 20 |
| 圖表 21 走廊特徵示意圖..... | 20 |
| 圖表 22 地點辨識正確率..... | 21 |
| 圖表 23 辨識執行時間與正確率..... | 21 |
| 圖表 24 模擬夜間走廊影像..... | 22 |
| 圖表 25 降低光線對影像辨識正確率影響..... | 22 |
| 圖表 26 大小 10*10 空間狀態..... | 23 |
| 圖表 27..... | 24 |
| 圖表 28..... | 25 |
| 圖表 29..... | 26 |
| 圖表 30 計算完成後顯示出根據過去 Q 值找到的最佳路徑..... | 26 |

1. 研究目的

現今室內定位的方法有多種，常見的有 WiFi 定位，藍芽定位，ZigBee 等等，上述三種皆須依據場地規模佈署相對的設備，其中以 WiFi 成本最低，但精準度不高，其二則需要一定的成本，以下圖表針對各項技術做比較：

| 定位技術 | 精準度 | 穿透性 | 抗干擾性 | 部局複雜度 | 成本 |
|----------|-------|------|-------|-------|-------|
| 藍芽定位 | ★★★ | ★★★ | ★★ | ★★★ | ★★★ |
| Wi-Fi定位 | ★ | ★★★ | ★★★★★ | ★ | ★ |
| ZigBee定位 | ★★ | ★★★★ | ★★★ | ★★ | ★★★ |
| RFID定位 | ★★★★★ | ★★★ | ★★ | ★★ | ★★ |
| 超音波定位 | ★★★★★ | ★ | ★★★ | ★★ | ★★★★★ |

圖表 1 各項定位技術比較

本文以大學校園為背景，以電腦視覺領域為研究，欲發展一套通用於各個場景的定位系統，對於佔地不小的大學校園以及分布廣泛的多層樓教學大樓，佈署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，再透過行動裝置影像辨識地點達到定位，並透過馬可夫決策過程統計使用者路線，大量資料透過 Q-Learning 學習，訓練出使用者最佳路徑。後續章節安排如下：第二章介紹背景與相關知識，第三章則講述系統設計與研究方法，第四章展示場景辨識系統並且說明路徑訓練的實驗結果，第五章為結論與未來展望。

2. 背景與相關知識

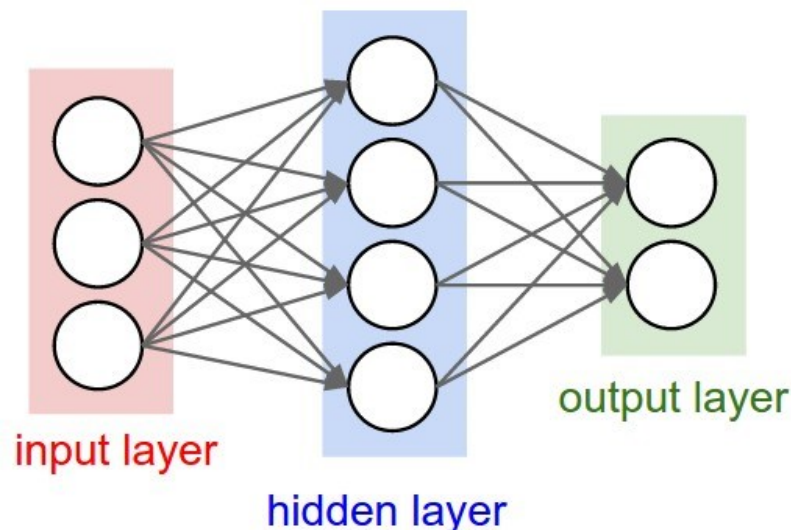
近年來計算機運算能力大幅提升，促使過往一度被捨棄的類神經網路死而復甦，發展為現在深度學習的核心概念，更因為深度學習的靈活運用，在人工智慧，電腦視覺，自然語言處理各種領域上面有更大突破。本章節介紹深度學習與 Q-Learning。

2.1. 深度學習

深度學習(Deep Learning)為機器學習的其中一個分支，由多層神經網路構成，是機器學習中一種針對數據進行特徵學習的方法。能夠透過長期的經驗從中學習當中的知識，當遇到問題時，能依過去的經驗給予對應的結果。至今已有數種深度學習框架，如深度神經網路 (Deep Neural Networks, DNN)、卷積神經網路 (Convolutional Neuron Networks, CNN)和深度置信網路(Deep Belief Networks, DBN) 和遞歸神經網路 (Recurrent Neural Network) 等等，已被應用於遊戲 AI、計算機視覺、語音識別、自然語言都有驚人的成果。

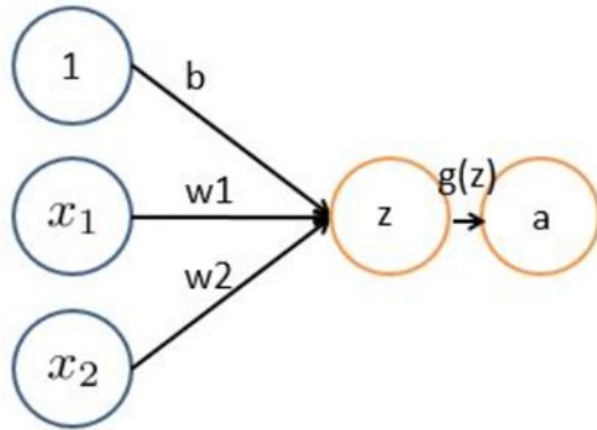
2.1.1 類神經網路

類神經網路(Neuron Networks)由大量神經元交互連結以分層概念構成，如圖表 2 所示，簡單的分為輸入層，隱藏層與輸出層。



圖表 2 神經網路

各神經元接受線性組合輸入後，起初是簡單的線性加權，之後各神經元加上了非線性激活函數，進行非線性轉換後輸出，輸出又成為下一層神經元的輸入。連結代表加權值，稱為權重(weight)。不同的權重和激活函數(Activation Function)，影響著神經網路的輸出。神經網路的每個神經元如圖表 3：



圖表 3 神經元

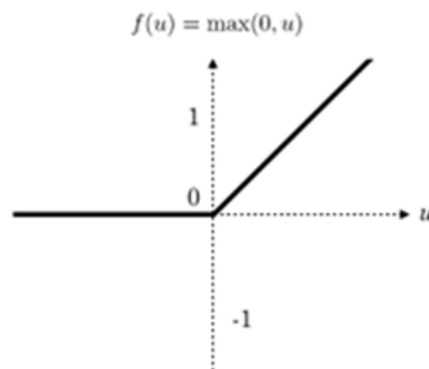
圖為基本 $w \cdot x + b$ 的形式，其中 x_1 、 x_2 ，表示輸入向量， w_1 、 w_2 ，為權重，幾個輸入則意味著有幾個權重，即每個輸入都被賦予一個權重， b 為偏置偏差 (bias)， $g(z)$ 為激活函數， a 為輸出。

2.1.2 激活函數

激活函數為神經元將輸入值加總計算後進行簡單分類的函數，通常為非線性函數。常用的非線性激活函數有 sigmoid, tanh, ReLU 等等，前兩者 sigmoid / tanh 比較常見於全連接層，後者 ReLU (Rectified Linear Unit) 常見於卷積神經網路的卷積層。這裡簡介 ReLU 函數，ReLU 的函數定義為：

$$f(x) = \max(0, x)$$

透過以往腦神經研究發現大腦神經需接收一定的刺激量，神經元才會激活，平時則處於抑制狀態，相較於 sigmoid，ReLU 函數能夠更精準的表達。卷積神經網路透過 ReLU 能夠解決訓練時梯度消失的問題，並且降低計算量，函數只需判斷輸入是否大於 0 即可輸出。



圖表 4 ReLU 函數，優點為速度快，減輕梯度消失問題還有稀疏性

2.1.3 Softmax 回歸

Softmax 回歸為 Logistic 回歸的推廣，相較於 Logistic 回歸的二分類，Softmax 可處理兩種類別以上的分類問題。將標籤 y 分為 k 類，訓練集為：

$\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ ，而 $y^{(i)} \in \{1, 2, \dots, k\}$ ，表示有 k 種類別。函數估計 x 對於每個類別 j 的分類結果的概率 $p(y = j|x)$ ，而此函數將輸出 k 維向量，表示 k 個概率，此函數 $h_\theta(x)$ 可由下式表示：

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1|x^{(i)}; \theta) \\ p(y^{(i)} = 2|x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k|x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

其中 $\theta_1, \theta_2, \dots, \theta_k \in \mathbb{R}^{n+1}$ 為模型參數，而 $\frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}}$ 對機率分布正規化，使機率總和為 1。

2.1.4 反向傳播算法

反向傳播算法(Backpropagation)通常與最優化方法（如梯度下降法）結合使用，為訓練神經網路的方法。該方法計算對網路中所有權重計算損失函數的梯度。此梯度會反饋給最優化方法，用來更新權值以最小化損失函數。反向傳播要求每個輸入對上已知輸出，計算損失函數梯度。以下為反向傳播算法步驟：

1. 進行前饋傳導計算，算出每一隱藏層神經元的激活值
2. 對於輸出層 n_l 的每個神經元 i ，根據以下公式計算誤差值：

$$\delta_i^{(n_l)} = \frac{\partial}{\partial z_i^{(n_l)}} \frac{1}{2} \|y - h_{W,b}(x)\|^2 = -(y_i - a_i^{(n_l)}) \cdot f'(z_i^{(n_l)})$$

3. 對 $l = n_l - 1, n_l - 2, n_l - 3, \dots, 2$ 的各個層，第 l 層的第 i 個節點的誤差計算方式如下：

$$\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_j^{(l+1)} \right) f'(z_i^{(l)})$$

以上逐次從後向前求導的過程即為「反向傳導」。

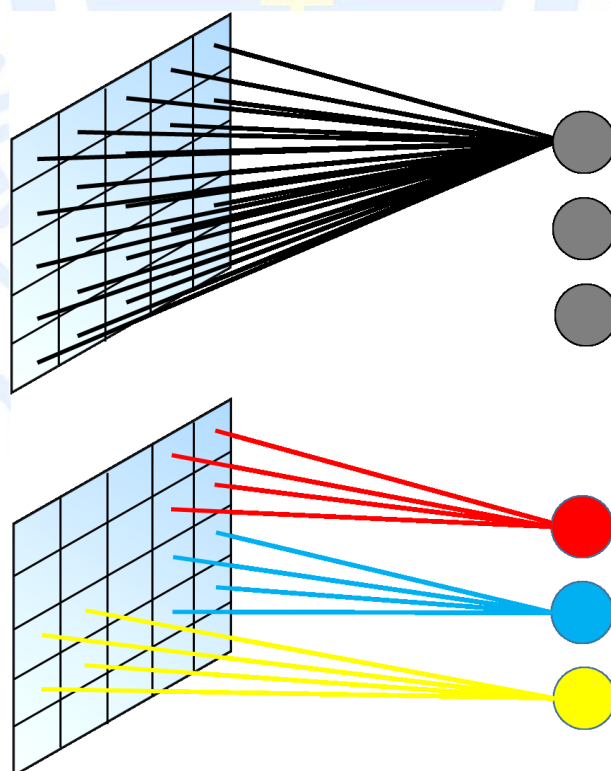
4. 計算偏導數，計算如下：

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x, y) = a_j^{(l)} \delta_i^{(l+1)}$$
$$\frac{\partial}{\partial b_i^{(l)}} J(W, b; x, y) = \delta_i^{(l+1)}.$$

2.1.5 卷積神經網路

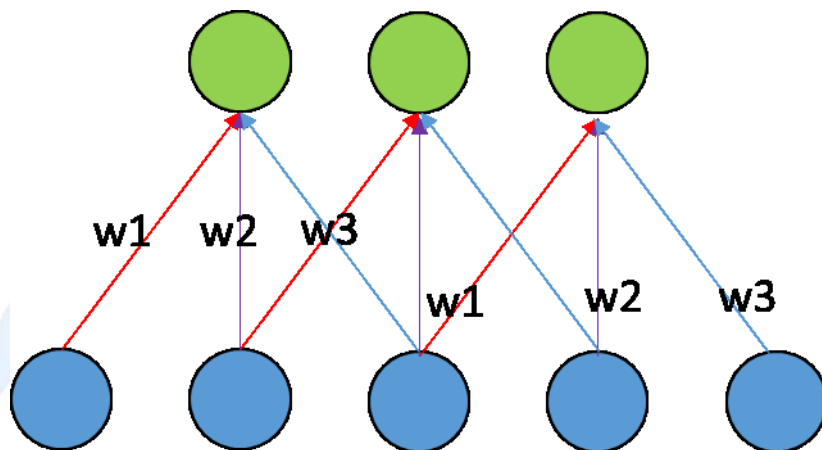
卷積神經網路(Convolutional Neural Network, CNN)為深度學習中具代表性的網路結構之一，目前已被廣泛應用於影像處理領域。傳統神經網路對於圖片的輸入都是採用全連接的方式，即輸入層到隱藏層的神經元都是全部連接的，每個神經元將會連接所有圖片像素，因此導致參數量過大，使得網路訓練耗時甚至無法收斂。CNN 以下列兩個概念大幅度降低所需訓練的參數。

1. 局部連接：以生物視覺為想法為概念，以人類視覺來說，觀看一影像只需注視一部分影像即可了解此畫面，不須全畫面掃描。以此概念延伸，每個神經元不須和上一層的所有神經元連結，只和一小部分神經元相連，瞬間減少了大量參數。圖表五上每個神經元對應到 5*5pixel 圖片的每個 pixel，則 n 個神經元所需要訓練的參數為 5*5*n。圖下採局部連接，捲積核只連接 2*2 的 pixel 圖片，因此需要訓練的參數為 2*2*n。



圖表 5 上圖為單一神經元連結所有像素，下圖使用局部連結，每神經元作為 2*2 的 Filter

2. 權值共享：卷積核和圖片的连接可以共享同一個權重，而不是圖片每個连接各有不同的權重，換句話說，一張圖片對一卷積核來說權重都是一樣的，這樣又減少了很多參數。以圖表 6 為例，將整張圖片對一特徵卷積，藍色為圖片 pixel，綠色為卷積核，由於卷積特徵固定，因此權重皆相同。

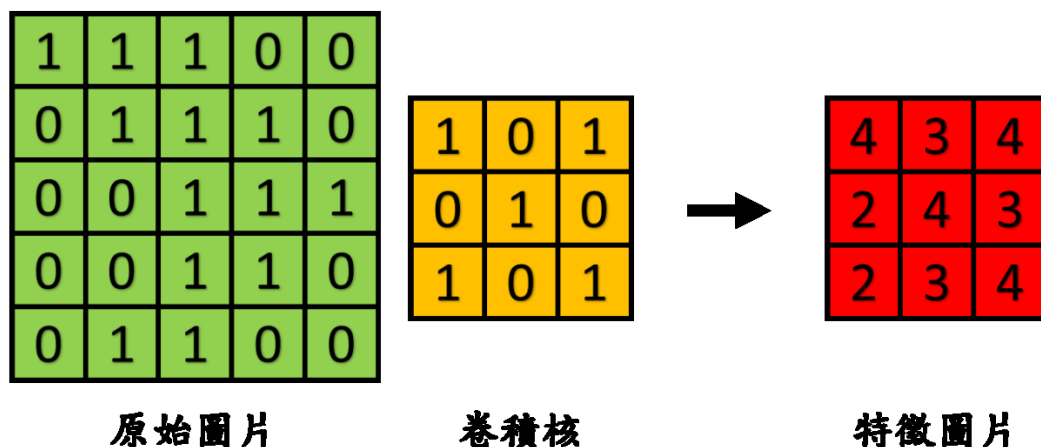


圖表 6 藍色為輸入，綠色為卷積核

卷積核由以上兩個概念設計，也可將卷積核視為圖片的濾波器(Filter)。卷積核根據其參數能夠提取圖片局部特徵，這即是卷積神經網路識別圖片的基本原理。

卷積神經網路兩個主要層面計算，卷積層與採樣層。

卷積層：是卷積核在上一輸入層通過逐一計算卷積，卷積核中的每一個參數都相當於傳統神經網路中的權值參數，與對應的局部像素相連，將卷積核的各個參數與對應的局部像素值相乘之和，得到卷積層上的結果，如圖表 7 所示。

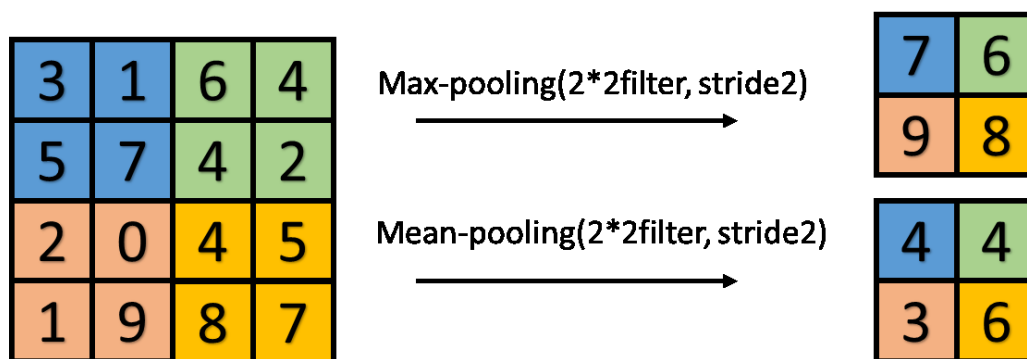


圖表 7 特徵卷積計算

採樣層：通過卷積層獲得了圖像的特徵之後，理論上可以直接使用這些特徵訓練分類器，但是這樣做將面臨巨大的計算量的挑戰，而容易產生過擬合的現象。為了進一步降低網路訓練參數及模型的過擬合程度，在對卷積層進行池化/採樣

(Pooling) 處理池化/採樣的方式通常有以下兩種：

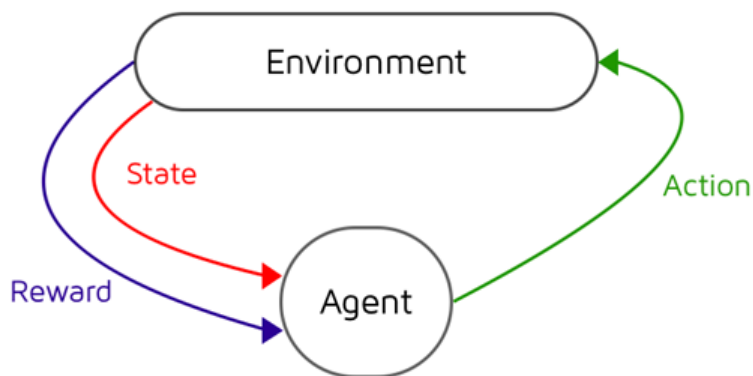
- Max-Pooling：根據 hyperparameter(filter 和 stride 的大小)選擇窗口最大值作為採樣值。
- Mean-Pooling：根據 hyperparameter(filter 和 stride 的大小)將窗口平均值作為採樣值。



圖表 8 採樣計算

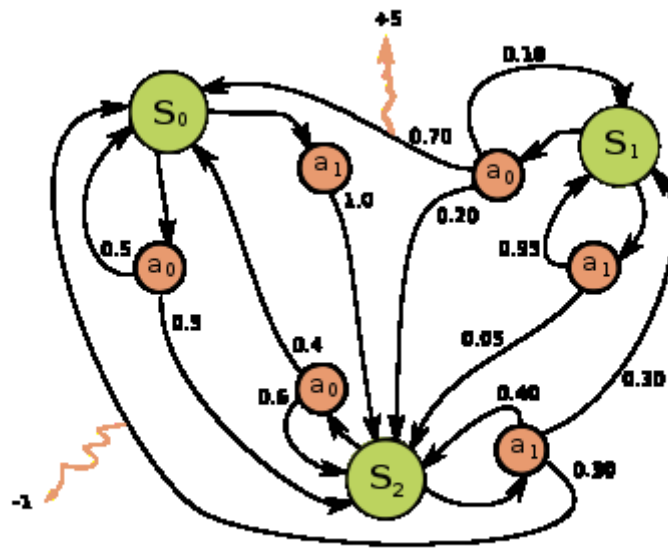
2.2 Q-Learning

Q-Learning 為增強式學習(Reinforcement Learning)當中常用的演算法，是一個建構於馬可夫決策過程 (MDP) 的遞增式動態規劃演算法。增強式學習為透過不斷的探索環境並採取對應動作，根據採取的動作獲取獎勵訓練 Agent，圖表 9 為增強式學習概念。



圖表 9 強化式學習概念

Agent 採取 Action，Environment 回傳對應 State 與 Reward 給 Agent，如此不斷重複訓練即為強化式學習。環境所有的狀態與獎勵對應關係以 MDP 的方式表現，如圖表 10 所示：



圖表 10 馬可夫決策過程

當到達不同狀態時，環境會給予對應獎勵。MDP 以 (S, A, P_{sa}, R) 四個元素構成：

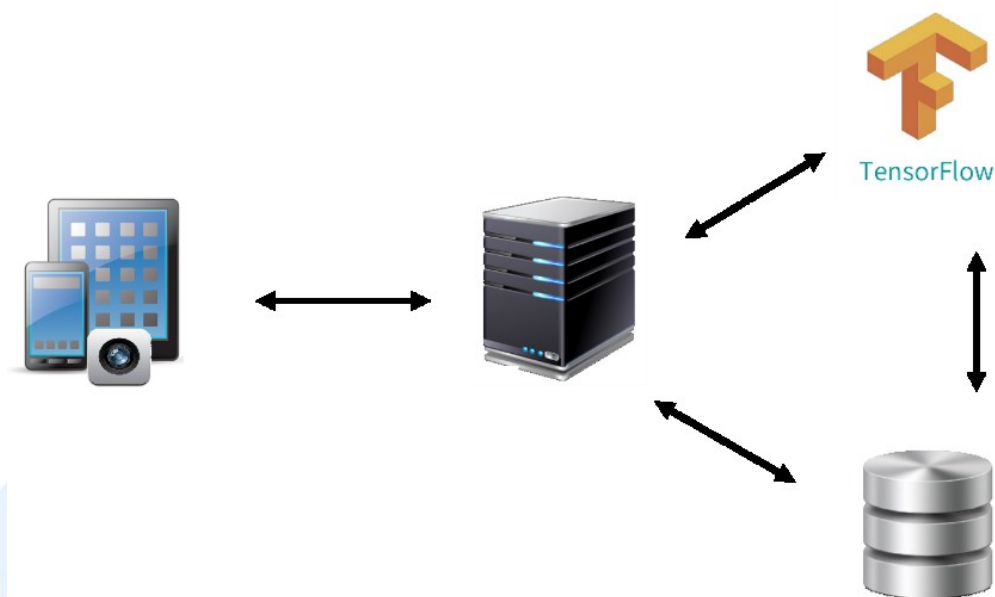
- S ：表示狀態集(states)， $s \in S$ ， s_i 表示第 i 步的狀態
- A ：表示一組動作(actions)，有 $a \in A$ ， a_i 表示第 i 步的動作。
- P_{sa} ：表示狀態轉移概率。 P_{sa} 表示的是在當前 $s \in S$ 狀態下，經過 $a \in A$ 作用後，會轉移到的其他狀態的概率分佈情況。例如在狀態 s 下執行動作 a ，轉移到 s' 的概率可以表示為 $p(s'|s, a)$ 。
- R ： $S \times A \rightarrow \mathbb{R}$ ， R 是獎勵函數(reward function)。有些獎勵函數狀態 S 的函數，可以簡化為 $R: S \rightarrow \mathbb{R}$ 。若一組 (s, a) 轉移到了下個狀態 s' ，則獎勵函數可記為 $r(s'|s, a)$ 。如果 (s, a) 對應的下個狀態 s' 是唯一的，那麼獎勵函數也可以記為 $r(s, a)$ 。

Q-learning 會記錄每個狀態下，執行所有可執行動作能獲得的預期獎勵，以下列公式更新所記錄的預期獎勵：

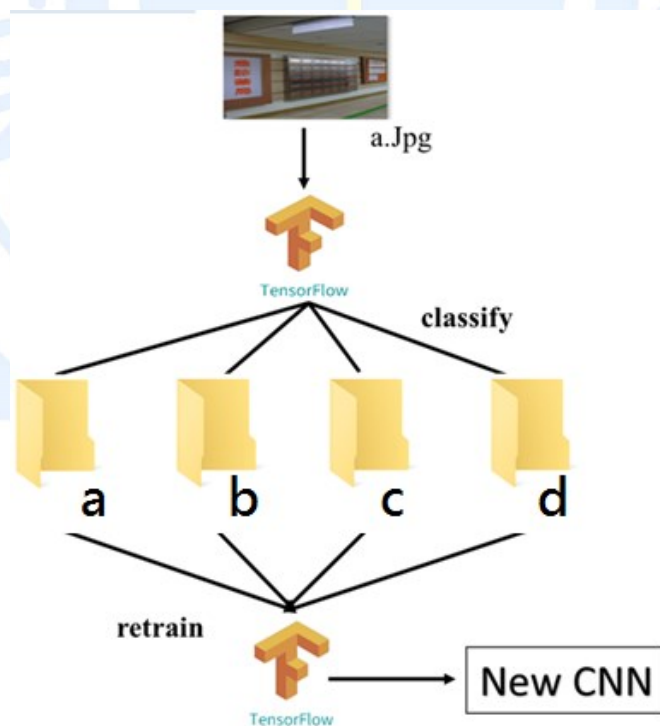
$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha_t}_{\text{learning rate}} \cdot \left(\underbrace{r_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right)$$

3. 系統架構與方法

系統主要由 Client 與 Server 構成，Client 部分針對行動裝置設計，主要功能為即時地點辨識，路徑導航，路徑資料紀錄。Server 提供 CNN 模型的訓練與更新 CNN 至手機 APP，圖片與路徑資料庫的儲存，路徑訓練。



圖表 11 系統架構



圖表 12 圖片辨識分類與模型訓練

當圖片傳送至 server，後端程式執行場景辨識，並且將圖片分類至該場景圖片資

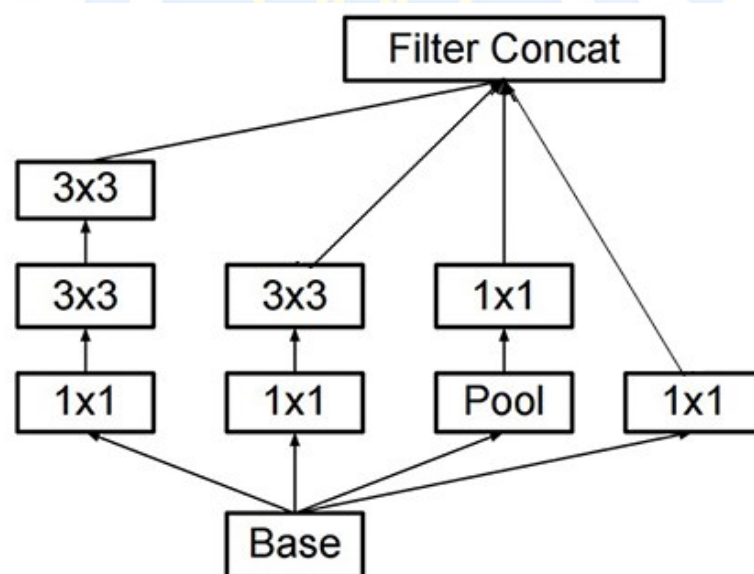
料夾，如此一來能方便之後的監督式學習(Supervised learning)訓練 CNN。後端背景程式會定期 retrain 產生新的 CNN 模型，最後再更新模型至 client APP。以上流程當擁有大量使用者時將能隨著使用量多寡強化本系統的辨識能力。

3.1 圖片資料集

由於透過使用者的上傳照片沒辦法短時間建立場景資料庫，因此必須預先收集大量圖片集，收集由人員於場地內行走時，透過手機拍攝影片取得。隨後將影片以每秒 1 張的方式圖片分割，將整組圖片標籤為該地點，利於之後系統的監督式學習。

3.2 CNN 架構

本文採用 GoogLeNet(InceptionV3)作為 CNN 模型。GoogLeNet 在 ILSVRC2014 電腦視覺競賽中提出，獲得了 ILSVRC2014 的冠軍。該網路的特點提升了計算資源的利用率，可以在網路計算資源不變的前提下，通過微調的設計增加網路的寬度與深度，基於 Hebbian 法則和多尺度處理來優化性能。GoogLeNet 共 22 層網路，用的參數比 ILSVRC2012 冠軍 AlexNet 少 12 倍，但準確率更高。GoogLeNet 由 Inception module 構成，圖為 Inception module 的結構：

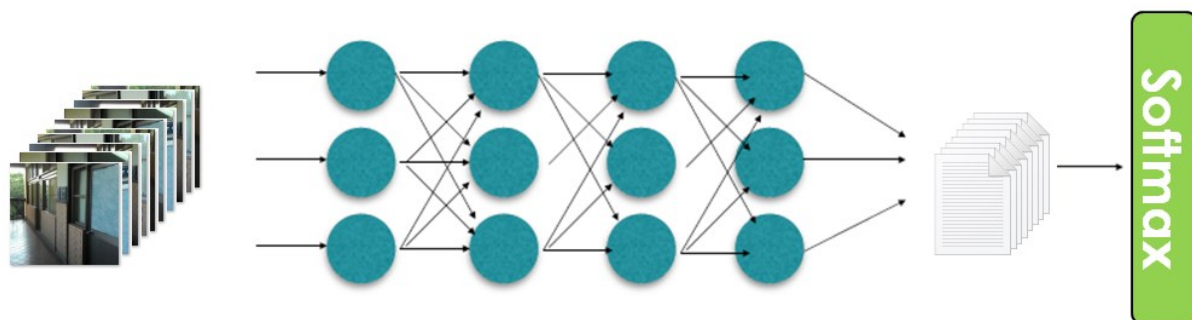


圖表 13 Inception V3 模組

此模塊由 1×1 、 3×3 卷積核的組成，而 1×1 卷積核正為此模組最重要的概念之一，能夠減少後續並行操作的特徵數量，且不壓縮圖片大小。這種概念稱作“bottleneck layer”，簡單來說就是在重要的訓練之前，降低參數而特徵卻不失真的方法。bottleneck 將在下章節重新訓練模型的最後一層分類器中用到。

3.3 CNN 訓練程序

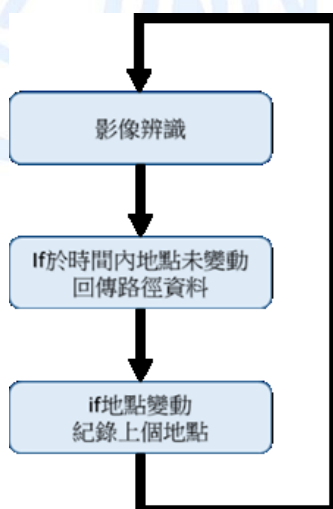
如之前所說，我們採用已經預先透過 ImageNet 訓練好的 InceptionV3 作為 CNN 模型，ImageNet 圖片庫龐大，因此預先訓練好的 InceptionV3 從中已經學習大量特徵，也就是說卷積核的參數大致上以訓練完成，已經能夠辨識大部分圖片，接著我們在藉由我們處理過並分類的圖片訓練最後一層 Softmax 分類器，這樣一來便能辨識與分類地點圖片。在訓練最後一層網路前會先計算圖片的 bottleneck，也就是依原本 InceptionV3 分類出來的值，訓練分類器時為了加速過程避免重複計算 bottleneck，會將計算好的 bottleneck 存檔方便之後的調用，最後開始進行最後一層 softmax 分類器的訓練。



圖表 14 CNN 訓練流程

3.4 路徑資料收集

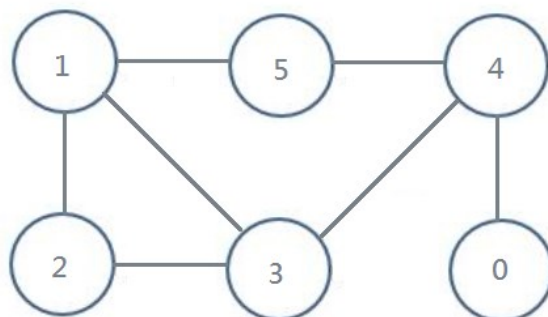
當 CNN 模型完成訓練並置入手機即時辨識 APP，人員於地點行走並使用手機 APP 辨識地點，辨識後得到的結果以路徑方式記錄下來，當收集完成則回傳至 Server 路徑資料庫。例如：使用者路徑為 A>B>F>D>C，當到地點 C 10 分鐘沒變動，則回傳該筆路徑資料 A>B>F>D>C 至資料庫，若 10 分鐘後變動，則重新記錄新路徑。



圖表 15 路徑辨識流程

3.5 Q-Learning 設計

當擁有足夠的路徑資料，我們將從大量的資料裡找出使用者常走的路線，這個段落將說明 Q-Learning 的設計，圖表 16 為 6 個地點的範例地圖



圖表 16

以範例的 6 個地點來說，每個地點都有 6 個目的地，所以共有 6*6 個起點到目的地的對應關係。根據各地點連接的關係建立 6 個 Reward 矩陣 R0~R5 和 6 個 Value 矩陣 Q0~Q5=0 如圖表 17：

| | | Action | | | | | |
|-------|---|--------|----|----|----|----|-----|
| State | | 0 | 1 | 2 | 3 | 4 | 5 |
| R = | 0 | -1 | -1 | -1 | -1 | 0 | -1 |
| | 1 | -1 | -1 | -1 | 0 | -1 | 100 |
| | 2 | -1 | -1 | -1 | 0 | -1 | -1 |
| | 3 | -1 | 0 | 0 | -1 | 0 | -1 |
| | 4 | 0 | -1 | -1 | 0 | -1 | 100 |
| | 5 | -1 | 0 | -1 | -1 | 0 | 100 |

| | | 0 | 1 | 2 | 3 | 4 | 5 |
|-----|---|---|---|---|---|---|---|
| Q = | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 5 | 0 | 0 | 0 | 0 | 0 | 0 |

圖表 17 R5 矩陣，Q5 矩陣

Q-Learning 演算法

1. 給定折扣率 γ 和矩陣 Q 與 R
2. for 資料庫每筆路徑 {
 - 令起始狀態= S_i 目標狀態= S_o
 - if($S_i \neq S_o$) {
 - 令此路徑下個點為下個狀態 S_i' (表示執行動作 a)
 - 計算 $Q(s,a)$ 並更新至矩陣 Q
 - 令 $S_i = S_i'$

當執行足夠的訓練，矩陣 Q 收斂後，便能根據矩陣內給予較大的獎勵值選出最佳路徑。

4. 系統實作與實驗結果

ImageNet 訓練好的 InceptionV3 已包含大量圖片的特徵權重，但 ImageNet 圖庫中並未針對場景的圖片分類，故 InceptionV3 並無相關的特徵權重。且一個由多種物件所集合之地點是否也能正確辨識是值得探討的，本文實際收集東海大學 7 個地點的影像進行實驗，系統與實驗環境如圖表 18

圖表 18 實驗設備與開發環境

| | |
|---------------|--|
| OS | Ubuntu14.04 ,Android6.0 |
| CPU | AMD Athlon ii x4 640 processor 3.0 Ghz |
| Memory | 16GB DDR3 1600 |
| Camera | ASUS Zenfone 6 A600CG |

東海大學場景辨識器

img/56Ns.jpg



[首頁](#)

以下為圖片辨識預測機率

input:

56Ns.jpg

output:

the luce chapel (score = 0.98289)

library (score = 0.00527)

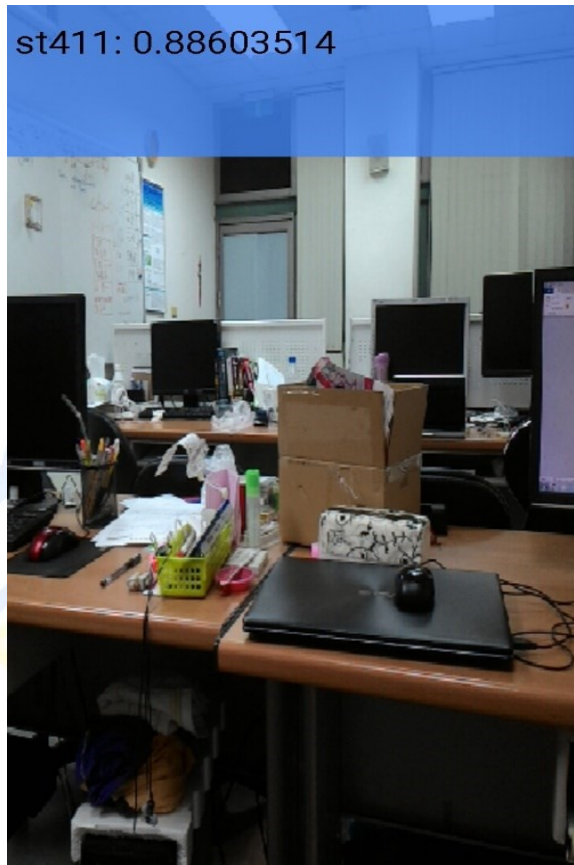
7 11outside (score = 0.00496)

st (score = 0.00316)

memorial hall (score = 0.00143)



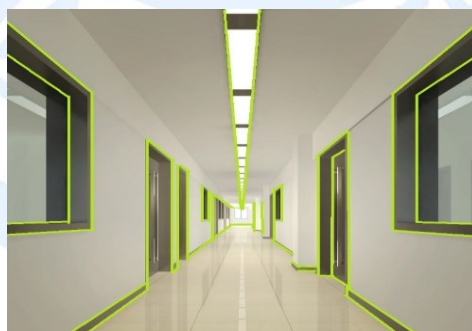
圖表 19 網頁使用介面



圖表 20 及時辨識 APP

4.1 深度學習影像辨識地點之辨識正確率

圖表 21 為走廊的影像，若是將地點中固定建築體如走廊、門窗、燈飾、擺飾等視為物體的特徵，將地點視為一個物件那麼深度學習對於地點之辨識是可行。



圖表 21 走廊特徵示意圖

如以下圖表 22 實測東海大學 7 個地點的影像辨識正確率，依地點大小不同拍攝若干張照片進行 10000 次訓練，實驗結果其正確率平均有 97% 以上，數據結果支持深度學習影像辨識可有效辨識地點。

圖表 22 地點辨識正確率

| 場景地點 | 圖片數量 | 測試次數 | 訓練次數 | 平均正確率 | 執行時間 (s) |
|-----------|------|------|-------|----------|----------|
| ST411 實驗室 | 79 | 4 | 10000 | 0.977518 | 4.71925 |
| ST436 教室 | 90 | 4 | 10000 | 0.987625 | 4.773 |
| ST 四樓廁所 | 37 | 4 | 10000 | 0.98701 | 4.7653 |
| ST 大樓 | 161 | 4 | 10000 | 0.996063 | 4.87525 |
| 教堂 | 110 | 4 | 10000 | 0.995223 | 4.8315 |
| 福利社 | 191 | 4 | 10000 | 0.986603 | 4.78875 |
| 校園側門 | 72 | 4 | 10000 | 0.991705 | 4.7075 |

4.2 深度學習影像辨識各影像大小與解析度之辨識時間

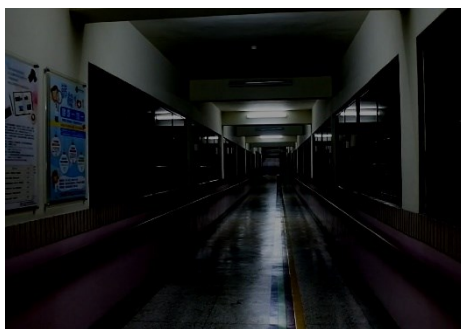
使用者在使用導航系統時自身會不停移動，為求能正確辨識使用者當下所在的位置，辨識時所花費的時間不能過高。本文分析辨識各種尺寸與不同解析度之影像時，所耗費之時間。

圖表 23 辨識執行時間與正確率

| 場景地點 | 圖片比例 | 正確率 | 執行時間(s) | 圖片比例 | 正確率 | 執行時間 (s) |
|-----------|------|---------|---------|------|---------|----------|
| ST411 實驗室 | 25% | 0.88395 | 4.795 | 75% | 0.99505 | 4.905 |
| ST436 教室 | 25% | 0.6604 | 4.68 | 75% | 0.95584 | 4.678 |
| ST 四樓廁所 | 25% | <0.5 | 4.713 | 75% | 0.99505 | 4.658 |
| ST 大樓 | 25% | <0.5 | 4.726 | 75% | 0.9593 | 4.876 |
| 教堂 | 25% | <0.5 | 4.738 | 75% | <0.5 | 4.695 |
| 福利社 | 25% | 0.94777 | 4.79 | 75% | 0.9845 | 4.844 |
| 校園側門 | 25% | <0.5 | 4.837 | 75% | 0.86625 | 4.67 |

將影像尺寸縮放至 75%後平均值行時間為 4.75 秒，正確率為 0.884(低於 0.5 以 0.5 計算)，平均值行時間提高 0.09 秒並無顯著差異。縮放至 25%後平均值行時間為 4.71 秒正確率降至 0.693 以下。實驗結果分析降低尺寸無法提高辨識時間且會降低正確率。

4.3 深度學習影像辨識對開方式地點日夜間之辨識正確率



圖表 24 模擬夜間走廊影像

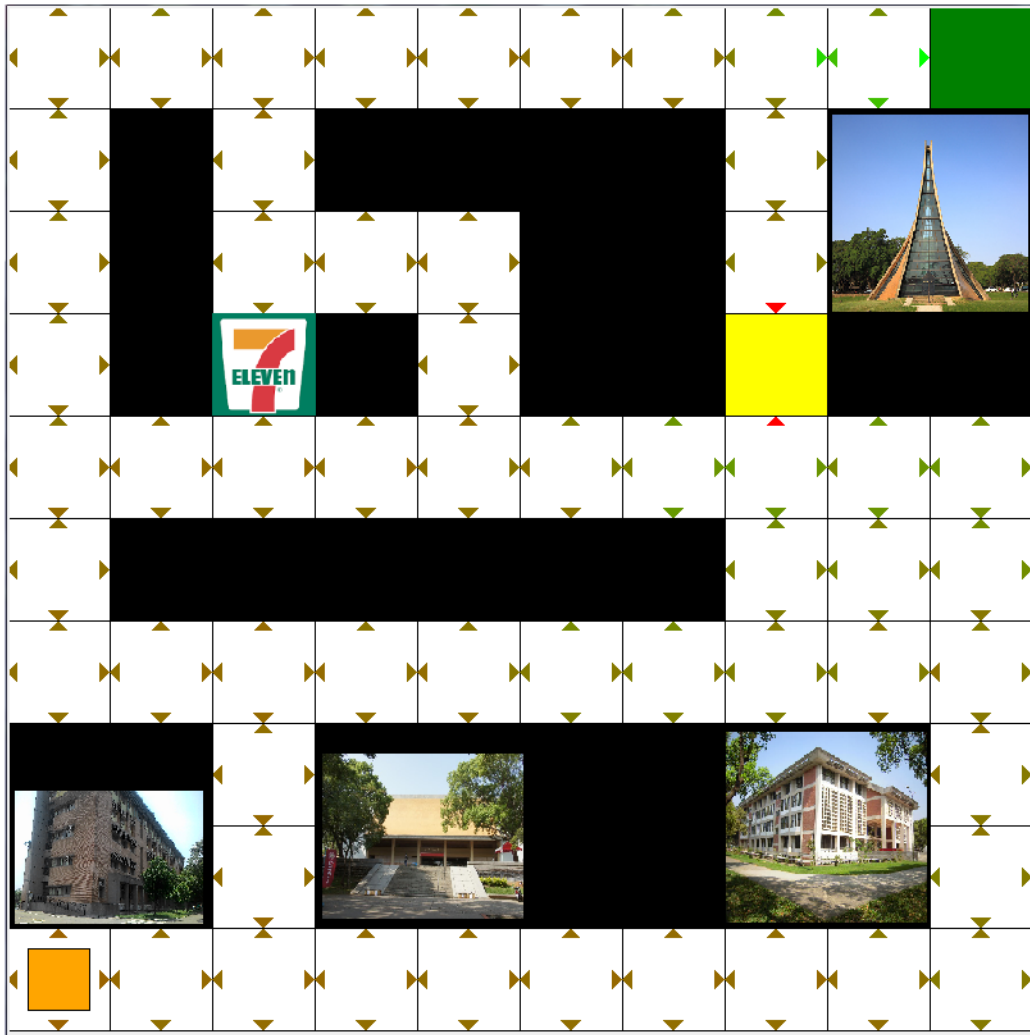
對於開放式地點(如走廊、停車場)，會有夜間光線不足造成影片模糊失真之情況，以下本文測試在不同光線強度下，影像辨識的正確率。在光線對比降低 50%的影像中平均正確率為 0.9314，降低至 100%後平均正確率為 0.8287。如圖表 22 為光線對比降低 100%之走廊影像，在圖片相當昏暗的情況下此圖片也有高達 0.993 之正確率，依實驗結果分析低光源對於深度學習影像辨識影響差異不大(降低 0.005%)。

圖表 25 降低光線對影像辨識正確率影響

| 地點 | 亮度比例 | 正確率 | 亮度比例 | 正確率 |
|-----------|-------|---------|------|---------|
| ST411 實驗室 | -100% | 0.98509 | -50% | 0.99649 |
| ST436 教室 | -100% | 0.50195 | -50% | 0.98585 |
| ST 四樓廁所 | -100% | 0.54272 | -50% | 0.64026 |
| ST 大樓 | -100% | 0.97743 | -50% | 0.99945 |
| 教堂 | -100% | 0.58785 | -50% | 0.99319 |
| 福利社 | -100% | 0.99745 | -50% | 0.99261 |
| 校園側門 | -100% | 0.86831 | -50% | 0.7756 |

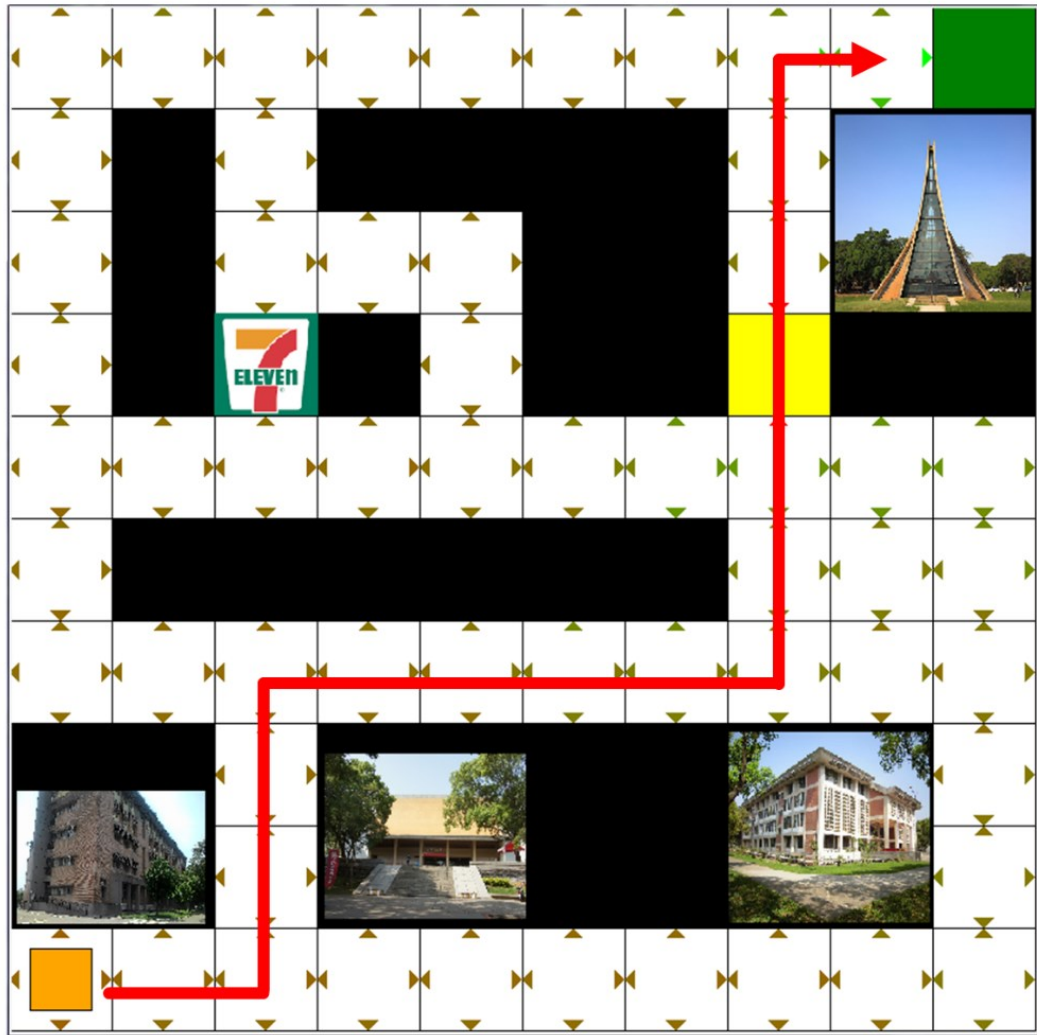
4.4 Q-Learning 實驗

此實驗模擬使用者於環境找尋目的地，以 10*10 狀態空間為模擬環境，每一個方格都為一個地點。左上角定為座標(0,0)右下定為(9,9)，起點為(0,9)，目標終點為(9,0)。



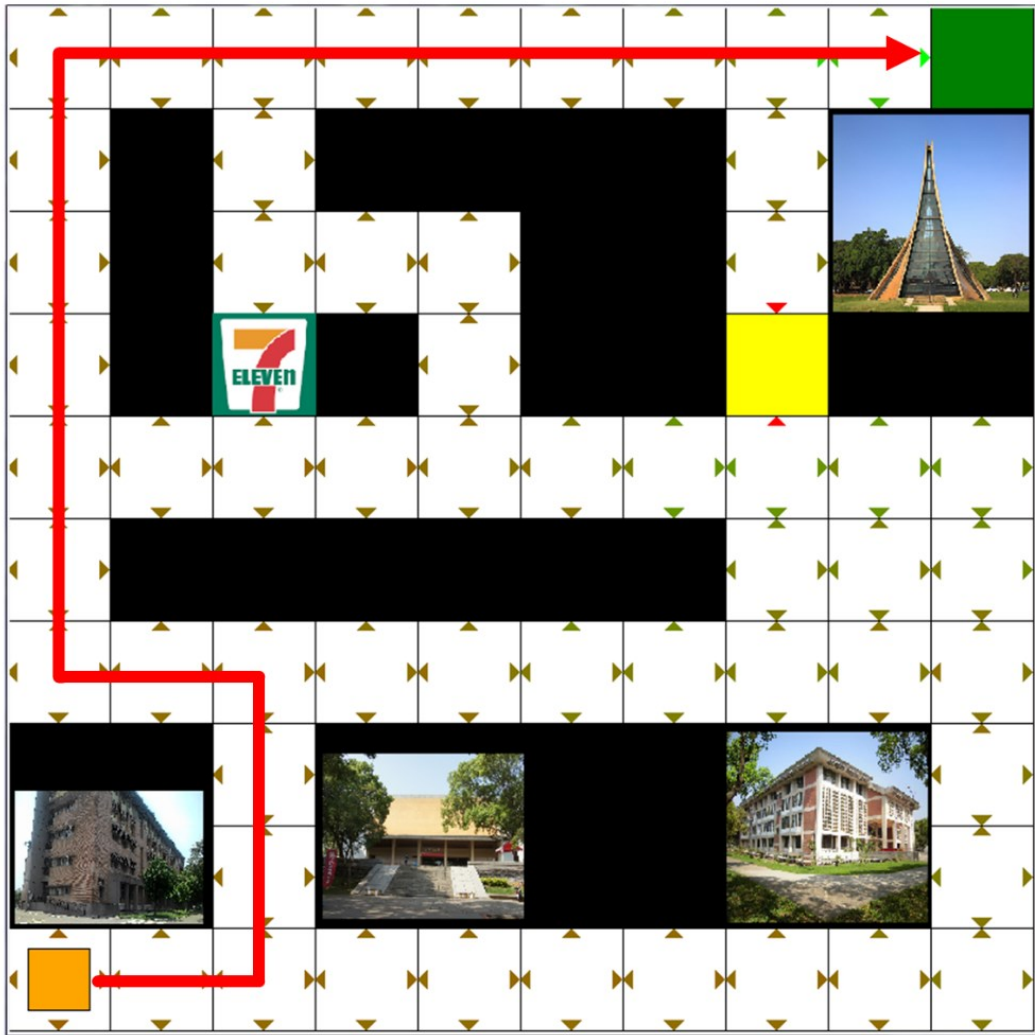
圖表 26 大小 10*10 空間狀態

橘色方格為起始點。綠色方格為目的地，當到達目的地時，給予 agent 正獎勵。黃色方格代表各種可能影響到達目的地之因素，例如障礙物、人潮壅擠等等因素，經過則給予 agent 負獎勵。



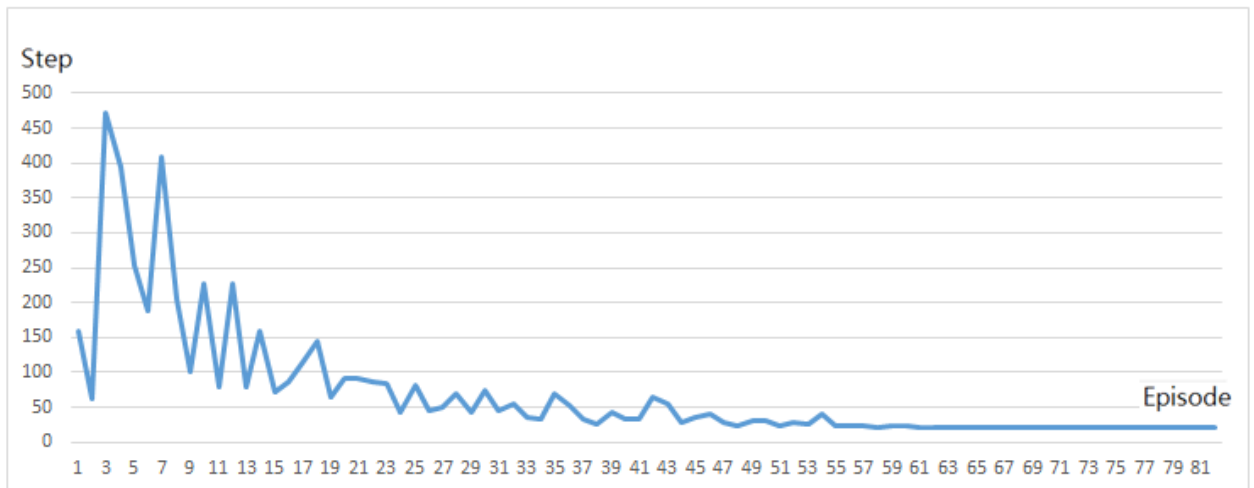
圖表 27

圖表 27 為訓練初期尚未完全探索整張地圖，agent 所計算之較佳路徑。此路徑雖然以距離來說屬於最近的路徑，但由於經過黃色方格，可能影響行徑時間，因此以整體來說可能不是最佳路徑。



圖表 28

圖表 28 為當訓練達收斂時，agent 所計算出最佳路徑，此路徑避開可能影響行走的點，為成本最低之路徑



圖表 29

圖表 29 顯示實驗結果，縱軸 Step 為 agent 到達目的花費之步數，橫軸 Episode 為訓練回合數。上圖當 Episode 於 20 之前花費成本波動幅度較大，最多花費將近 470 步才找到目標，之後步數成本逐漸下向。而在 Episode 於 50 之後，逐漸穩定於最短路徑(最短路徑為耗費 22 步之路徑)。

```

使用者位置
<0, 9>
路徑
[<0, 9>, <1, 9>, <2, 9>, <2, 8>, <2, 7>, <2, 6>, <1, 6>, <0, 6>, <0, 5>, <0, 4>,
<0, 3>, <0, 2>, <0, 1>, <0, 0>, <1, 0>, <2, 0>, <3, 0>, <4, 0>, <5, 0>, <6, 0>,
<7, 0>, <8, 0>, <9, 0>]
目的地
<9, 0>

```

圖表 30 計算完成後顯示出根據過去 Q 值找到的最佳路徑

5. 結論與未來展望

本文實現一套透過深度學習達成場景辨識的系統，依據本研究實驗結果，當訓練樣本足夠時，深度學習可以有效地辨識地點，且在夜間光線昏暗時也有一定的辨識率。平均約 4.8 秒的辨識耗費時間是值得討論與改善的地方，若要將本系統用於行車導航明顯效能不足，但是對於以步行為主的大型室內空間例如校園、博物館、大賣場等相當適合本系統的運用。另一項優勢即是系統建置完善後不需使用網路或 wifi 便可導航，在網路訊號不佳的地點也相當適合影像導航。未來希望結合 AR 擴增實境使系統更便利與人性化。



參考文獻

- [1] 陳信宇,「機器人之增強式學習:整合遺傳規劃與類神經網路」,國立雲林科技大學資訊工程研究所碩士論文,民國99年6月。
- [2] 曹瑋宸,「基於深度學習之性別辨識與人員計數」,國立雲林科技大學資訊工程研究所碩士論文,民國104年6月。
- [3] 魏子量、吳政翰、林忠宏、林泓宏,「基於深度學習之魚種辨識」,國立成功大學系統及船舶機電工程學系,南臺科技大學資訊工程系,2016 Conference on Information Technology and Applications in Outlying Islands
- [4] M. K. Alsmadi and K. B. Omar and S. A. Noah and I. Almarashdeh, "Fish recognition based on robust features extraction from size and shape measurements using neural network," Journal of Computer Science, vol.6, no.10, pp.1088, 2010.
- [5] Hsien-I lin, Wei-Kai Chen, "Human Intention Recognition using Markov Decision Processes", Automatic Control Conference (CACs), 2014 CACS International
- [6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna, "Rethinking the Inception Architecture for Computer Vision", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818-2826
- [7] Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report (2009)
- [8] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors (2012). arXiv:1207.0580
- [9] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
- [10] N. Kaggle CIFAR-10 (5th) [Online]
<https://github.com/nagadomi/kaggle-cifar10-torch7>