

2022 年臺灣國際科學展覽會 優勝作品專輯

作品編號 190022
參展科別 電腦科學與資訊工程
作品名稱 彩色二維條碼手持產品開發之探討
得獎獎項 四等獎

就讀學校 國立嘉義高級中學
指導教師 黃冠夫
作者姓名 李艾登

關鍵詞 影像辨識、彩色二維條碼、機器學習

作者簡介



進入嘉義高中就開始我的科展，我的題目中文資源非常少，幸好我是最勁爆的書呆子，把我讀過的英文書疊起來，你就可以爬上銀河系了。研究期間我花了3000小時在網路上搜尋資料，上課與讀論文，事後英文獲得大進步，雅思成績8分。

《唐三藏為甚麼不搭舢斗雲去西天取經》據說在對岸有2億次點閱率，是我和同學的傑作。

摘要

QR Code 是由黑白模組組成的二維數位條碼，掃描後可讀取儲存的訊息。受限於設計原理，QR Code 使用二進位制儲存資料。增加模組數目可增加資料量，但若在條碼內塞進太多模組時，尺寸太小的模組將無法被掃描器讀取。此外，目前 QR Code 掃描器僅支援單張掃描，並無法應付同時多張條碼掃描的實務需求。

如能克服顏色辨識，理論上彩色二維條碼將能克服現行 QR Code 的限制，但市面上並無相關產品可供測試。因此本專題設計了一款 10×10、具 8 顏色的"Colour Matrix"，並利用 Raspberry Pi 開發 Colour Matrix 在手持裝置上運作的軟硬體來進行實驗。此實驗成功利用機器學習演算法在 Raspberry Pi 上進行的顏色辨識。開發的程式在單張掃描上效能與使用 pyzbar 辨識 QR Code 相當；在多張掃描方面，使用 pyzbar 辨識 QR Code 的解碼成功率為 3.1%，而本專題的方法將成功率提升至 92.4%，擴增數位條碼的使用範圍，具商用價值。

ABSTRACT

A Quick response Code (QR Code) is a scannable two-dimensional barcode with black and white modules that encode data in the binary form. There is limitation in the storage capacity of a QR Code because the amount of data in a QR Code depends on the number of modules while the maximal number of modules is limited by the scanner's resolution. Moreover, there is a practical need for multiple QR Code reading the commonly used QR Code method does not support. Currently, QR Code scanner reads one code from one image and is not able to scan multiple codes at the same time.

It is reasonable that digital code with coloured modules is the solution to solve the shortcomings of QR Code if the colours in the coloured module could be accurately recognized. However, there is no such product available. In this project, a 10×10 eight-coloured code, "Colour Matrix", was designed and a mobile device combining hardware and software solution was built using Raspberry Pi. This proposed method successfully transformed the color information into text using machine learning algorithms. For single two-dimensional code decoding, this Colour Matrix method yielded similar performance to standard QR Code reading by pyzbar. For multiple code decoding, the accuracy was raised from 3.1% by QR Code method up to 92.4% by this Colour Matrix method. The results of this project indicate that colour code is a feasible technique and has the potential for broad application.

一、 前言

(一)、 研究動機

Quick Response Code (QR Code)在二維空間以模組(module)儲存資料，再經由掃描器解碼資料後以文字呈現，被廣泛應用於各行各業。QR Code 資料儲存量取決於模組數，但在 QR Code 內塞進太多模組時，模組可能因尺寸太小而無法被掃描器判讀。增進掃描器解析度及軟體辨識準確度是解決方案之一，我曾在 2020 台北國際自動化工業展中看過如何利用高價的精密儀器準確快速地從視野中找出數個 QR Code 的技術，但這個方案缺點是昂貴，且需在嚴格控制環境光線下進行，實務運作上有困難。

增加模組的變化量也可以提高資料量，Microsoft High Capacity Color Barcode (HCCB)這個產品是利用顏色模組來增加資料密度，這個設計擺脫了 QR Code 的二進位制，將資料儲存方式提昇至 4、8 甚至 16 進位制。但使用顏色模組的致命缺點在於無法克服光線造成的色彩失真，目前為止，就算是 Microsoft 這樣的大公司，仍無法在室外光線下使用彩色條碼，相關的研發工作相繼在 2016 年截止。

精密的二維條碼掃描系統並不是所有的商家、工廠或網拍業者等有能力負擔的，多數業者的需求是利用簡單手持裝置或是手機來增進品管和降低出貨錯誤造成的退貨支出。若能設計出一款可以利用顏色來儲存資料，並且可以用平價手持裝置來讀取資料的彩色二維條碼系統，相信一定能打開另一片尚待開發的藍海市場。因為相信有彩色二維條碼這個需求，我開始投入彩色二維條碼的研究。

(二)、 文獻探討

1、 數位行動條碼發展史

最早是數位條碼是一維條碼(圖 1)，由黑白平行線條組成，掃描後可對應至特定資料。由於具保密性，讀取的錯誤率低，又可快速處理資料，因此被廣泛應用。一維條碼可存放的資料量約 20 characters (字元)，資料量不足是其設計上的缺點。二維條碼在設計上增加了垂直方向這個維度的空間，最常見的是 1994 年由日本 Denso-Wave 公司發明的 QR Code (圖 1)，基本結構是由模組構成的正方形，黑色模組代表 1，白色模組代表 0。模組解碼後可以對應到網址、手機號碼、簡訊、字串等資料。第 1 代的 QR Code 設計為 21×21 模組，可存放 35 個字元。每增加一個版本，長寬各增加 4 個模組。目前

第 40 代 QR Code 為 177×177 模組，可存放高達 2953 個字元的 8 位元組訊息或 7089 字元的數字(Mishra & Mathuria, 2017)。

一維條碼 (Code 39)	二維條碼 (QR Code)	彩色二維條碼 (HCCB)
		

圖 1. 數位條碼

為了增加辨識度，QR Code 在左上、右上及左下角各設計了一個定位標記(圖 2)及數個較小的校正標記。QR Code 中並不是所有的模組都被用來儲存資料，而是保留部分模組用來容錯，因此當部分模組毀損時，QR Code 仍可以被正確解碼。

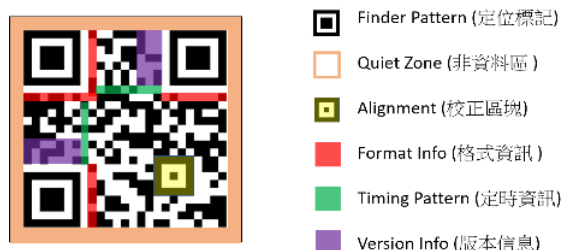


圖 2. QR Code 結構圖

為了在二維條碼有限的區域內儲存更多的資訊，最簡單的做法是增加模組的數量，但這會提高掃描及辨識的困難度。另一個方法是提高單一模組的變化量，例如使用顏色模組來取代黑白模組。使用彩色二維條碼雖可增加資料量，但在實務上解碼彩色二維條碼會比黑白二維條碼困難，程序也較複雜。因為擷取色彩過程中，物體顏色會受到環境光源的色溫及照度影響，產生顏色失真。甚至相鄰模組之間的顏色也可能會互相干擾，產生混色及造成誤判(Yang, Xu, Deng, Loy, & Lau, 2018)。

關於彩色二維條碼的資料相當少，文獻中報告過模組設計可選用三角形或正方格兩種顏色模組(John & Raahemifar, 2015)。網路上可以查到的產品只有微軟在 2007-2013 年開發，利用 CMYK 這 4 個顏色的小三角形製作的 5×10 模組的商品 Microsoft High Capacity Color Barcode (HCCB) (Jancke, 2007)及 High Capacity Colored Two Dimensional Codes (HCC2D) (Grillo, Lentini, Querini, & Italiano, 2010)(圖 1)，但目前 HCCB 相關資訊在微軟網站上已停止更新。

2、電腦彩色視覺

電腦的世界是由 0 和 1 組成，所以它需要將”看”到的影像先轉成 0 和 1 的數位訊號才能讀懂。電腦視覺的產生得流程需要先利用鏡頭感光元件來接收光線及感應顏色，利用影像擷取卡將類比視訊轉為數位訊號，再將訊號輸出至電腦主機裡進行分析處理。在電腦”眼”中，色彩是一組 RGB 數值，R 指紅色(Red)、G 指綠色(Green)、B 是藍色(Blue)。RGB 模式中，所有顏色都是由紅、綠、藍按照不同的比例疊加而成。RGB 的”值”指的是顏色的強度，從 0 到 255 共分 256 級。當 RGB 值為 0 或 255 時，此時色彩的飽和度(彩度)位於極端，這種情況共有 8 個(圖 3)。理論上，這 8 個極端色彩應該是比較容易被電腦分辨出的顏色，因此被選用來製作本實驗的 Colour Matrix。



圖 3. 具極端彩度的 8 個顏色及其相對的 RGB 數值。

3、彩色辨識

人的眼睛及大腦能藉由生活經驗及試誤學習，在不同光源及亮度下判斷正確的顏色。但是，電腦沒有這個能力。電腦感知的顏色是利用接收到的 0 和 1 數位訊號算出來的，而不是”看”出來的，當計算的結果不準確，電腦就會看到錯誤的顏色。因此，會發生在清晨拍的彩色數位照片偏紅而在黃昏拍的照片偏黃的情形。但由於目前相機主要被設計來拍攝人物及風景，賣點是高畫質影像及賞心悅目的色彩，顏色是否失真反而是比較不被重視的問題(Akkaynak et al., 2014)，因此，正確顏色辨識是開發彩色二維條碼的一大挑戰。

彩色二維碼開發過程需要經過編碼、彩色影像輸出、影像擷取、彩色濾光片 RGB 分色、分色資料提取、還原原始彩色影像，再經過解碼對應出資訊內容。為了使顏色提取正確，可以使用影像後處理技術如 luminance enhancement、thresholding binarization、salt and pepper refinement algorithm 等(Bhardwaj, Kumar, Verma, Jindal, & Bhondekar,

2016)。為了克服 RGB 值因環境照度與照相條件而改變，可使用機器學習的分類演算法辨識顏色，如隨機森林演算法(random forest)、支援向量機演算法(support vector machine)、差商演算法(quadratic discriminant analysis)等(Yang et al., 2016)，讓電腦預測及校正所讀取的顏色資訊。

4、影像處理與影像切割(Image segmentation)

在一張圖片中搜尋目標影像時，可以針對目標影像的特徵來進行適當的切割，以簡化待處理的資訊，加速後續圖像的分析。閾值分割法是一種利用設定的灰階閾值將整張影像分割的技術(圖 4)。在一張 8 bits 灰階圖中，最暗(灰階值為 0)到最亮(灰階值為 255)共有 256 種亮度變化。

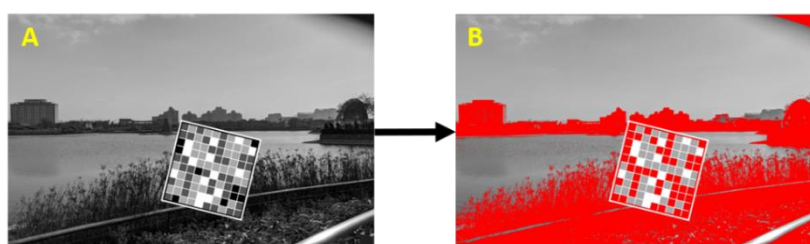


圖 4. (A)原始影像 (B)閾值分割法(灰階值<100 的區域以紅色標示)。

圖像中兩個區域在交界處會存在灰階值差異，因此可以利用這種灰階值的梯度落差檢測邊緣。Canny edge detection 是一種常見的邊緣檢測法，原理是先利用高斯濾波器來呈現灰度梯度值分布，計算出可用來區分相鄰兩區塊的灰度梯度極大值及極小值。當灰度梯度較極大值高時，認定為區塊邊緣(圖 5A 藍線 a 段)；灰度梯度介於極大值及極小值之間但是和灰度梯度大於極大值像素相連的(圖 5A 藍線 b 段)，認定為區塊邊緣；灰度梯度介於極大值及極小值之間但是未和灰度梯度大於極大值像素相連的(圖 5A 綠線)，則非區塊邊緣；灰度梯度較極小值低的，亦為非區塊邊緣(圖 5A 洋紅線)。圖 5C 利用邊緣檢測方法來分類圖片中的物體，Colour Matrix 的邊緣可偵測出來。

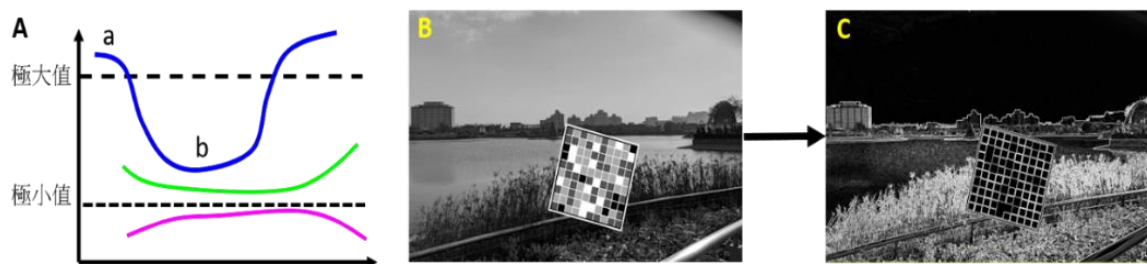


圖 5. Canny edge detection (A)邊緣連線原理 (B)原始影像 (C)邊緣檢測法。

5、影像處理-透視轉換(Perspective Transformation)

3D 實物轉換成 2D 平面影像的過程中，距離遠近的資訊會因壓縮而消失，物體形狀也會扭曲形變。當掃描 Colour Matrix 後，圖片中線與線的交叉點座標及線條的尺寸會失真。因此，當在現實世界中掃描 Colour Matrix 後，圖片必須轉換回正面鳥瞰圖，才能對應到模組的座標，進行模組顏色解碼的動作。本實驗使用四點轉換(4-point perspective transform)來進行轉換(圖 6)。

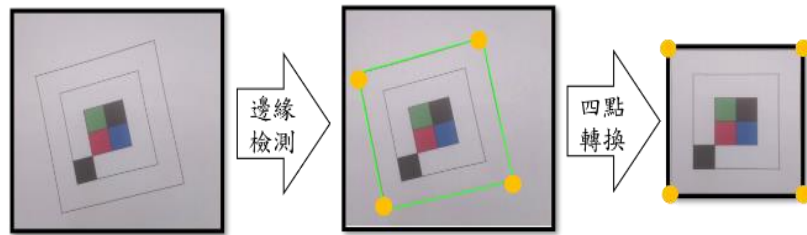


圖 6. (A)拍攝圖影 (B)邊緣檢測 (C)四點轉換。

6、機器學習

機器學習是一種利用人工智慧技術，讓電腦經由處理及學習大量資料，訓練分辨資料特徵及產生解決問題的能力。所以當新的數據出現時，便能根據過往訓練的經驗(模型)進行預測。進行機器學習時，會預留一部分資料作為測試集，用來評估演算法的表現。其他資料則用於訓練演算法，稱為訓練集。模型的預測能力是評估模型好壞的依據，最常見的指標是準確率(Accuracy)，準確率越高的模型其預測能力通常也越好。由於個人經驗有限，進行顏色辨識時，先考慮較常見的機器學習分類演算法，包括決策樹、k-近鄰演算法(k-nearest neighbor, kNN) 和多層感知器(multilayer perceptron, MLP)。

決策樹是一種利用像樹一樣具有分支及葉片的結構圖來做決策的分類方法，樹的每個分支節點代表判斷條件，分類結果則連結到下一層的決策節點或是分類終點(葉片)(Quinlan, 1996)。決策樹可以透過分類建立關聯性規則，歸納出資料的共同屬性，用於預測。

kNN 是一種常用的數據探勘法。當處理的是分類問題時，分類結果將由 k 個鄰近樣本以多數決的方式決定；當處理的是樣本是數值時，預測值則由鄰近 k 個樣本平均值來決定(Altman, 1992)。以圖 7 為例，當要判斷一個未知物體的顏色時，可以先將訓練用的顏色數值先繪於座標系統中，再比對未知顏色的座標位置和哪個顏色的座標點最接近，便預測未知物體為那一個顏色。進行 kNN 演算法時，可以選擇的參數除了 k 值，

還可以對 k 個鄰近樣本的貢獻進行加權，賦予較近距離的樣本較大的距離加權 (weighted)。這種改良版的 kNN 稱為加權 kNN。

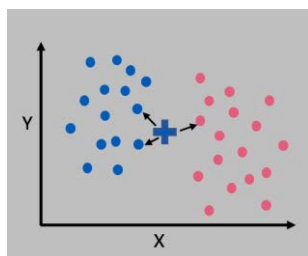


圖 7. k-近鄰演算法(kNN)

MLP 是一種類神經網路，包含一個輸入層、數個隱藏層及一個輸出層(圖 8)。文獻中這種方法曾被成功用來判斷汽車的顏色(Zhang et al., 2018)。MLP 的設計是模仿人類神經系統的傳遞，網路中的節點扮演著神經元的功能(圖 8 紅色圓圈)，節點和上層及下層間有對應的連結。節點在處理輸入及輸出訊號的過程中，會學習到與上下層之間的連結強度，並修正每條迴路之權重值，進行最佳化推演。

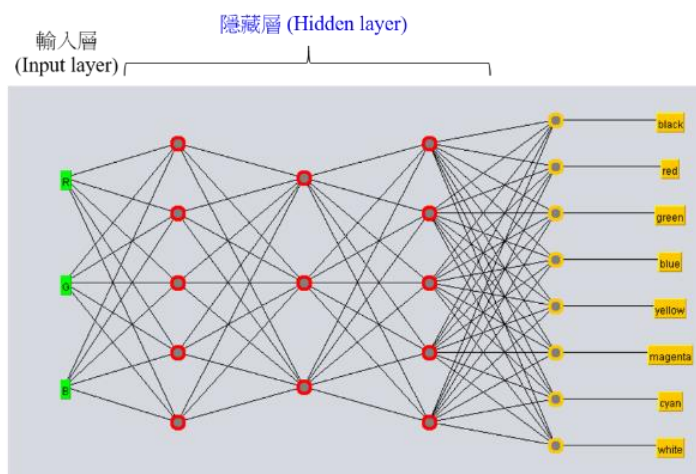


圖 8. 多層感知器(MLP)

(三)、 研究目的

- 1、 設計一款彩色二維條碼”Colour Matrix”及其相應編碼與解碼，能儲存更多資料。
- 2、 發展在手持裝置處理 Colour Matrix 的顏色辨識及資料讀取的方法。
- 3、 所開發的 Colour Matrix 套件和現行 QR Code 的效能評比。

二、 研究方法或過程

(一)、 軟硬體選擇

開發彩色二維條碼平台需選用支援影像擷取的硬體及資料後處理的軟體。評估的硬體包括筆記型電腦、Arduino 及 Raspberry Pi (簡稱 RPi) (表格 1)。受限於經費及手持裝置，開發板式如 Arduino 及 RPi 是較適合的平台。最後選用 RPi 作為開發的平台，因為 RPi 有專屬 Raspberry Pi Camera (Pi Cam)相機，鏡頭為 800 萬像素的 Sony IMX219 8-megapixel sensor，解析度和市售低階手機差不多，加上 RPi 也搭配 Pi Cam 軟體，使用方便。RPi 另一個優勢是支援 Linux 作業系統及 Python。由於 Python 提供許多影像處理及機器學習套件，加上網路上有非常多學習管道，可望降低開發門檻。

表格 1. 硬體規格比較表

硬體	優點	缺點
筆記型電腦	64 位元微處理器 記憶體 4GB 起 作業系統(+) 內建顯示晶片	體積大，不適合手持 價格高，15000 起台幣
Arduino Uno	體積小 價格低，762 台幣	8 位元微處理器 記憶體 0.002MB 未內建顯示晶片，無法處理影像 作業系統(-)
Raspberry Pi 4	體積小 64 位元微處理器 記憶體 4GB 作業系統(+) Linux 內建顯示晶片	價格中等，2400 台幣

1、 硬體

Raspberry Pi 4 Model B	Micro SD card (128GB)
Raspberry Pi Camera Module v2	Micro HDMI to HDMI adapter
Logitech C922 Pro Stream WebCam	3A type-C cable
筆記型電腦(乙太網路)	乙太網路線
電腦螢幕	PASCO 無線光感應器
USB 鍵盤/滑鼠	Double A 影印紙 80 磅
HP LaserJet Pro CP1025 彩色印表機	列印解析度 600 x 600 dpi

2、軟體與工具

Python 3.7.3

opencv-Python 3.4.6.2

pyzbar 0.1.8

Webcamoid

scikit-learn 0.23.1 (SKlearn)

Raspberry Pi OS GNU/Linux 10 (buster)

相關套件： numpy、scipy、pillow、pandas、matplotlib。

(二)、 Colour Matrix

1、 設計

進行手持彩色二維條碼開發的過程中，我參考了網路及參考文獻自己設計了一款彩色二維條碼，命名為"Colour Matrix"。彩色二維條碼必須具備類似 QR Code 的定位標記，程式才能將它定位、擷取及判讀。我設計的彩色二維碼 Colour Matrix (如圖 9) 選用 10×10 的模組，模組有 8 種顏色，為具高度飽和彩度的色彩。為了增加辨識率，Colour Matrix 的四個頂角設計為定位格，功能類似 QR Code 中的定位標記。左上、右上、左下、右下的定位格顏色分別是紅、黑、綠、藍。符合時程式將自動判定掃描到的圖像為 Colour Matrix。定位格的顏色同時也用來作為 Colour Matrix 轉正時的方向校正基準，協助程式呈現 Colour Matrix 鳥瞰圖。當 Colour Matrix 正確轉正後，模組讀取的順序為由左而右、由上而下。

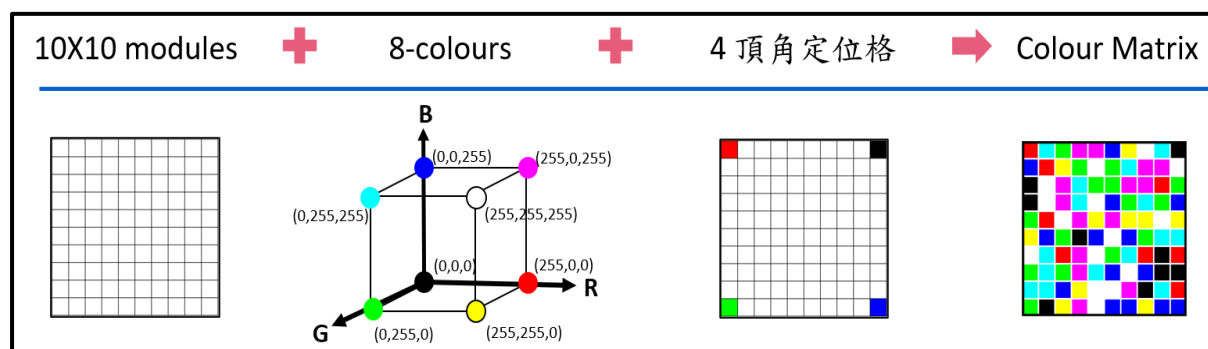


圖 9. Colour Matrix 設計

2、 編碼

Colour Matrix 顏色模組所對應的代碼及編碼原則如圖 10。由於設計的 Colour Matrix 有 8 個顏色，使用不用到三個模組就可以完整的儲存一個子元。相較之下，QR Code 只有黑與白色，所以須要 8 個模組才能儲存一個字元。

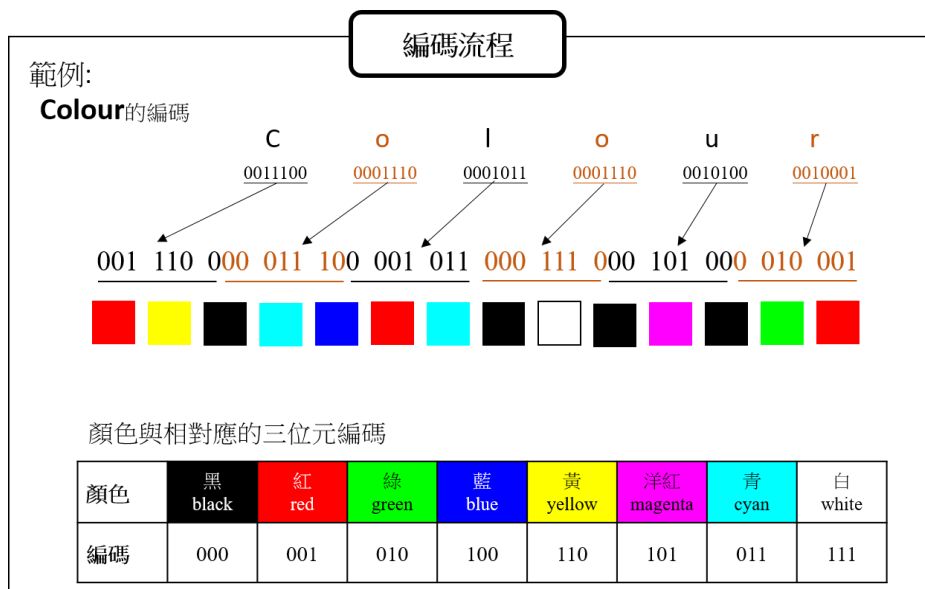


圖 10. 編碼流程圖

QR Code 的編碼方式為 8 位元編碼，有數字，英數，位元組，日語漢字模式及其他模式五種。在位元組模式中，編碼可以是 ISO-8859-1 字元或是 UTF-8 的編碼。ISO 8859-1 是國際標準化組織內 ISO/IEC 8859 的第一個 8 位字符集，以 ASCII 為基礎，可用來表示使用附加符號的拉丁字母，而 ASCII 同樣也是 UTF-8 的子集。因此我的構想是使用類似的編碼方式，資料讀出來後可以使用一串類似 ASCII to Binary Code 的編碼來解讀 Colour Matrix。URL 中的字元共 86 個，由大小寫英文字母、數字 0-9、特殊符號、終止碼組成，所以至少需 7 位元編碼(即 2 的 7 次方=128)，才能完整的涵蓋，所以本實驗 Colour Matrix 的字元編碼採 7 位元制(表格 2)。每個 10×10 模組的 Colour Matrix 在扣除四個定位點後會剩下 96 個模組，每個模組能儲存 8 個顏色(2 的 3 次方=8)，即 3 位元，因此共可儲存 288 個位元。字串最後須加上 END 字元(編碼為 1010110)，END 後不足 288 位元的部分填入亂數。

表格 2. 字元編碼本

字元	代碼	編碼	字元	代碼	編碼	字元	代碼	編碼	字元	代碼	編碼	字元	代碼	編碼
a	0	0000000	u	20	0010100	O	40	0101000	8	60	0111100	-	80	1010001
b	1	0000001	v	21	0010101	P	41	0101001	9	61	0111101	_	81	1010010
c	2	0000010	w	22	0010110	Q	42	0101010	!	62	0111110	.	82	1010011
d	3	0000011	x	23	0010111	R	43	0101011	*	63	1000001	~	83	1010100
e	4	0000100	y	24	0011000	S	44	0101100	"	64	1000000	SPACE	84	1010101
f	5	0000101	z	25	0011001	T	45	0101101	(65	1000010	END	85	1010110

g	6	0000110	A	26	0011010	U	46	0101110)	66	1000011			
h	7	0000111	B	27	0011011	V	47	0101111	;	67	1000100			
i	8	0001000	C	28	0011100	W	48	0110000	:	68	1000101			
j	9	0001001	D	29	0011101	X	49	0110001	@	69	1000110			
k	10	0001010	E	30	0011110	Y	50	0110010	&	70	1000111			
l	11	0001011	F	31	0011111	Z	51	0110011	=	71	1001000			
m	12	0001100	G	32	0100000	0	52	0110100	+	72	1001001			
n	13	0001101	H	33	0100001	1	53	0110101	\$	73	1001010			
o	14	0001110	I	34	0100010	2	54	0110110	/	74	1001011			
p	15	0001111	J	35	0100011	3	55	0110111	?	75	1001100			
q	16	0010000	K	36	0100100	4	56	0111000	%	76	10011.01			
r	17	0010001	L	37	0100101	5	57	0111001	#	77	1001110			
s	18	0010010	M	38	0100110	6	58	0111010	[78	1001111			
t	19	0010011	M	39	0100111	7	59	0111011]	79	1010000			

3、尋找、定位與確認

尋找 Colour Matrix 時，可以利用正方形形變後呈現四邊形的特徵，使用 Canny edge detection 來辨識圖片中物件的邊緣線條，接著篩選出這些線條中可以組成一個四邊形的集合。程式偵測到一個四邊形後，會經過灰階及邊緣辨識，尋找符合 Colour Matrix 的定義的四邊形，確認後再進行解碼(圖 11)。4 個頂角定位點的確認步驟是先將四個定位點應該出現的地方擷取下來，由機器學習進行顏色辨識流程，確認四個定位點的顏色順序是否跟 Colour Matrix 一樣，如果顏色順序對但方向不符合，就進行旋轉，直到旋轉至正確的角度為止，這樣就得到一個被確認過且轉正的 Colour Matrix。

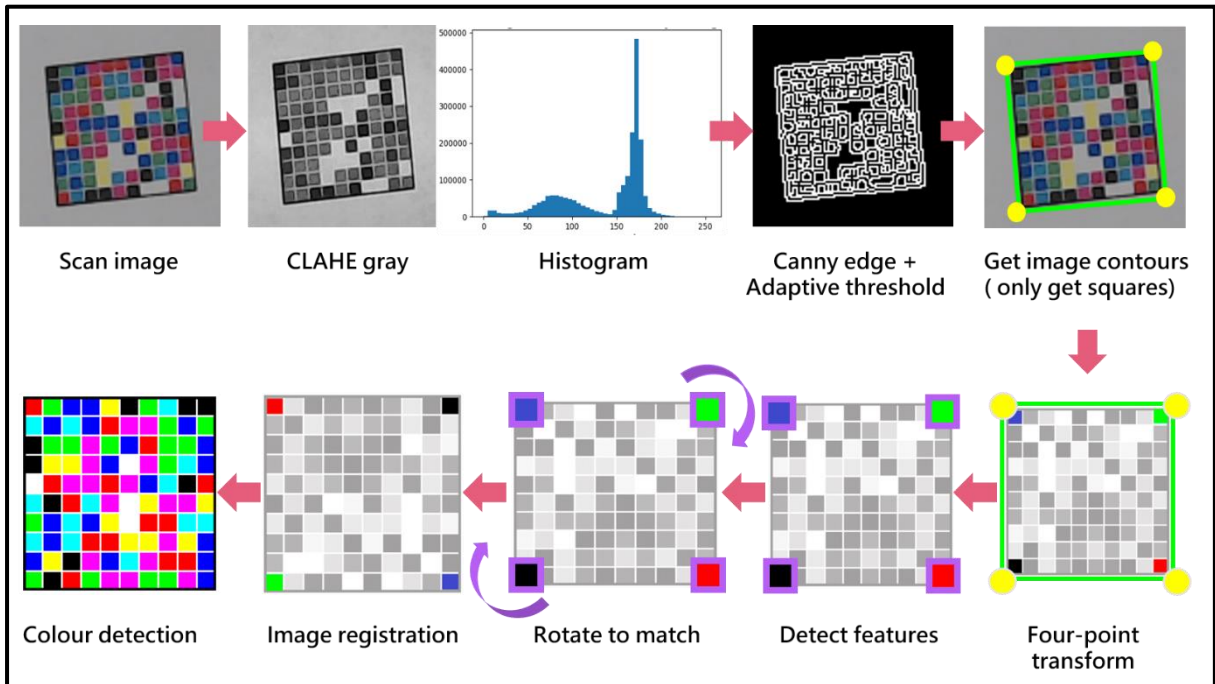

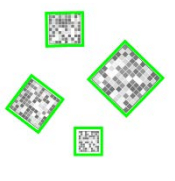

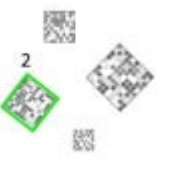
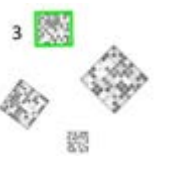
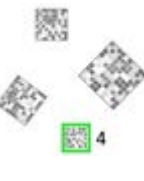


圖 11. Colour Matrix 定位及辨識流程

當圖片中找到的四邊形超過一個時，程式會依據四邊形面積大小排序，依此順序進入迴圈，進行後續處理(表格 3)。

表格 3. 多張 Colour Matrix 時的辨識順序

原始影像	多張 Colour Matrix 辨識順序 (找到四邊形→依面積大小排序→進行後續處理)				
					

4、解碼

Colour Matrix 為 8 個顏色組成的 10×10 模組，包含著 4 個定位點跟 96 個每個顏色模組，每個顏色模組都可以定應到一個三位元的代碼，利用對照表可以解出 Colour Matrix 的資訊，解碼範例如圖 12。

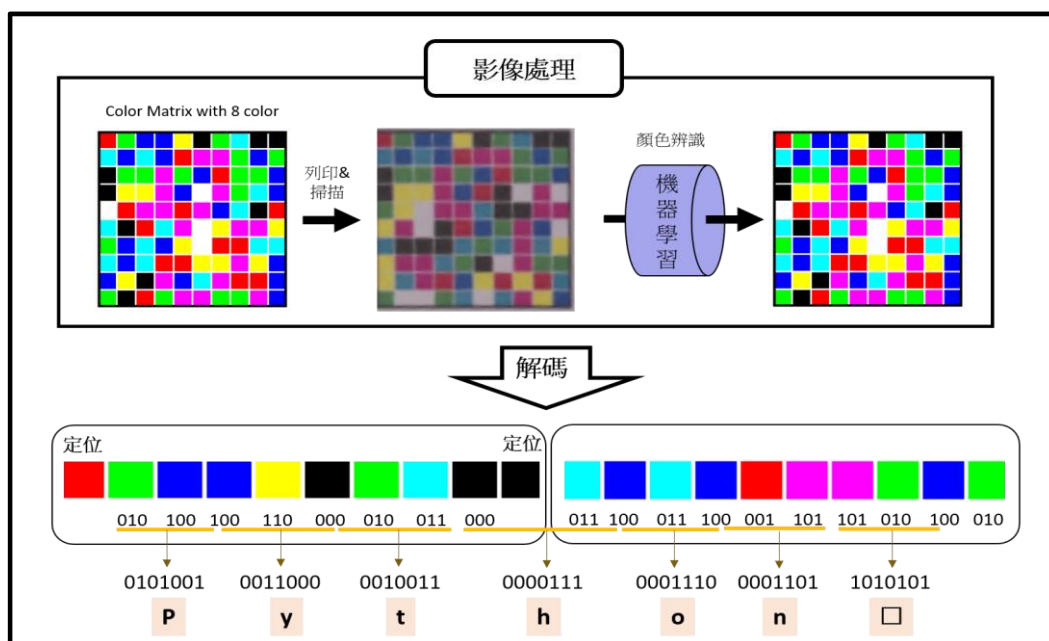


圖 12. Colour Matrix 解碼的流程圖

(四)、 實驗 1: 建立顏色辨識訓練模型

電腦判斷顏色時是由一組 RGB 值決定，但實務上鏡頭在擷取顏色過程中會因環境光源及照度的影響而失真，導致相同顏色在不同環境照度下具有相異的 RGB 座標。本實驗採用機器學習的分類法來訓練及建立顏色辨識模型。訓練集的資料在盡可能多樣性的環境下進行，並透過模型再訓練學習分類錯誤的資料(miss-classified observations)，增加訓練集資料多樣化，提高預測的正確性。

[實驗步驟]:

1. 製作邊長 2 公分×2 公分，具 8 種顏色，10×10 模組的 Colour Matrix。
2. 在室內、LED 燈、色溫 4000K、亮度 100-700 Lux 下拍攝影像(圖 14)。
3. 使用 opencv 辨識物件並輸出 RGB 值。
4. 進行監督式學習，手動標示每個 RGB 值所對應的正確顏色。
5. 使用 SKlearn 建立顏色辨識的演算法，包括決策樹、kNN 及 MLP。
6. 重複上述流程，將被辨識錯誤的模組資料正確標示後，加入訓練集，直到能得到一個有效辨識顏色的訓練集。

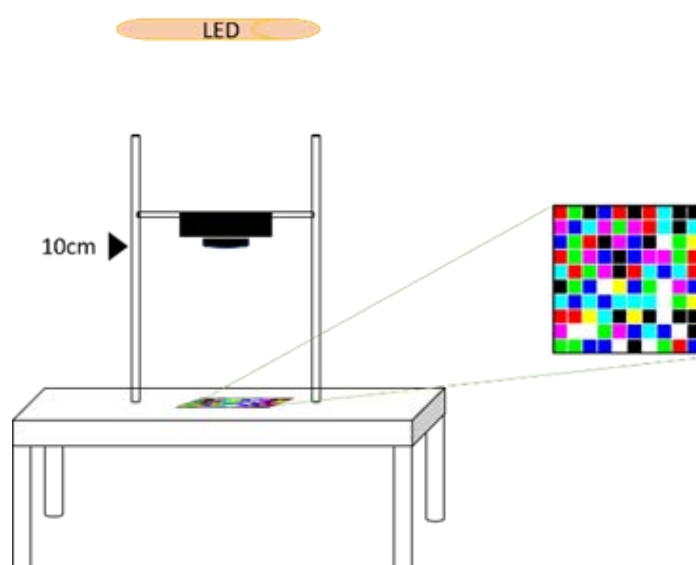


圖 14. 實驗設置圖

(五)、 實驗 2: 單張 Colour Matrix 和 QR Code 的效能比較

以 Model 2 v1 QR Code 為對照組，比較 Colour Matrix 及 QR Code 在不同拍攝距離、傾角及亮度下的解碼效能。使用的 Colour Matrix 及 QR Code 邊長同為 2 公分，儲存內容為 - Python is fun 或 I Like Python。

[實驗步驟]:

1. 以 Pi Cam 拍攝桌面上 Colour Matrix，光源為色溫 4000K 的 LED 燈。
2. 拍攝條件如下，每個條件拍攝 3 次(圖 15)，進行影像解碼流程。
 - (1) Pi Cam 高度依序調整為距離桌面 5、7.5、10、12.5、15、17.5、20、22.5、25、27.5、30、32.5、35 公分。
 - (2) Pi Cam 固定在 Colour Matrix 正上方 10cm，傾斜桌面使拍攝傾角分別為 0° 、 15° 、 30° 、 45° 、 60° 、 75° 。
 - (3) Pi Cam 固定在 Colour Matrix 正上方 10cm，調整桌面照度為 100、200、300、400、500、600 及 700 Lux
3. 於 RPi 上安裝 pyzbar 0.1.8 來讀取 QR Code，pyzbar 是一款常用讀取的 QR Code 的程式，支援多組 QR Code 解碼。
4. 以 QR Code 替換 Colour Matrix 並按重複以上步驟，拍攝後以 pyzbar 進行解碼。

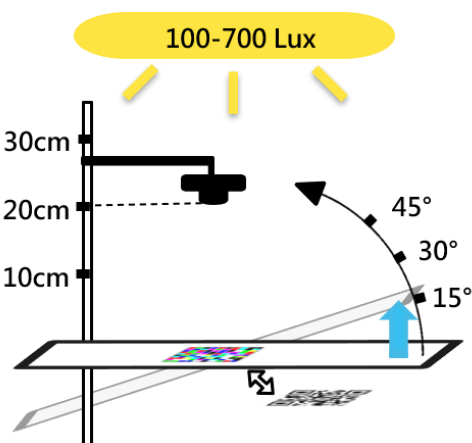


圖 15. 實驗設置圖

(六)、 實驗 3: 多張 Colour Matrix 和 QR Code 的效能比較

市面上的 QR Code 掃描及解碼器只支援單張 QR Code，如能同時掃描及解碼多張條碼，將有機會提高條碼的應用範圍。以下實驗目的是為了測試所開發的 Colour Matrix 套件在進行多張 Colour Matrix 辨識的效能。

[實驗步驟]:

1. 列印 Colour Matrix 於 A4 紙張，每列 10 個，共 7 列。

2. Pi Cam 固定在 Colour Matrix 正上方 20cm 處，進行 10 次拍攝(圖 16)。
3. 進行影像解碼流程。
4. 分別輸出邊框辨識及解碼結果。
5. 列印 QR Code 於 A4 紙張，每列 10 個，共 7 列。重複以上步驟。

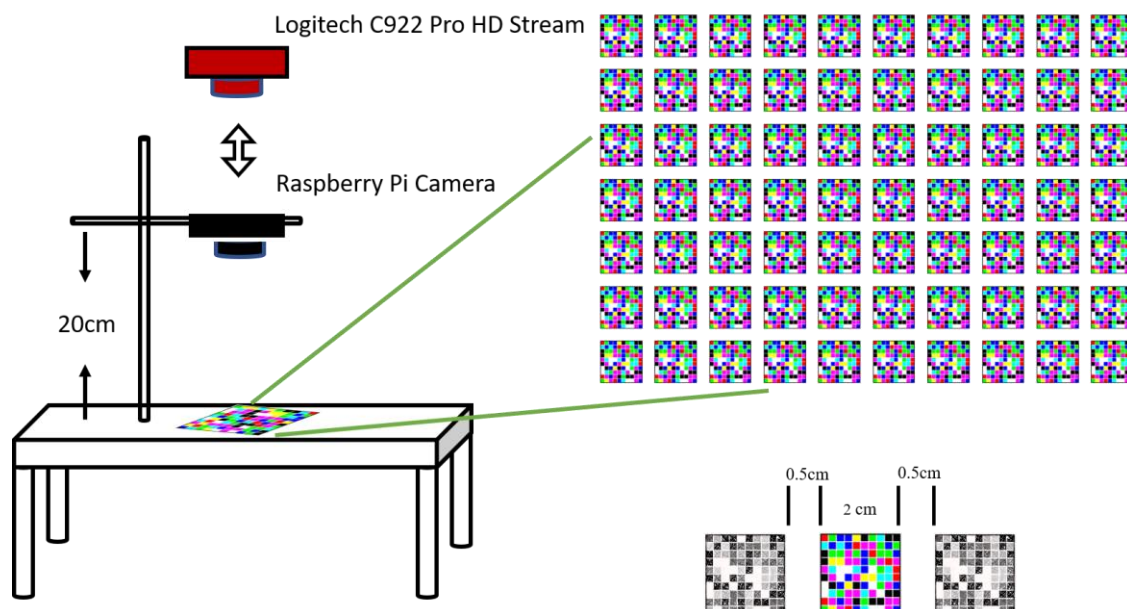


圖 16. 多張條碼拍攝

(七)、 實驗 4: 選用不同鏡頭對掃描效能的影響

初期採用 Pi Cam 鏡頭進行測試的原因是因 RPi 內建 Pi Cam 驅動程式，可隨插即用，方便操作。但 PiCam 是一個相對規格陽春的鏡頭，雖然配備有很好的 Sony CCD，但是因採用塑膠鏡頭，拍攝品質不及玻璃鏡頭，照片相對上景深較淺、影像較不清晰。Pi Cam 另一項缺點是只有手動對焦功能，拍攝較為耗時。

進行了前面數個實驗後，由實驗結果推斷如果能改良"相機"這個因素，或許有機會提升辨識率。評估了台灣境內可取得且支援 RPi 的所有 webcam 後("RPi USB Webcams,")，發現 Logitech C922 Pro Stream WebCam 具最高解析度，配有玻璃鏡頭，且提供 Full HD 1080P 畫質，又能自動對焦，為最佳選擇。但是 Logitech C922 不支援 Pi Cam 的拍照軟體，只能用 OpenCV 來進行拍照。但是最後發現 OpenCV 無法進行自動對焦，因此最後是使用軟體 Webcamoid 來進行手動對焦拍照。

[實驗步驟]:

1. 將實驗 3 之相機更換為 Logitech C922。其他步驟同實驗 3。

三、 研究結果與討論

(一)、 實驗 1: 建立顏色辨識訓練模型

1、 建立訓練集

在多樣化拍攝條件下，選取資料來建立訓練集，共 713 筆資料，分布如圖 17。

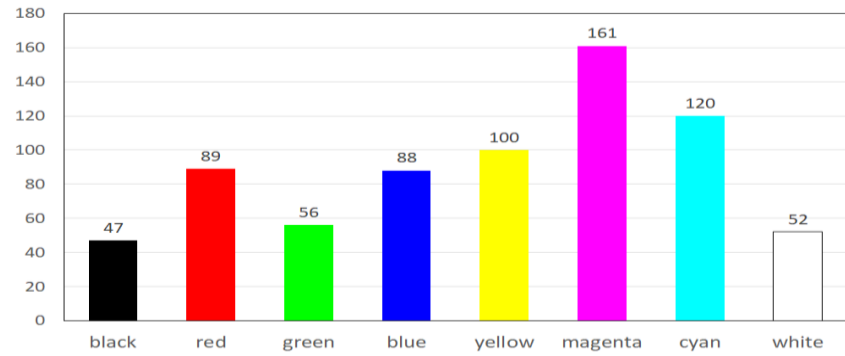


圖 17. 各顏色的訓練資料數量

2、 評估各機器學習演算法的效能

使用各個機器學習演算法進行顏色辨識的最佳效能如表格 4。由於 kNN 較 MLP 簡單而效能好，而訓練的模型正確率相同，所以 kNN 被選用來進行接下來的顏色辨識實驗。此部分的顏色辨識機器學習先在筆電上進行測試，接下來實驗完全在手持裝置 RPi 上執行，事後將所有實驗數據存成 csv 檔，輸入 RPi 的 SKlearn，用來訓練後續實驗顏色辨識的機器學習分類法，並將 kNN 之最佳模型以 pickle library 儲存。

表格 4. 各演算法進行顏色辨識的正確率

演算法	決策樹	MLP (Layer = 1)	kNN (k = 3)
正確率 (%)	98.0	99.0	97.4 to 99.0

3、 進行 kNN 最佳化

選擇不同 k 值及權重時，顏色辨識正確率結果如表格 5。因此選用 k = 3 及權重設定為 weighted 來進行接下來的實驗。

表格 5. 不同參數設定時 kNN 對顏色辨識的正確率

k	1	3	5	7	9
Distance	97.4	99.0	99.0	99.0	99.0
Uniform	97.4	98.4	99.0	99.0	99.0

4、 建立高品質訓練集

表格 6 為不同環境照度下 8 個顏色模組的 RGB 數值。可發現相同顏色在不同照度下具有明顯不同的 RGB 數值，可能導致顏色預測失準。例如紅色理論上 R 值為 255，但實際測得的 R 值介於 132 和 161 之間；而紅色的 G 值和 B 值原本應該皆為 0，但在 200Lux 下，G 值高達 48 而 B 值高達 62。

此外，同樣照度下各顏色的 RGB 值範圍有時也不符預期。例如當照度為 500 Lux 時，紅色為(132,32,45)，其中 R 值為 132，明顯高於 G 值和 B 值，這樣的情況符合我的期待。但此條件下的綠色為(81,109,54)，其中 G 值只有 109，與 R 值 81 非常接近，和理論上的綠色 RGB 值(0,255,0)有所出入。藍色的 RGB 理論值為(0,0,255)，實際測量為(38,39,92)，其中最高的為 B 值只有 92，比預期低。此外，黑色在 200 Lux 下會近似深灰色，而白色則接近極淺的灰色。由此實驗數據發現高亮度的環境會造成畫面偏紅，對綠色的辨識不利，且普遍藍色 RGB 的數值均偏低。因此，增加訓練集資料多樣化對進行正確顏色的預測有其必要性。

表格 6. 不同照度下平均 RGB 值

亮度 顏色	200 Lux	300 Lux	400 Lux	500 Lux
紅	(161,48,62)	(136,39,53)	(135,37,51)	(132,32,45)
綠	(109,137,72)	(93,120,61)	(91,116,62)	(81,109,54)
藍	(58,56,115)	(44,44,100)	(45,45,97)	(38,39,92)
黑	(56,47,47)	(45,39,41)	(43,37,39)	(34,30,32)
青	(47,92,117)	(43,65,113)	(59,103,128)	(68,107,130)
洋紅	(118,31,79)	(134,39,89)	(135,40,93)	(145,53,101)
黃	(159,149,73)	(175,161,74)	(176,165,79)	(177,165,83)
白	(188,180,182)	(177,166,172)	(166,159,164)	(165,158,164)

(二)、 實驗 2: 單張 Colour Matrix 和 QR Code 的效能比較

1、 掃描距離

當鏡頭和 Colour Matrix 的距離在 30 公分之內時，Colour Matrix 可被正確偵測到，並成功解碼，顯示程式運作正常，但是當距離超過 30 公分時，邊緣檢測無法成功執行(表格 7)。原因可能是影像邊緣太過模糊，以致於程式偵測的到的邊緣並非實際上 Colour

Matrix 的邊框。當掃描距離在 QR Code 的 5 公分內時，QR Code 無法被偵測出，距離 5-30 公分時 QR Code 則可以正確辨識及解碼(表格 7)。

表格 7. 不同掃描距離下條碼辨識及解碼

距離 (cm)		5	7.5	10	12.5	15	17.5	20	22.5	25	27.5	30
Colour Matrix	邊框辨識	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	解碼	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
QR Code	解碼	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

2、拍攝傾角評比

以 0°至 60°傾角拍攝 Colour Matrix 時，可以成功邊框辨識及解碼；當拍攝傾角為 75°時，Colour Matrix 邊框辨識及解碼失敗(表格 8)。當使用 0°至 45°傾角拍攝 QR Code 時，解碼成功；但拍攝傾角 $\geq 60^\circ$ 時，解碼失敗(表格 8)。推測造成 Colour Matrix 邊框辨識及解碼失敗的原因可能是拍攝鏡頭的極限或程式無法克服 Colour Matrix 的影像形變。

表格 8. 不同桌面傾角下條碼辨識及解碼

角度		0°	15°	30°	45°	60°	75°
Colour Matrix	邊框辨識	✓	✓	✓	✓	✓	✗
	解碼	✓	✓	✓	✓	✓	✗
QR Code	解碼	✓	✓	✓	✓	✗	✗

3、環境照度

在室內光源為 4000K 及照度 100-700Lux 下，Colour Matrix 及 QR Code 皆能被掃描成功且正確解碼(表格 9)。顯示本實驗所建立的顏色辨識模型能成功克服日常環境照度對顏色 RGB 值的干擾，可成功在日常室內環境下使用。


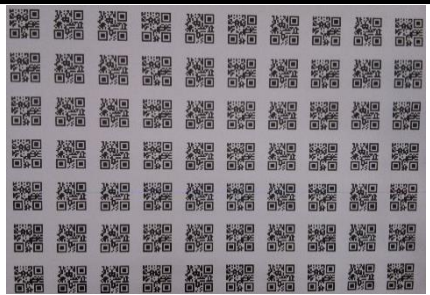
表格 9. 不同環境照度下條碼辨識及解碼

照度(Lux)		100	200	300	400	500	600	700
Colour Matrix	邊框辨識	✓	✓	✓	✓	✓	✓	✓
	解碼	✓	✓	✓	✓	✓	✓	✓
QR Code	解碼	✓	✓	✓	✓	✓	✓	✓

(三)、 實驗 3: 多張 Colour Matrix 和 QR Code 的效能比較

以 Pi Cam 拍攝多張 Colour Matrix 的實驗中，邊框辨識率為 70.0%-82.5%，平均 $74.3 \pm 5.0\%$ 。解碼成功率為 65.7%-80.7%，平均 $70.2 \pm 6.1\%$ (表格 10)。邊框辨識成功的 Colour Matrix 絕大多數在後續的解碼步驟也都是成功的。顯示邊框辨識應是關鍵步驟。拍攝多張 QR Code 時，解碼均未能成功 (表格 10)。


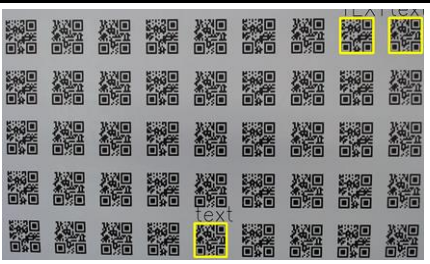
表格 10. 使用 Pi Cam 拍攝多張條碼的效能

	Colour Matrix	QR Code
邊框辨識率	$74.3 \pm 5.0\%$	
解碼成功率	$70.2 \pm 6.1\%$	0%
範例		

(四)、 實驗 4: 選用不同鏡頭對掃描效能的影響

鏡頭更換為 Logitech C922 時，多張 Colour Matrix 邊框辨識率由 74.3% 提升至 99.1%，解碼成功率由 70.2% 提升至 92.4% (表格 11)。使用 Logitech C922 拍攝多張多張 QR Code 時，成功率為 0-6.7% ($3.1 \pm 1.9\%$)，較使用 Pi Cam 拍攝時微幅改善，但遠低於 Colour Matrix 的成功率 (表格 11)。

表格 11. 使用 Logitech C922 拍攝多張條碼的效能

	Colour Matrix	QR Code
邊框辨識率	$99.1 \pm 1.1\%$	
解碼成功率	$92.4 \pm 3.7\%$	$3.1 \pm 1.9\%$
範例		

註: Pi Cam 視野長寬比為 4:3，Logitech C922 為 16:9。

四、 討論與應用

(一)、 高品質訓練集的建立決定顏色辨識的成功率

在檢視實驗數據時，發現顏色的 RGB 值在不同拍攝環境下，數值分布的範圍很大，有時和理論值相去甚遠，有時甚至會超乎預期。為了讓建立的機器學習模型能在各式各樣的情況下進行正確的顏色辨識，訓練時便需提供足夠、有代表性且多樣化的資料，這樣訓練出來的模型預測能力才會好。過程中也發現，過多的訓練資料也可能會造成過度擬和(overfitting)的現象，因此，建立一個適量而高品質的訓練集相當重要。

本實驗訓練資料是不同的環境光源及照度下所收集的 Colour Matrix 影像 RGB 值，再手動標記 RGB 值所對應的顏色，建立第一版的訓練集資料。接著，拍攝數張 Colour Matrix，挑選預測錯誤的資料，進行正確標記後加入更新版的訓練集。重複步驟多次後，觀察到顏色辨識正確率提升至一個穩定程度後，停止收集訓練集的新資料(圖 18)。結果顯示，依此步驟所建立的訓練集能有效在不同拍設環境下進行正確顏色辨識。

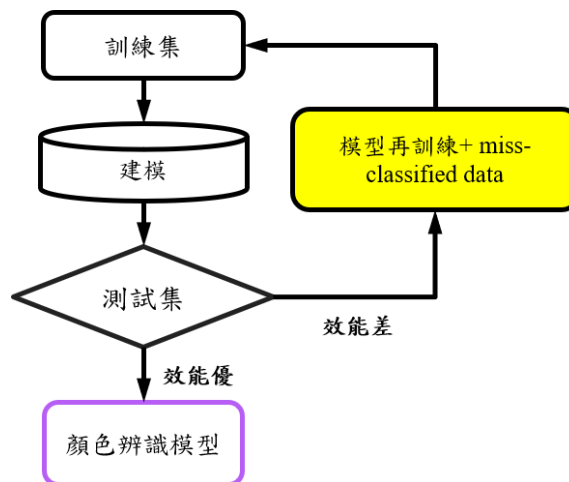


圖 18. 模型訓練流程圖

(二)、 訓練模型的參數最佳化

SKlearn 的 kNN 分類器，可以調整的參數除了 k 值外，還有權重的參數。權重的參數可以設定為 uniform、weighted 或是由使用者自己定義。預設是 uniform，意思是均等的權重，所有的鄰近點的權重都是相等的。weighted 是不均等的權重，距離近的点比距離遠的点的影響大。這個實驗的結果是 k=3 時，正確率最高。而權重方面則是 weighted 略優於 uniform。因為若 k 值較小，對鄰近的点會非常敏感，因此本研究中 k 取 3，權重採用 weighted，其他參數設定則使用預設值。

(三)、 解碼失敗的原因探討

1、 未偵測出 Colour Matrix

理論上，即使 Colour Matrix 因拍攝傾角而造成形變，形狀仍會近似多邊形。利用這個原理，邊框辨識流程先找出圖像中邊緣的集合，再利用 cv2.approxPolyDP 的方法找出邊緣所組成的圖像是否近似四邊形(如圖 19)，如果近似四邊形，再提取是否具有 Colour Matrix 的特殊設計。如果拍攝的 Colour Matrix 邊緣過於模糊、或印刷 Colour Matrix 紙張不平整、有摺痕時，Colour Matrix 拍攝後可能就不會近似四邊形，而是被判為多邊形，此時程式就會認定圖形非 Colour Matrix，便不會進行後續解碼流程。

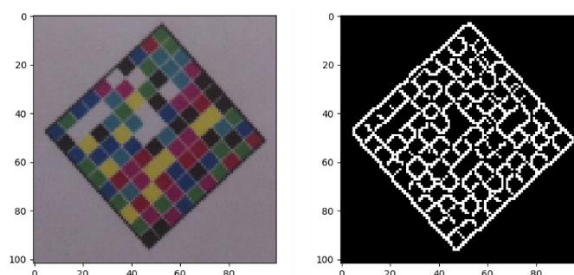


圖 19. Colour Matrix 頂角在邊緣辨識時呈現弧形，導致無法被判斷為四邊形

2、 定位點顏色辨識失敗

已知相同顏色在不同環境亮度下具有不同的 RGB 數值，因此，在一個較暗的環境中，理論上黑色 RGB 座標可能為 (50, 50, 50)，但也有機會測得的 RGB 座標為 (50, 50, 76)，對應到深藍，而被演算法辨識為藍色。為了避免發生在 RGB 三維空間中的同一個座標點對應到不同的顏色，可以增加用來訓練演算法的資料量，以及在更多元環境亮度下取得訓練模型用的資料。此外，選取最佳的演算法參數也會影響正確率。

3、 解碼失敗

即使成功找到 Colour Matrix 的四邊形區域，也對四邊形進行了正確的裁切，但由於圖像形變過大，仍可能導致辨識失敗(圖 20)。這通常發生在相機拍攝的角度過大時，Colour Matrix 的型變無法被程式修正，導致程式在抓取四邊形的四個頂角座標時，不符合實際上 Colour Matrix 的四個定位點位置，造成影像透視轉換失敗。或是當邊緣不清楚的時候，影像解碼時未辨識出 Colour Matrix。此外，當模組間色彩模糊時，也可能發生辨識失敗。



圖 20. Colour Matrix 邊緣可以被辨識(左)，但頂角未被正確提取(右)。

4、邊緣偵測的限制

實驗 4 結果發現高階鏡頭(Logitech C922)能增進 Colour Matrix 流程中邊緣檢測能力，進而提升解碼成功率。在掃描距離方面，初步的測試結果顯示更換高階鏡頭只讓可操作距離增加 5 公分(見表格 12)，在實際使用中幫助有限。由這兩個結果推測，更換鏡頭對改善 Colour Matrix 邊框辨識的效果可能侷限於掃描距離在 32.5 公分以內。如果要進一步提高邊框辨識的正確性，可能得嘗試使用人工智慧，例如 Convolutional Neural Network (CNN)這類方法來取代傳統的 Canny edge detection。

表格 12. 不同鏡頭對掃描距離的影響

距離 (cm)		2.5	5	7.5	10	12.5	15	17.5	20	22.5	25	27.5	30	32.5	35
Pi Cam	邊框辨識		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	解碼		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Logitech C922	邊框辨識		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	解碼		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

五、 結論與應用

(一)、 結論

- 1、 本研究設計一款 8 色 10×10 模組的 Colour Matrix，在 RPi 平台上利用 Python 以及數十個資源包撰寫 800 多行程式，標記了成千上萬筆資料，並選用了其中數百個合適且有效的資料來訓練機器學習演算法，成功開發出一套具視覺化使用者界面的完整彩色二維條碼產品。
- 2、 同樣尺寸，本實驗設計的 Colour Matrix 儲存容量可提升至 QR Code 的 3.4 倍。
- 3、 研究結果發現 kNN 為最佳的顏色辨識演算法，利用 kNN 開發之程式可正確進行顏色辨識及解碼，且不受鏡頭周邊形變影響。
- 4、 實驗挑選照度 100-700Lux 的資料來增加訓練集的多樣性。此外，還挑選被機器學習分類錯誤的資料(miss-classified observations)，重新進行標示後加入訓練集，進行模型再訓練(model retraining)。透過優化訓練集，成功提高顏色辨識正確性。
- 5、 本研究設計的邊框辨識流程用來尋找 Colour Matrix 的有效辨識距離可達 32.5 公分，支援拍攝傾角 0°至 60°，在單張條碼讀取的效能上，與 pyzbar 解碼 QR Code 的能力相當。
- 6、 使用高階鏡頭掃描多張 Colour Matrix 的辨識率高達 92.4%，而相同條件下，使用 pyzbar 讀取多張 QR Code 辨識率則只有 3.1%。在多張條碼掃描的效能上，Colour Matrix 方法明顯較 QR Code 佳。
- 7、 使用 Colour Matrix 將可擴增條碼的使用範圍，例如用在倉儲管理(Eversmann, 2019)等。

(二)、 使用限制

- 1、 Colour Matrix 需彩色輸出及彩色掃描，實務上會較使用黑白印刷的 QR Code 價格貴及不方便。
- 2、 Colour Matrix 辨識和解碼侷限於拍攝距離≤32.5 公分及拍攝傾角≤60 度。

(三)、 未來展望

- 1、 改進 Colour Matrix 設計以提高辨識成功率及掃瞄距離，這是有待測試的項目。可能方法有放大定位格、改變定位格設計、使用 fiducial markers 來設計邊框等
- 2、 選用其他邊框辨識方法來克服 Canny Edge + Adaptive Threshold 的極限。可考慮深度學習或人工智慧的演算法，例如 Convolutional Neural Network (CNN)。
- 3、 加入容錯的設計，降低因 Colour Matrix 毀損或顏色辨識錯誤所造成的解碼失敗。
- 4、 可以加入 image enhancement、image filtering 等影像處理技術，來改善 Colour Matrix 的影像品質。

六、 參考文獻資料及其他

- Akkaynak, D., Treibitz, T., Xiao, B., Gürkan, U. A., Allen, J. J., Demirci, U., & Hanlon, R. T. (2014). Use of commercial off-the-shelf digital cameras for scientific data acquisition and scene-specific color calibration. *Journal of the Optical Society of America. A, Optics, image science, and vision*, 31(2), 312-321.
- Altman, N. S. (1992). An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, 46(3), 175-185.
- Bhardwaj, N., Kumar, R., Verma, R., Jindal, A., & Bhondekar, A. P. (2016). Decoding algorithm for color QR code: a mobile scanner application. *2016 international conference on recent trends in information technology (ICRTIT)*, 1-6.
- Eversmann, H. Y. N. (2019). *Computation offloading of augmented reality in warehouse order picking*. (Bachelor). University of Twente, Retrieved from <http://essay.utwente.nl/80710/>
- Grillo, A., Lentini, A., Querini, M., & Italiano, G. F. (2010). *High capacity colored two dimensional codes*. Paper presented at the Proceedings of the International Multiconference on Computer Science and Information Technology.
- Jancke, G. (December 18, 2007). High Capacity Color Barcode. *High Capacity Color Barcodes (HCCB)*. Retrieved from <https://www.microsoft.com/en-us/research/project/high-capacity-color-barcodes-hccb/>
- John, R. A., & Raahemifar, K. (2015). Designing a 2D color barcode. *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 297-301.

- Mishra, A., & Mathuria, M. (2017). A Review on QR Code. *Int. J. Comput. Appl*, 164(9), 17-19.
- Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5. *J. Artif. Int. Res.*, 4(1), 77–90.
- RPi USB Webcams. (September 15 2020). Retrieved from https://elinux.org/RPi_USB_Webcams
- Yang, Z., Cheng, Z., Loy, C. C., Lau, W. C., Li, C. M., & Li, G. (2016). Towards robust color recovery for high-capacity color QR codes. *2016 IEEE International Conference on Image Processing (ICIP)*, 2866-2870.
- Yang, Z., Xu, H., Deng, J., Loy, C. C., & Lau, W. C. (2018). Robust and fast decoding of high-capacity color QR codes for mobile applications. *IEEE Transactions on Image Processing*, 27(12), 6093-6108.
- Zhang, Q., Zhuo, L., Li, J., Zhang, J., Zhang, H., & Li, X. (2018). Vehicle color recognition using Multiple-Layer Feature Representations of lightweight convolutional neural network. *Signal Processing*, 147, 146-153.

【評語】 190022

1. 設計彩色二維條碼可以存入多個條碼，有原創性。
2. 利用 knn 做色彩辨識可以採更先進技術。
3. 有實務應用潛力。
4. 缺少容錯方面的評估。