

2018 年臺灣國際科學展覽會 優勝作品專輯

作品編號 100010

參展科別 工程學

作品名稱 對弈機器人運用於預防及延緩失智症之研究

得獎獎項 大會獎：三等獎
國際科學博覽會正選代表

就讀學校 臺北市立內湖高級工業職業學校

指導教師 陳昭安、陳永華

作者姓名 陳皇宇、黃一烜

關鍵詞 人機對弈、人工智慧、失智照護

作者簡介



作者照片（左黃一烜、右陳皇宇）

我們是來自內湖高工資訊科 MBLPJ Project 中的兩位成員，這個團隊是由五位學生所組成的，大家都有各自在這個團隊中的工作也有理想與抱負，但大家還是會以團隊的理想為前進目標及方向，我們感謝一路上幫助我們的許多人，不論是正向的給予我們鼓勵亦或者是負面的刺激我們使我們成長。我們會繼續努力邁向我們的目標。

我是陳皇宇，上了高職後我熱衷於寫一些酷炫小遊戲，因此負責撰寫作品的人工智能，身為一位酷炫人，最喜歡在程式裡加一些酷炫的隱藏功能，讓使用者有驚奇的感覺，希望未來能有更多有趣經驗充實自己。

我是黃一烜，是團隊中的吉祥物，負責和諧團隊氣氛，雖然平時動作慢半拍、行動力差而且神經大條，但總能記住些細枝末節的事情，在關鍵時刻會變得比任何人還要冷靜，除了鑽研資訊領域的技術外，也喜歡繪畫、學習日文等興趣。

摘要

本研究目標在於改良對弈機械人及分析長期對弈資訊，運用於早期老人失智症之預防。據文獻及詢問專家後得知對弈遊戲具有預防失智症狀的功效，而研究中的對弈機器人以大腦功能狀態評估（JOMAC）為研究方向，研究資料分析如「玩家思考時間、對弈棄局次數、下錯頻率」等資訊並設計檢測橋段等機制以符合評估項目達到預防提醒之目的。對弈人工智慧，針對玩家棋力會自動調整機器人的棋力。採用 UCT 演算法，透過搜尋模擬能計算出各點落子勝率，藉由選擇較高或較低勝率的落子點，使人工智慧棋力與使用者旗鼓相當，保持使用者遊戲體驗及強化思考使失智症狀得以延緩。此外運用類神經網路改良影像自動校正，利用學習的方式解決不穩定環境光下誤判過多的問題，且提高校正速度，使對弈過程更加穩定。

Abstract

The purpose of this study is to improve the chess robot and analyze the long-term game information, and apply it to early dementia prevention. According to the literature and experts, the game has the effect of preventing dementia, and the chess robot to the state of mental status examination (JOMAC) as the research direction, research data analysis such as "length of player think time, Under the wrong frequency "and other information and design inspection part and other mechanisms to meet the assessment project to achieve the purpose of prevention alerts. The game of artificial intelligence, the robot's strength automatically adjust from the player's strength, by using UCT algorithm, through the search simulation can calculate the winning rate of each point, by selecting the higher or lower winning rate of the drop point, keeping the artificial intelligence and user strength as same strength. Quite equal to maintain the user experience which was good and strengthen the game to dementia symptoms which can be regulated. In addition, using neural networks to improve the image automatic correction, the use of learning to solve the problem of too much misjudgment in unstable ambient light, and improve the correction rate, let the game process be more stable.

壹、前言

一、研究動機

透過學校安排的校外教學，有幸參觀建置為工業 4.0 規格的企業，該工廠大量使用機械手臂提高生產力並減少人力及人為錯誤，最重要的可以因應多樣化的智慧生產需求。由這一個啟發加上日前資訊研究社很熱衷於人機對弈的人工智慧黑白棋遊戲，於是構想：「如果建構一套系統可以讓人與機器人或機器人與機器人對弈，分析視訊了解盤局的變化，根據變化使用微電腦運算下一步棋的決策，根據決策指揮機器手臂正確的運作落子、取棋動作，似乎是個有趣的方向」因此實現了對弈機器人系統。

在對弈機器人研究過程中，因緣際會下，從參觀我們作品的觀眾中意外得知極早期失智症測量表(AD8)的存在，又因高齡化趨勢的來臨，且發現原研究「使用機械視覺及人工智慧的黑白棋人機對弈機械手臂」之成果與其測量表許多項目吻合，因此本次研究著重於優化對弈系統及分析對弈機器人系統中的對弈資料，進而運用於早期失智症預防與提醒，讓使用者能在閒暇之於保持優良嗜好進而活化大腦思考以達預防失智症之功效。

二、研究目的

在玩家與對弈機器人對弈的過程中蒐集對弈中所產生的數據如「玩家思考時間、對弈棄局次數、下錯頻率……」等資訊並記錄於資料庫中，配合大腦功能狀態評估(JOMAC)的評估項目，製作檢測項目，透過長期分析玩家狀況及觀察檢測結果觀察玩家在對弈過程中的改變，若有大幅改變如「對弈時間異常拉長、忘記對弈規則、棋力大幅下降」等狀況適當給予照護人適當的提醒與建議以達預防失智症之目的。為了保持玩家的對弈新鮮感以及保持玩家持續思考的目的，系統會自動分析玩家棋力分佈，將玩家棋力分析結果納入 UCT 演算法選點依據，透過選擇較高或較低勝率的落子點，動態調整對弈人工智慧難易度，讓玩家能保持一定思考深度且保持一定勝率，以達持續思考延緩失智症狀。

貳、研究過程或方法

研究流程主要分為三個部分，研究類神經路應用於影像辨識、實地訪康寧大學相關科系進行需求訪查，了解需求及方向、根據 JOMAC 評估項目做出符合項目之檢測機制及功能，進而分析檢測數據，研究流程如下圖：

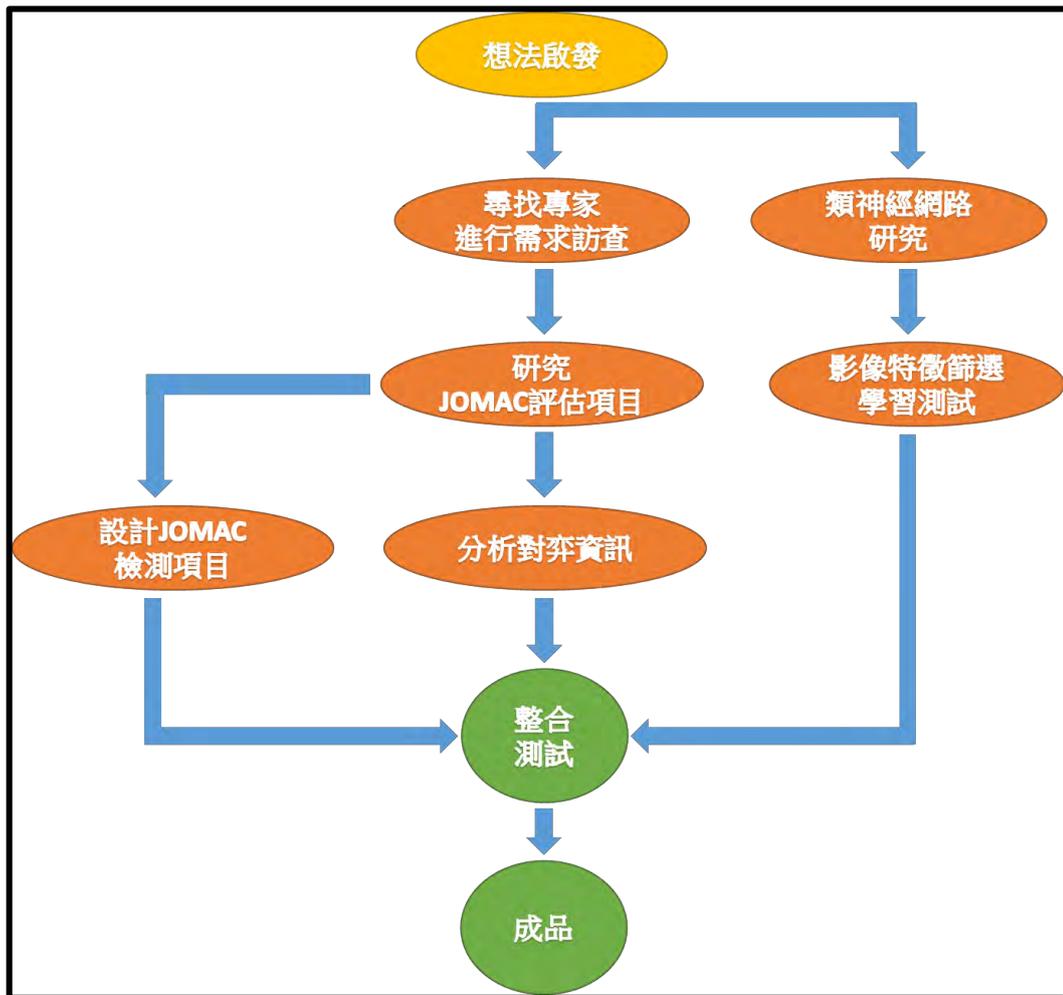


圖 1 研究流程圖

一、系統架構說明

系統架構依照硬體的功能，如圖 3 參考工業 4.0 建構網宇實體系統與雲端資料處理兩個部分。

(一)網宇實體系統：

主要是由網路攝影機、電腦控制系統與機械手臂組成一個閉迴路系統，工作平臺為黑白棋的棋局。透過電腦控制系統分析影像所得數據指揮手臂完成準確動作。符合網宇實體系統感測器和致動器裝置的整合控制系統。而資料庫所收集的資訊亦可作為使用者行為分析的用途。

(二)雲端資料處理：

由影像分析程式及人工智慧黑白棋程式所構成，影像分析程式主要是負責將接收到的影像處理後轉換為數據，傳送至人工智慧黑白棋程式進行運算，並依據運算完的資料進而選出最優良或最有實驗價值的位置，並且會將運算結果傳送至雲端資料庫保存，以利於下次進行搜尋時能夠增加整體的思考深度。

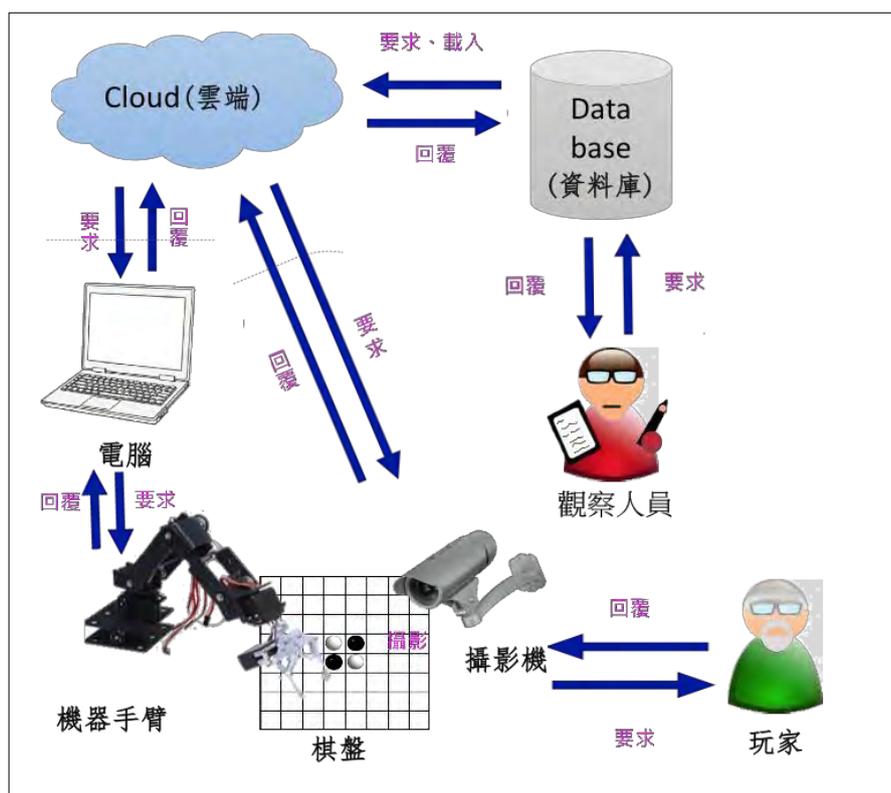


圖 2 參考工業 4.0 的系統架構示意圖

二、失智症分析系統

(一)AD8：

表一：AD8 極早期失智症簡易測量表內容

題 目	有改變 /1 分	無改變 /0 分	不知道 /不適用
1.判斷力上的困難: 例如落入圈套或騙局、財務上不好的決定、買了對受禮者不合宜的禮物。			
2.對活動和嗜好的興趣降低。			
3.重複相同的問題、故事和陳述。			
4.學習如何使用工具、設備、和小器具上有困難。例如:電視、音響、遙控器、冷氣機、洗衣機、熱水器、微波爐等。			
5.忘記正確的月份和年份。			
6.處理複雜的財務上有困難。例如:個人或家庭的收支平衡、繳費單、所得稅等。			
7.記住約會的時間有困難。			
8.有持續的思考和記憶方面的問題。			

1. 在下棋時，在黑白棋的規則內能限制下棋的選擇，當使用者在使用該系統時會提示可

ID	Name	Age	Sex	Phone_Number	Account	Password	Next_Date	Date_Period
1	[REDACTED]	89	男	[REDACTED]	[REDACTED]	[REDACTED]	20171101	2
2	[REDACTED]	83	女	[REDACTED]	[REDACTED]	[REDACTED]	20171112	1
4	[REDACTED]	93	男	[REDACTED]	[REDACTED]	[REDACTED]	20171116	1
5	[REDACTED]	76	女	[REDACTED]	[REDACTED]	[REDACTED]	20171015	1
6	[REDACTED]	90	女	[REDACTED]	[REDACTED]	[REDACTED]	20171018	2
7	[REDACTED]	63	男	[REDACTED]	[REDACTED]	[REDACTED]	20171211	1

圖 3 玩家資料庫圖

圖 3 為使用者帳戶內容及下一次約定時間(Next_Date)內容

(為保護使用者已針對姓名、連絡電話及帳戶內容進行處理)

8. 透過紀錄使用者的每步對弈時間及該場總花費時間做統計，因每次思考及記憶時間，如有異常則很容易無法完成一局遊戲，所以可藉由每次的時間差及錯誤率做為該項指標。

(二)大腦功能狀態評估(JOMAC)；

1、判斷(Judgment):

在一個判斷力薄弱的使用者，可能會遇到在多次同一狀況下無法判斷正確行動的情況，因此可藉由統計在棋局中同一原因的錯誤量以及單步思考時間做統計，以此作為判斷(Judgment)的指標。

2、方向感(Orientation)

方向感較不好的使用者很容易對於現在的人地時等狀況較不熟悉，藉由機械去詢問使用者現在時刻、現在地點、看護人員的名字或家人名字等問題將結果紀錄於資料庫以統計使用者對於現況的熟悉度，以此作為方向感(Orientation)的指標。

3、記憶(Memory)

記憶較弱的使用者對於記憶事務、時間等能力較為薄弱，所以在進行棋局後，不管棋局是否有完成皆會詢問使用者下次約定的下棋時間，透過每次啟動與資料庫比對的誤差值做為統計，以此作為記憶(Memory)的指標。

4、專注力(Attention)

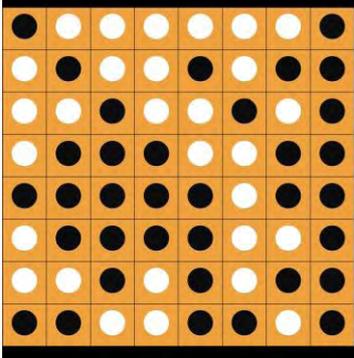
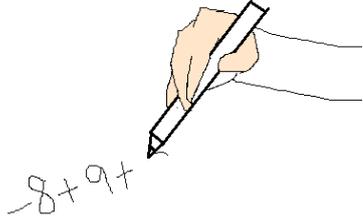
專注力較低的使用者很常無法進行較長時間的思考或是無法耐住性子不做任何思考就馬上下棋，因此可以透過使用者在棋局中，每場思考時間及是否完成局數等資料來做

統計，在長期觀察下來透過改善幅度來作為專注力(Attention)的指標。

5、計算(Calculation)

計算力較弱的使用者無法進行連續的簡易計算(如：連續加法、連續減法等)，透過棋局最後的小遊戲，利用圖像及數字對應要求使用者進行簡易計算，如在圖像行進路徑中，遇白則加、遇黑則減等，透過使用者最後答的答案誤差作為統計，以誤差情況做為計算(Calculation)的判斷指標。

表二：小遊戲示意圖

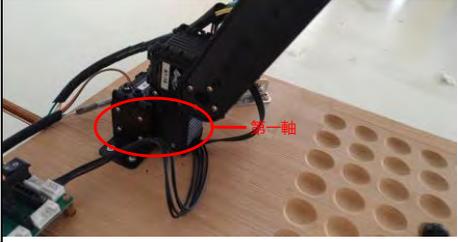
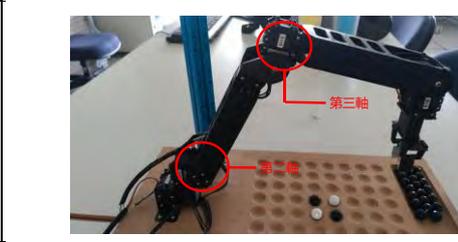
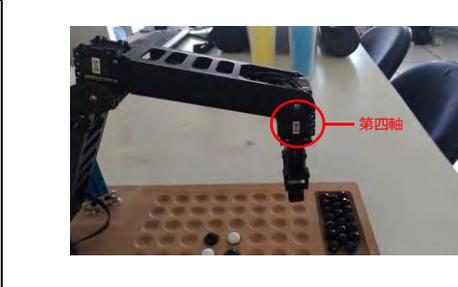
	<table border="1"> <tbody> <tr><td>5</td><td>8</td><td>6</td><td>1</td><td>2</td><td>3</td><td>5</td><td>4</td></tr> <tr><td>7</td><td>8</td><td>9</td><td>6</td><td>5</td><td>4</td><td>1</td><td>2</td></tr> <tr><td>3</td><td>6</td><td>5</td><td>4</td><td>1</td><td>2</td><td>5</td><td>8</td></tr> <tr><td>7</td><td>4</td><td>6</td><td>6</td><td>8</td><td>5</td><td>4</td><td>8</td></tr> <tr><td>9</td><td>6</td><td>3</td><td>2</td><td>8</td><td>5</td><td>2</td><td>2</td></tr> <tr><td>4</td><td>8</td><td>2</td><td>2</td><td>2</td><td>5</td><td>5</td><td>6</td></tr> <tr><td>3</td><td>2</td><td>5</td><td>8</td><td>9</td><td>6</td><td>3</td><td>2</td></tr> <tr><td>2</td><td>4</td><td>6</td><td>6</td><td>3</td><td>2</td><td>5</td><td>5</td></tr> </tbody> </table>	5	8	6	1	2	3	5	4	7	8	9	6	5	4	1	2	3	6	5	4	1	2	5	8	7	4	6	6	8	5	4	8	9	6	3	2	8	5	2	2	4	8	2	2	2	5	5	6	3	2	5	8	9	6	3	2	2	4	6	6	3	2	5	5	
5	8	6	1	2	3	5	4																																																											
7	8	9	6	5	4	1	2																																																											
3	6	5	4	1	2	5	8																																																											
7	4	6	6	8	5	4	8																																																											
9	6	3	2	8	5	2	2																																																											
4	8	2	2	2	5	5	6																																																											
3	2	5	8	9	6	3	2																																																											
2	4	6	6	3	2	5	5																																																											
結束棋盤示意圖	電腦提示及數字	實際計算(示意圖)																																																																

此小遊戲為針對失智老人欠缺的空間感及計算力還有判斷力的簡易遊戲，透過螢幕指示 2D 與實體 3D 棋盤狀況的空間連結，並以簡單的黑白判斷判斷加減，最後再進行簡易的連續計算，透過此手法可同時訓練空間感、計算力及判斷力等能力。

三、機械手臂製作過程

本研究著重於將黑白棋人工智慧程式判別的結果，使用機械手臂與人進行人機對弈。棋盤採用木板利用 CNC 進行雕刻，機械手臂的馬達以 AI 伺服馬達為主，另外夾爪採用一般的 RC 馬達。部分支架採用 3D 列印製作手臂、連接臂及夾具等部件。再利用螺絲將手臂各部分連接及固定，以建構機器手臂及棋盤與機械手臂的組合。

機械手臂必須要 5 個旋轉軸，分別定義為 A、B、C、D、E 軸。設計的說明如下：

A		<ol style="list-style-type: none"> 1. 用途：帶動水平方向的旋轉。 2. 旋轉角度：$-165^{\circ}\sim 165^{\circ}$
B、C		<ol style="list-style-type: none"> 1. 兩軸及兩臂形成三個主要支點。 2. B 軸旋轉角度：$0^{\circ}\sim 135^{\circ}$ 3. C 軸旋轉角度 $0^{\circ}\sim 90^{\circ}$
C		<ol style="list-style-type: none"> 1. 使機械手臂的爪子能夠與目標物互相垂直，使機械手臂在取子時，不會輕易的碰撞到其他棋子。 2. 旋轉角度：$0^{\circ}\sim 90^{\circ}$
D		<ol style="list-style-type: none"> 1. 以夾子形狀的構造來夾取棋子。 2. 使用 RC 伺服馬達。 3. 旋轉角度：$60^{\circ}\sim 120^{\circ}$

另外，各個轉軸點的伺服馬達如果有過載的情形，RC 伺服馬達並不會有任何保護作用，必須立刻斷電，否則馬達很容易燒壞，使用 AI 伺服馬達可以防止過載，當感測過載的時候，馬達會立刻斷電，以保護馬達。

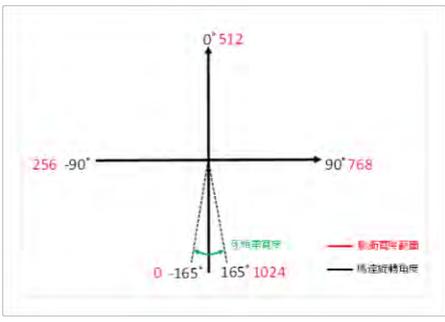
	
<p>AI 伺服馬達正面有一個轉盤，可以固定機械臂或其他機構件。反面有兩個串列連接孔可以串接伺服馬達。</p>	<p>設定伺服馬達的旋轉角度必須使用 0-1023 來換算成 0~330 度的角度數值，因此其解析度為 $330 / 1024 = 0.322265625^\circ$。</p>

圖 4 AI 伺服馬達外觀、構造及角度示意圖

程式撰寫上，每顆馬達均有自己的 ID，運用 SetPositionI_JOG 函式便可驅動馬達旋轉，範例如下：

```

void setup()
{
  Serial.begin(115200);
  A1_16_Ini(115200); // 設定鮑率為115200
  SetID(254,1); // 初始化馬達編號為1
}
void loop()
{
  SetPositionI_JOG(1,200,512); // 執行馬達動作 (第1馬達 · 0.2s · 角度)
  delay(1500);
  int p1=ReadPosition(1); // 讀取馬達目前角度
  Serial.println(p1);
  SetPositionI_JOG(1,200,0); // 執行馬達動作 (第1馬達 · 0.2s · 角度)
  delay(1500);
}

```

圖 5 Arduino 撰寫程式簡易示意圖

四、機械手臂的相關公式堆導

(一) Z 軸旋轉角(θ)及手臂長度 (L) 計算假設原點為了求得手臂與棋子的距離和角度，先在棋盤上方先標出與其水平、垂直的 X 軸和 Y 軸，將電腦中定義的格子示意如下圖所示。若 P 點的位置在 (L_x, L_y) 的位

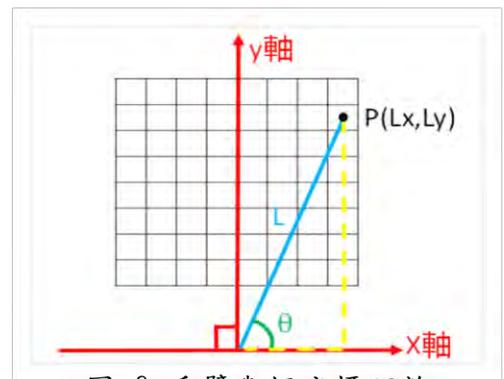


圖 6 手臂與極座標距離之

置，利用畢氏定理推算出機械手臂中心點到目標物的直線距離 L 。再將 L_y 與橫軸 L_x 利用反三角函數推算底座馬達 θ_{10} 的旋轉角度。

$$L = \sqrt{L_x^2 + L_y^2} \quad \theta = \tan^{-1} \frac{L_y}{L_x}$$

(二) 各軸馬達轉動角度換算

為求得第 2、3、4 軸伺服馬達轉動的角度 θ_1 、 θ_2 、 θ_3 為多少。由上面的說明，已經可以得知俯視的 L 長度，為進一步求得 θ_1 、 θ_2 、 θ_3 機器手臂的架構如圖 6 幾何架構，可以推算出適當的數學公式。

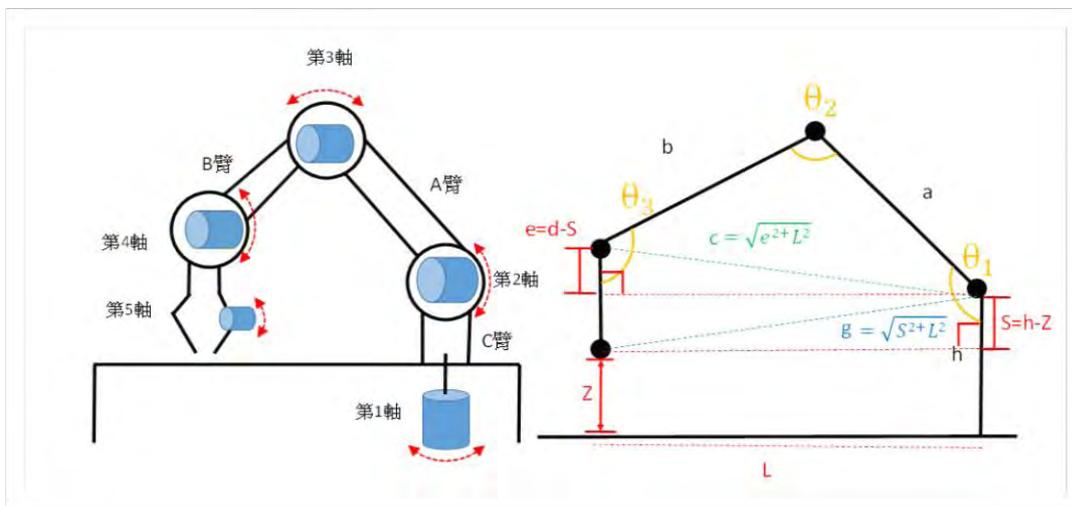


圖 7 機械手臂幾何說明

由圖 7 幾何架構可以利用餘弦定理和上圖所表示出的 a 臂、b 臂、c 臂，兩者之間求出三個內角 θ_1 、 θ_2 及 θ_3 ，公式如下：

$$\theta_3 = \cos^{-1} \frac{c^2 + e^2 - l^2}{2ce} + \cos^{-1} \frac{b^2 + c^2 - a^2}{2bc}$$

$$\theta_2 = \cos^{-1} \frac{a^2 + b^2 - c^2}{2ab}$$

$$\theta_1 = 360 - \theta_2 - \theta_3$$

經過餘弦定理的運算後，就可以利用手臂在四個點的位置去推算其餘 60 點的位置，因為棋盤為正方形，各棋目之間的距離又相等，所以只要抓出四個點的位置，就可以使機械手臂完成所有位置的取子動作。

與前一代相比，整體架構有許多的改變，為了使系統架構更加人性化和減輕重量，在系統上面改變了材質與使用的設備，以下是所有新增的項目：

- (一) 利用智慧手機的 IPCAM 來取代以往監視用的 IPCAM，不但迎合現代人手機的高持有率，未來也可以在智慧手機上新增互動式的介面，使下棋更加有可玩性。
- (二) 利用鋁合金的支架取代以往笨重的鋁擠，使其重量減輕。
- (三) 利用 3D 列印 PLA 的材質，取代前代手臂較重的金屬支架。
- (四) 不使用滾珠軸承的方式控制水平旋轉移動，原因 AI 伺服馬達有較高的力矩，又為減輕系統整體的重量，改由單顆 AI 伺服馬達進行動作。
- (五) 以往傳統伺服馬達為並接控制，因此可能產生雜訊而使用了舵機控制板，現在因為 AI 伺服馬達採用串接的方式控制，所以目前使用 ATmega1280 Arduino 馬達控制板，直接對馬達下達命令。
- (六) 為了使黑白棋遊戲更加的完整，由一開始的 6*6 的棋盤改為正式遊戲所使用的 8*8 棋盤，而考量玩家取子、落子的方便性，在棋盤上面刻有雙方棋子的放置點，讓遊戲進行時更加流暢。

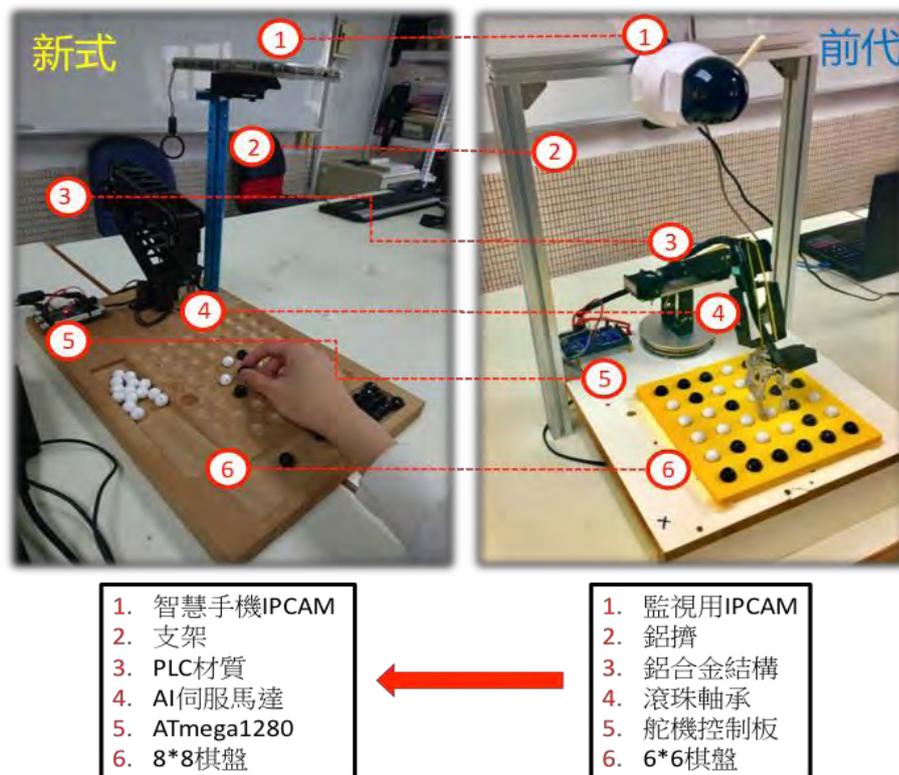


圖 8 新式與前代的比較

五、人工智慧黑白棋的設計

黑白棋又名翻轉棋，是 19 世紀末的英國人所發明的。遊戲方式是將對方的棋夾住後能將對方的棋子換成我方的棋子，在對弈的時候只要有能將對方的棋子換成己方的位置就一定要落子，若沒有能將對方的棋子換成己方的棋子時則換對方下，當雙方皆無法落子或是棋盤上沒有空位時則遊戲結束，以棋盤上棋子數多的人獲勝。

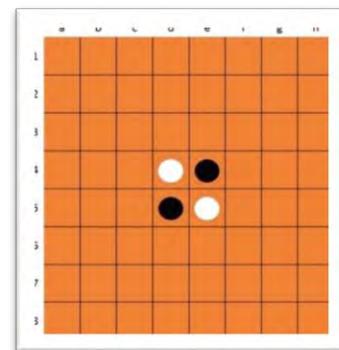


圖 9 黑白棋的棋面

(一)位置判斷法

使用翻棋數量的方法可以在遊戲前期取得很大的棋子數量的優勢，但是因為黑白棋獨特的規則，很容易使雙方棋子的數量劇烈變化。有可能在遊戲後期可能僅用幾個回合就將大量對方棋子變成己方，從而扭轉局勢。因此，太著重於棋子數量是沒有必要的，更重要的是占據有利的位置。

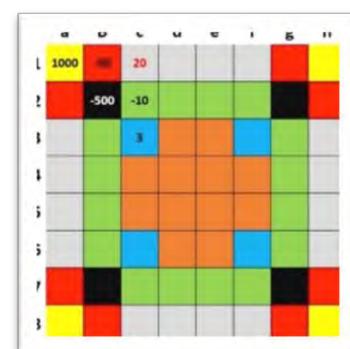


圖 10 各位置的加權分數

在撰寫這部分的人工智慧時，歸納出以下幾個重點

1. **黃色位置**：下在角落的棋子是絕對不會被對手翻面，所以它是最重要的點，而且可以確保邊上的棋子。
2. **紅色位置**：若是落子在此位置，會使黃色位置被搶走的機會變高，但是因為邊上的點只會有一個方向被翻面，所以危險性較低。
3. **黑色位置**：是棋盤上最需要小心的位置，雖然他只有一個方向會使角落被翻面，但由於棋盤中間的棋子很容易翻來翻去，使得此位置上的點很容易讓角落被搶，所以在下這個位置時要非常小心。
4. **灰色位置**：只會有一個方向會使它換顏色，而且占了這個位置後可一次翻許多棋，而且也可以提高搶到邊的機率。
5. **綠色位置**：若是落子在這裡的時候，會開始進行搶邊的動作。而這通常對先落子於此位置的人來說有些許不利。
6. **藍色位置**：剛開局時很重要的位置，因為落子於那個位置後，對方若想要將它翻面，必須要下在綠色的位置。

(二) 遊戲樹

位置判斷法從最重要的從黃色位置推導出每一部分的權重，可以使 AI 優先選擇經驗上較好的位置。但在棋局變得較激烈時，由此簡單邏輯的判斷會變得不夠精確，若直接在棋局中推測對手下一步落子位子，能更精確的掌握棋局發展進而取得勝利。

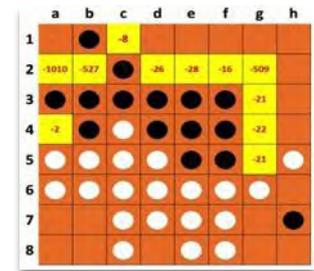


圖 11 遊戲數的加權值

遊戲樹方法會透過不斷的推測自己以及對手可能落子的點進行計算以找出最佳落子位置。以圖 11 的 a4 位置為例：AI 落子在邊上，所以這點的加權值加上 20 可以翻一粒敵方的棋子，在第二圈所以加權值減上 10。接下來要推測對手可以下的位置，假設對手會選擇加權分數最高的位置下，所以可以算出圖 4 的加權表。想要推算更後面的棋局只要重複以上的步驟就可以進行。最後再把我方的加權分數減上對方的加權分數後就可以得到一個遊戲樹推算出的加權分數。

在這個遊戲樹演算法當中，有額外加入了 MiniMax 搜尋法、 α - β 剪枝演算法跟第二版位置判斷演算法以及翻棋樹演算法進去作為輔助。加入 α - β 剪枝是由於搜尋層樹的提升導致過多不必要的搜尋拖慢了整體搜尋的速度，因此加入此演算法刪剪掉較為不重要的節點以提升搜尋速度；而 MiniMax 搜尋是找出對於我方加權值高（極大）的地方以及對手加權值高（極小）的地方，判斷兩方各別的下棋走向，尋找出最為符合現實狀況的推論。

(三) UCT (Upper Confidence Bound 1 applied to trees)

以上述的遊戲樹 AI 與線上玩家進行多場對弈後發覺，在對於棋力為中階左右的玩家仍可以保持較高的勝率，不過對於較高階的玩家卻下的有些吃力，由於高階玩家時常使用長時間布局的方式將局面控制在他的理想狀態中，等待時機成熟進而一舉反攻導致我方輸棋。而我方的演算法是透過遊戲樹的搜尋來模擬對方的下法，不過若對方是以布局的方式下棋，在前期會故意削弱自己地盤的優勢以提升局末機動性，

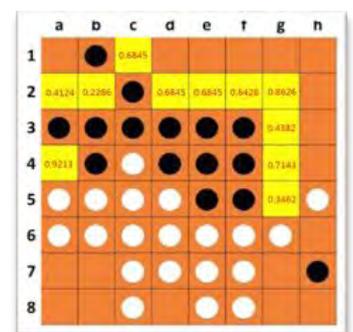


圖 12 UCT 各可落子點的權重

因此遊戲樹演算法無法準確預估對手行動，將一步一步掉入對手陷阱中。

因此希望能夠透過學習的方式讓我方 AI 在對弈的過程中學習對手棋步，經過多次對弈後能夠自我修正以探索出勝利棋步取得勝利。經過資料的查詢後找到了 UCT 演算法，由於 UCT 演算法的判斷依據來自於過往的對弈勝率以及探索的結果，與先前的固定依據不同，因此希望能夠過此演算法修正先前 AI 會掉入對手陷阱的問題。以下為組成 UCT 演算法的 3 種不同要素：

6、UCB (Upper Confidence Bound)

以上述的遊戲樹來說，它的加權判斷是以位置加權來做為依據，但在各種場合下，位置加權表上加權高的點不一定是好位置，也會因此容易掉入陷阱，所以透過統計學中的 UCB(上信賴界限)作為 UCT 的判斷依據，這能夠計算出下該點的價值。UCB 公式如下：

$$UCB = \frac{\omega_i}{n_i} + c \sqrt{\frac{\log T}{n_i}}$$

- ω_i 代表 i 點模擬棋局獲勝次數
- n_i 代表 i 點模擬總次數
- C 為常數，當數值越高則勝率比重越低，探索慾望越大
- T 為所有點模擬的總次數，取 \log 平衡測資不均的問題

UCB 公式包括獲勝機率以及模擬該位置的次數。優先選擇尚未模擬的點進行模擬，確認此點是否能夠到達理想的獲勝結果。常數值 C 主要是影響 AI 的探索慾望的參數。當 C 值提高，勝率的比重會下降，AI 探索慾望將會提高，因此探索寬度會越寬闊。找到適當 C 值來取得探索廣度與深度的比重，也同時影響了 AI 學習的速率。透過這個公式，能使 UCT 所需的資料更有參考價值且將好的點做較深的搜尋以達到不平衡的遊戲樹搜尋。

7、蒙地卡羅 (Monte Carlo)

蒙地卡羅方法是利用隨機採樣的方式及龐大的模擬資料來取得近似解。此方法剛提出時由於硬體無法跟上演算法所需的龐大數據而並未受到重視。近幾年設備的快速提升下，蒙地卡羅方法也逐漸浮出檯面。蒙地卡羅方法常運用於數學求的最佳解的問題上也常用於決策遊戲中，若將事件以及許多條件設定好，也能模擬一些無法在現實世界中所

進行的實驗例如：核彈模擬、粒子碰撞等。

本研究不針對黑白棋的攻略進行 AI 設計，此外亦無較好的盤面評估方式可以判斷棋局優劣，因此若以蒙地卡羅樹搜索 (MCTS) 進行，利用勝率的統計得出較佳的棋步，由於 MCTS 的下棋依據來自於自我探索的結果，即使撰寫者的黑白棋棋力很弱，只要程式給予黑白棋的規則，就可探索出勝率較高的下法。

8、UCT 演算法

UCT 演算法是結合了蒙地卡羅方法及 UCB 公式的遊戲樹搜尋，此演算法與一般遊戲樹搜尋的不同之處在於其加權值 UCB 會在探索時不斷的更新，因此 UCT 在選擇搜尋點時並不會平均的將每個點進行平均尋而會對較好的點進行較深的探索，因此它屬於一個不平衡的遊戲樹搜尋，也因此不必額為加上剪枝法就能自我省略不必要的可能性。下圖為 UCT 演算法實際進行的流程圖，下圖中的數字為 (勝利次數/總模擬次數)，在經由 UCB 公式計算後能得出一個 UCB 值，下方流程圖中的選點依據來自於此。

下圖為 UCT 演算法實際進行的流程圖，說明如下：

- (1)、選點：透過 UCB 公式計算出各個可落子節點的 UCB 值，選出值最高的節點進行下一層的延伸探索。
- (2)、擴點：當已經沒有可以模擬的節點時，優先對無資料的節點進行模擬。
- (3)、模擬：亂數模擬直到分出勝負。
- (4)、更新完節：將勝負結果反向的更新到剛剛落子的節點，新增一筆勝/負的資料 (反向傳播)。

以上的四個步驟稱作 1 次輪迴，每一個節點的模擬都會衍伸上千或上萬次的輪迴模擬，讓數據更完整更具參考價值，藉不斷的刷新 UCB 值來取得較佳的落子點。

由於 UCT 演算法需要及龐大的資料以進行分析，而目前的設備無法符合蒙地卡羅的需求，因此透過資料庫的結合，將對弈過程中的棋譜、探索資料、棋譜及棋局結果等資訊上傳至資料庫，藉由分析過往對弈的數據進行近一步的棋局分析來達到經驗傳承的效果讓 UCT 演算法

具有學習及修正的能力。透過這個方式，UCT 演算法能在對弈的過程中自我探索、學習如何躲避陷阱，以達到這次人工智慧所期許的主要目標。

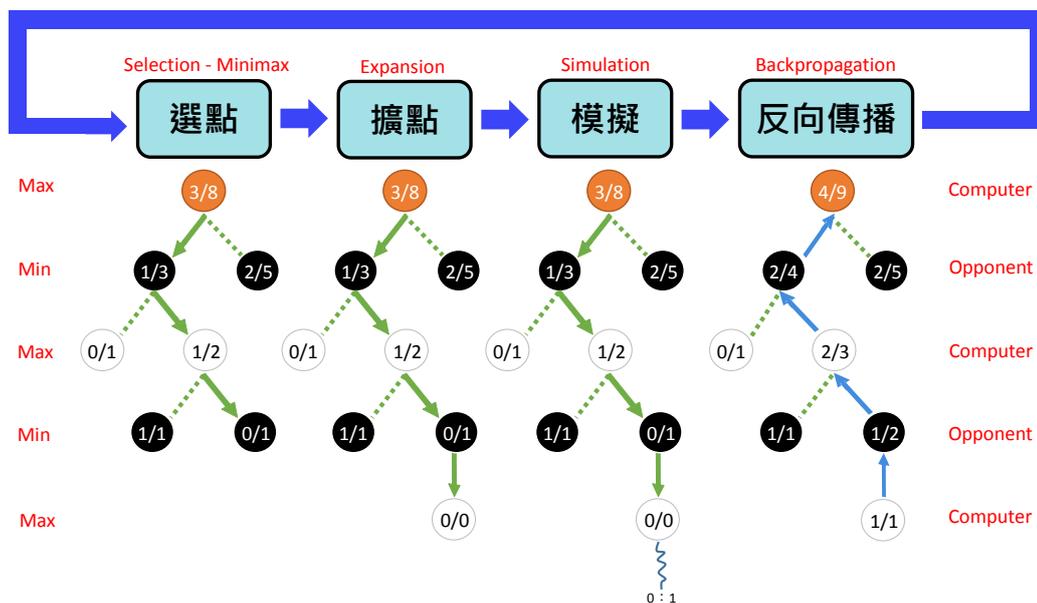


圖 13 運用蒙地卡羅樹搜尋的 UCT 流程

(四) RAVE 搜尋法

RAVE 搜尋法是建構在蒙地卡羅樹之上的搜尋法，其搜尋流程與 UCT 類似，唯差別在於最後的更新資料以及加權值的不同，在對弈時或許 A 點（棋盤上某一點）在模擬後得到敗場但是 A 點在其他棋局模擬時得到了不少勝場，所以推測 A 點或許也是個好點，因此在更新資料時多了一個「點」的權重。在黑白棋中有金角銀邊的觀念，也因如此，認為 RAVE 搜尋用於黑白棋或許是一個不錯的方法。透過此方法能讓 AI 更快速的學習到黑白棋的特性（金角銀邊）進而提升 AI 的學習速率，強化 AI 的學習。

六、影像分析

當棋盤上影像從雲端擷取後，必須進行影像處理，藉由判斷棋盤上棋子的位置、黑色、白色，來建構出一個棋局的陣列，供後續人工智慧的部份進行分析，以決定下一步的落子、取子、補子及違規，透過這些結果指揮機器手臂完成必要的動作。

(一)邊緣偵測

整個黑白棋對弈的流程中，有一部分需要進行影像處理轉成程式所需要的資料。影像處理時所需要最為重要的為圓形偵測以確保棋盤上目前對弈資訊，但在進行影像處理前，需先進行灰階化，因為只透過灰階明亮判別邊緣比含有色光資訊的邊緣，判斷上含有色光的部分反而還要在多上一些 RGB 的視覺亮度做運算，所以轉灰階能夠更加直觀的去做這類類比影像資訊轉數位資訊的影像處理系統。



(a) 灰階化的圖

(b) 邊緣偵測後的圖

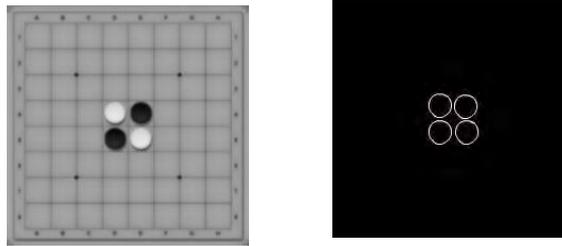
圖 14 邊緣偵測

通過灰階化後會因為背景有細小雜物而讓邊緣偵測錯誤的情況，將整張圖片進行高斯模糊後，細小的雜物會被模糊與背景同化，對於要偵測的大目標並不造成影響，而達到降噪的效果。降噪後進行邊緣偵測，當上面黑灰的圖畫成反明度的量圖時，會發現會有一個點的明度有巨大的變化，此時產生變化的 pixel 便是所需要找的邊緣。

(二)圓形偵測

使用的圓形偵測方式為霍夫找圓(HoughCircles)，霍夫找圓的找圓方式是利用圓方程式 $(x-a)^2 + (y-b)^2 = r^2$ ，其中 (a, b) 為圓心座標， r 為圓的半徑，圓形半徑是由作品所使用的棋子所產生，用這個二維數據，讓 (a, b) 在影像座標內不斷改變位置，最後當這三維數據組的

點數，超過訂定的閾值時就判斷為圓，而將圓繪製。



(a) 灰階化的棋盤

(b) 圓型偵測後的圖

圖 15 圓形偵測

將圓型偵測後繪圖的圓形畫在預設的黑色影像上，利用設定固定距離搜尋剛才的繪圖的結果，與高斯模糊後的黑白明度做比對讓程式知道黑白棋棋盤中目前哪一個棋在哪一個位置上，匯集成數位資料輸入於程式中傳進資料庫與程式，以便給人工智慧去判斷。

(三) 顏色偵測

在確立位置後可取得該圓的中心點 XY 目標點，可以透過將圖轉成 Bitmap 格式來取得該目標點的飽和度、明度來辨別，先利用該位置的飽和度辨別該點是否有棋，因有棋的情況下，只會有黑白等彩度偏低的顏色，所以當取該點的彩度有高於所設定的值，則可以先行判斷出此處為無子。假如此處有子，則利用明度偵測判斷該棋為黑棋或者白棋，來輸出成指定的圖片格式。

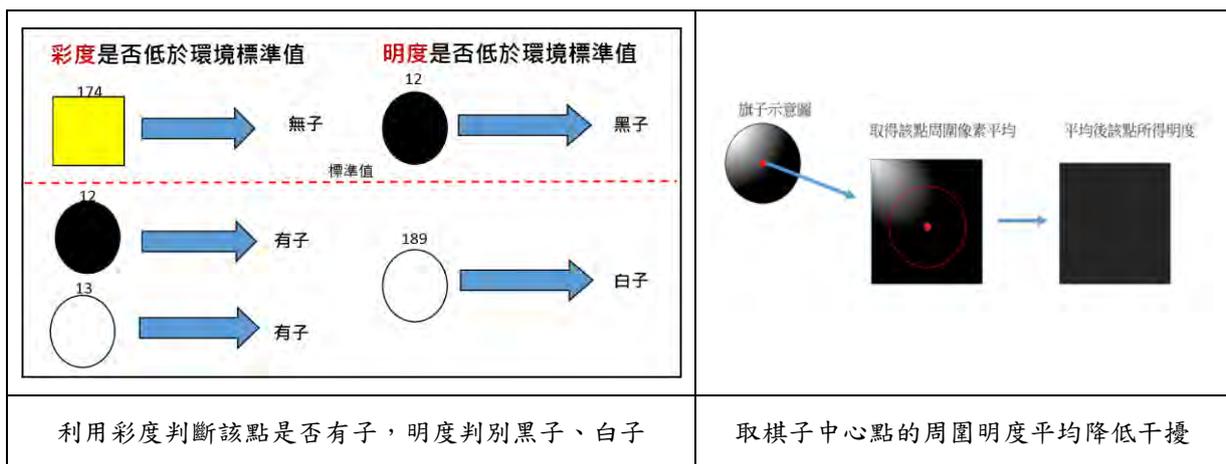


圖 16 黑白子的判別方法

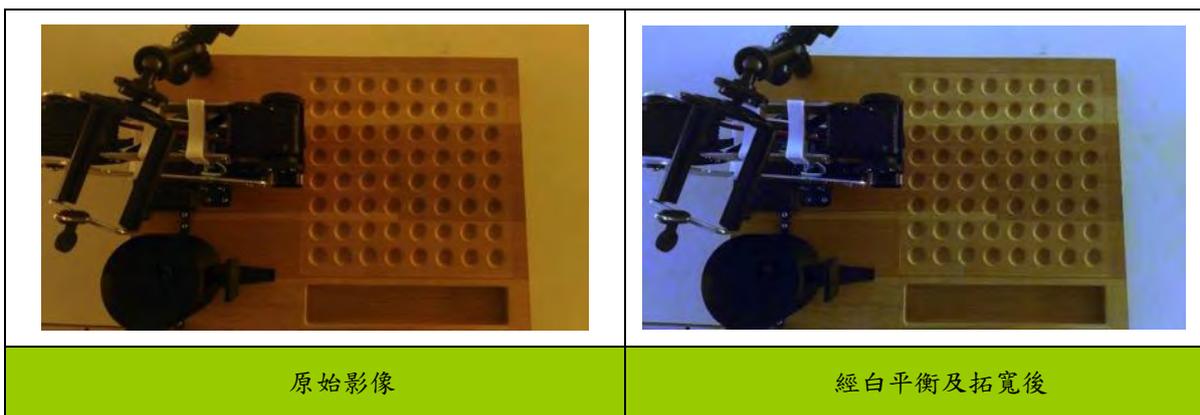
(四)白平衡演算法

從之前的研究結果得知，單純利用彩度與明度來判斷會因為環境色光的不同而導致棋子彩度大於棋盤彩度的情況，使影像就算調整彩度與明度的閾值也會有資料錯誤的情況，所以加入了白平衡演算法利用平均的 RGB 色光比例，增減每格像素的 RGB 色光來減少色光不均所造成的影響，以維持棋盤與棋子的彩度差異性。



(五)直方圖拓寬

此為影像增強的方法之一。雖然白平衡已經進行了對於色光比例的降噪，但本身影像的差異度並不會提高，故使用直方圖拓寬，針對每個點進行全局統計，運用全局統計的結果，可將每一像素把原強度範圍擴增至最大強度範圍，使影像的差異度等比例的提高，以利有效判斷範圍可以拓展。



(六)類神經網路

雖然經過白明衡及直方圖拓框等影像強化後，影像處理上卻實解決原先陰影擾動且減少環境色光對於影響處理的影響，不過在參展及比賽的過程中，由於會場環境光十分不均且陰影相當嚴重，甚至聚光燈打在上面造成嚴重反光，導致辨是區間大幅下降，嚴重時甚至需要調整攝影機，因此我們透過類神經網路，將影像處理資訊如：環境色光、平均亮度、直方圖平均值、HSV 色彩空間等資訊作為輸入資料，針對現場環境條件進行學習以增加影像處理的適應能力且更加快速的自我校正。

本研究採用倒傳遞類神經網路（Back Propagation Neural Network）將輸入數值乘上加權權重，並將數值透過活化函數活化，在傳至下一層，將環境條件加入輸入資料中，即使環境色光平均偏低或色光不均等問題都能透過不斷修正來適應當前環境以提高辨識度。

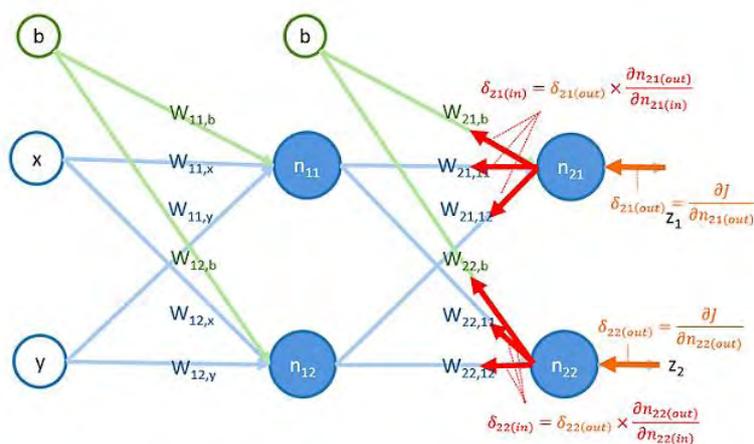
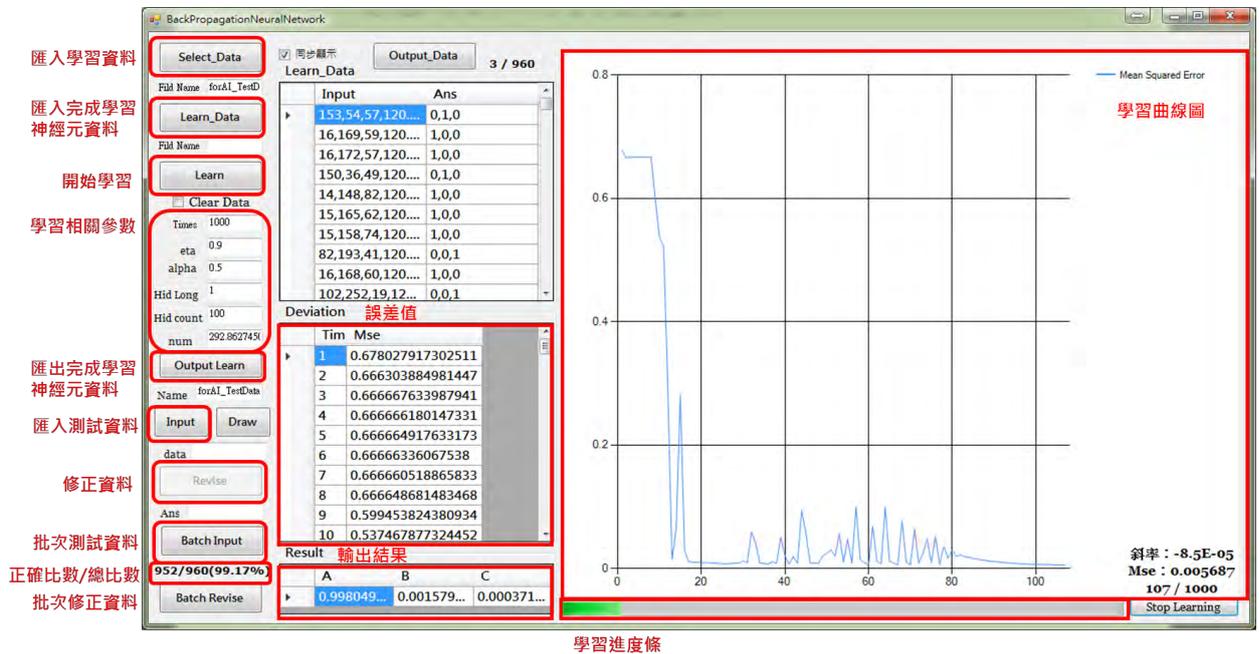


圖 17 倒傳遞類神經網路

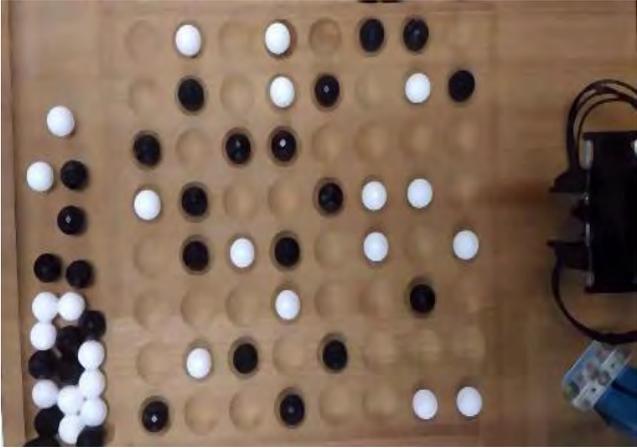
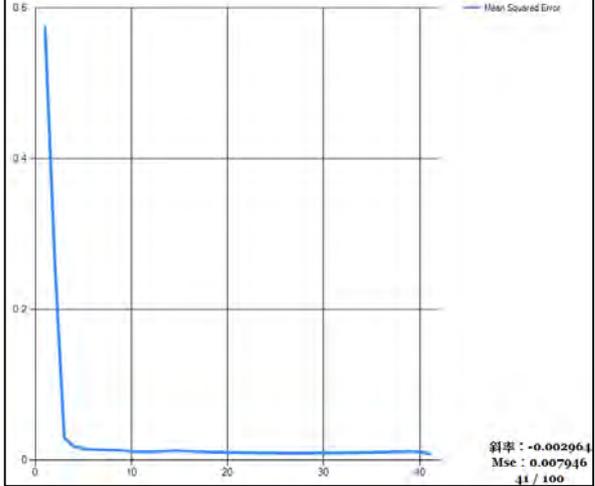
參、研究結果

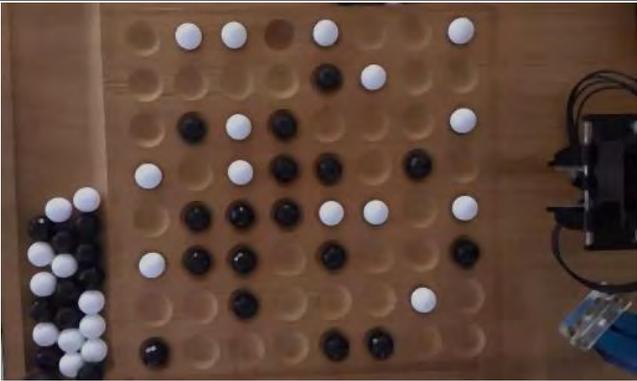
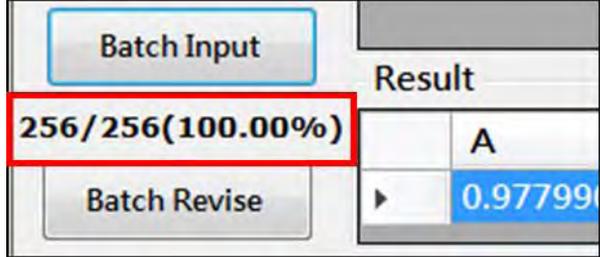
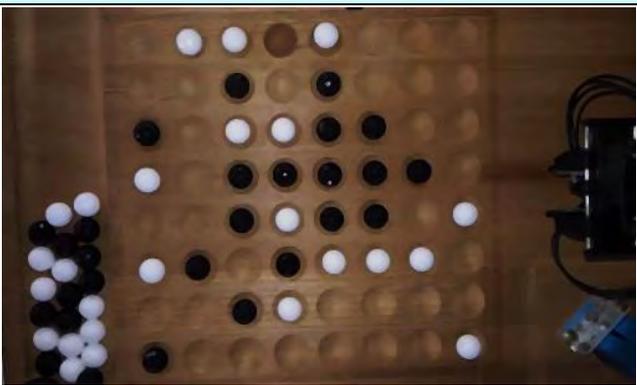
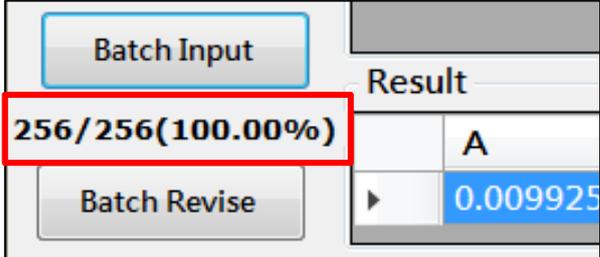
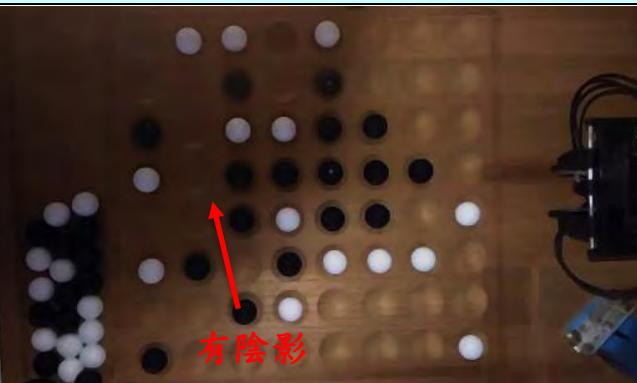
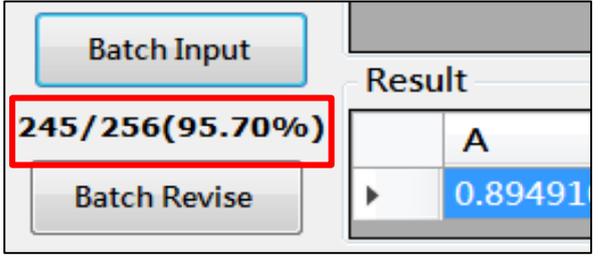
實驗一：使用類神經網路強化環境適應力



透過正常環境光源的圖片做為學習資料，透過正常資料測試是否可以在關燈、關燈後聚光、嚴重陰影下也能有高辨識度。

表三：學習曲線與辨識成效實驗表

學習資料 A：正常環境光(正常)	學習曲線圖
	 <p>斜率：-0.002964 Mse：0.007946 41 / 100</p>
<p>測試資料 B：關燈(亮度偏低)</p>	<p>256筆關燈資料辨識度達100%</p>

	
<p style="text-align: center;">測試資料</p> <p>C：關燈後從正上方打下閃光燈(光源不均)</p>	<p style="text-align: center;">256 筆開 LED 燈資料辨識度達 100%</p>
	
<p style="text-align: center;">測試資料</p> <p>D：關燈後從正上方打下閃光燈並產生陰影(嚴重干擾)</p>	<p style="text-align: center;">256 筆嚴重陰影資料辨識度達 95.7%</p>
 <p style="text-align: center;">有陰影</p>	

從以上實驗可得知，只利用正常光源下的資料學習的類神經網路，在環境條件改變後，依然能夠以 100% 的辨識率來分析棋盤，而有嚴重干擾(陰影)也可達 95%，由此可知透過類神經網路的學習能夠有效的提升影像辨識的適應能力且穩定度在一般狀態下，有近乎 100% 的辨識率。

實驗二：延緩預防失智症資料分析

針對資料庫所紀錄的資料進行分析，下圖為同一使用者與對弈機器人對弈的紀錄資料，包含勝負(Win)、對弈時間(Time)、下錯次數(Wrong)、人工智慧難易度(AI_Mode)、棋局是否完成(Done)、是否記得預約對弈時間(Remember)、完加編號(User_ID)，以下分析數據以每5比為單位，將時間平均，錯誤次數累加等方式分析使用者狀況並比對對弈日棋是法與上次預約日期相符以觀察記憶能力，並將異常狀態以紅色三角型標記。以下分析圖表的資料皆相同。

ID	Win	Time	Wrong	Date	AI_Mode	Done	Remember	User_ID
2365	1	17	25	20170925	1	0	1	7
2366	0	25	22	20170925	1	0	1	7
2367	0	12	22	20170926	1	0	1	7
2368	0	9	18	20170928	1	0	1	7
2369	1	20	26	20170929	1	0	1	7
2370	1	25	21	20171002	1	0	1	7
2371	1	16	18	20171004	1	0	1	7
2372	1	24	18	20171008	1	0	1	7
2373	1	18	23	20171012	1	0	1	7
2374	1	18	26	20171015	1	0	1	7
2375	0	15	24	20171017	1	0	1	7
2376	0	19	24	20171124	1	0	1	7
2377	0	12	20	20171028	1	0	1	7
2378	1	13	22	20171028	1	0	1	7
2379	1	58	15	20171028	1	0	1	7
2380	1	33	9	20171029	1	0	1	7

圖 19 對弈資料記錄圖

下圖為該使用者對弈的錯誤次數累計

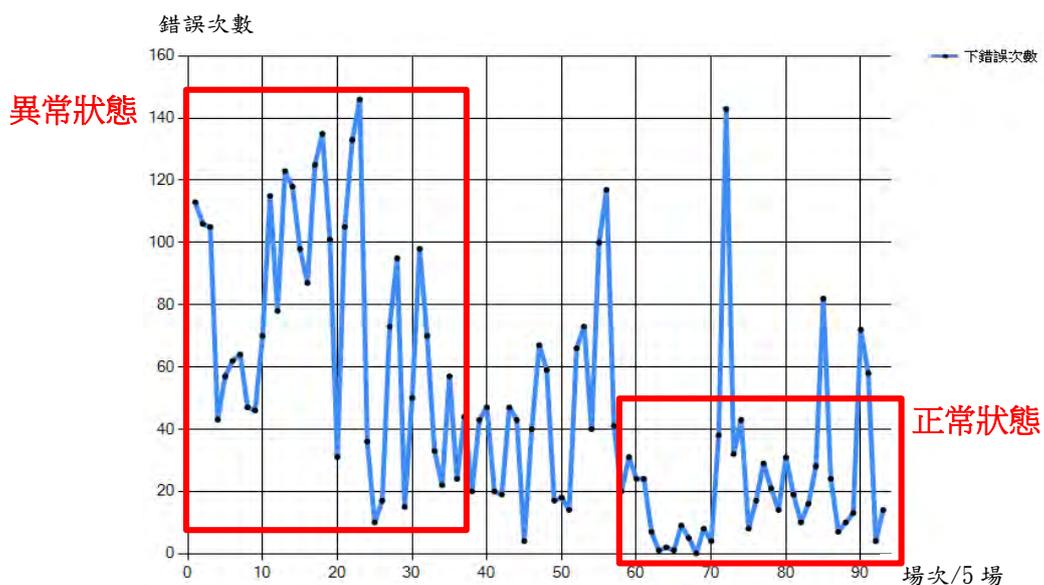


圖 20 錯誤次數記錄圖

圖 19 左邊下錯次數偏高，或許有忘記規則或落子無法落在指定位置的困難，因此有些異常，需透過以下資料加以檢視。

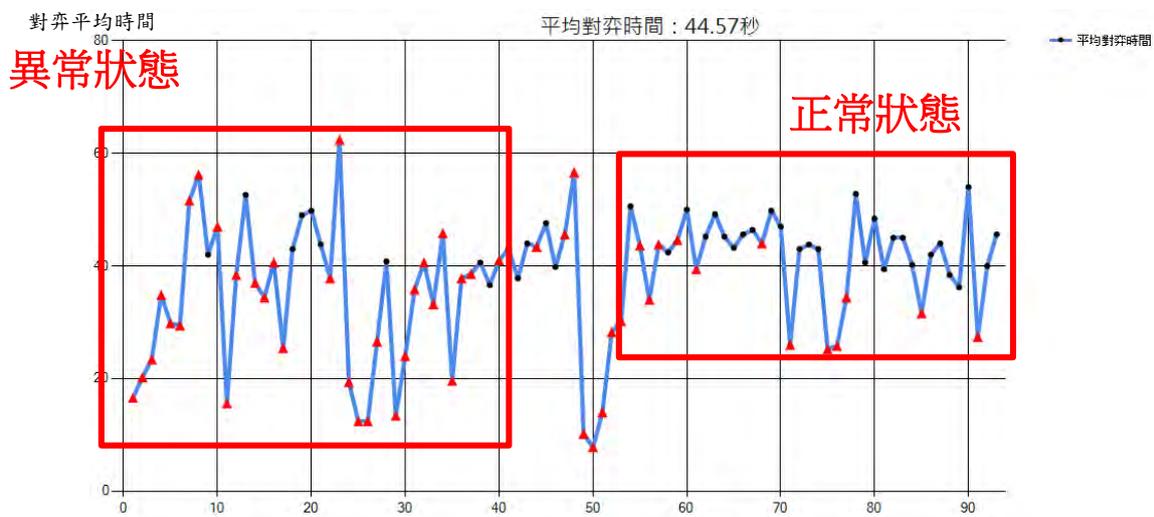


圖 21 對弈時間記錄圖

從左邊可以發現對弈時間十分不穩，有時 10 多分鐘就結束對弈，可能：場次/5 場 降 降的緣故，導致對同一事物無法持續專注；有時卻對弈了 1 小時多，由圖 19 資訊得知左邊錯誤多且時間長在加上即使玩了 1 小時多最後仍未完成棋局，因此使用者可能有忘記規則等徵兆，而到了右邊區塊可見時間相對平均許多且從第一張圖可知錯誤明顯減少，由此可知使用者有漸漸回穩的趨勢，是個好的徵兆。下圖是將二圖透過與平均對弈時間相檢，計算出與平均對弈時間的時間差，更能看出使用者的狀態。

計算出平均時間差後可明顯看出使用者對弈狀況，尤其正常狀態與異常狀態的穩定度明顯區別出來，在透過觀察使用者是否記得預約下棋日棋、是否完成棋局等資料謹慎觀察使用者狀態以做出適當提醒以達預防目的。

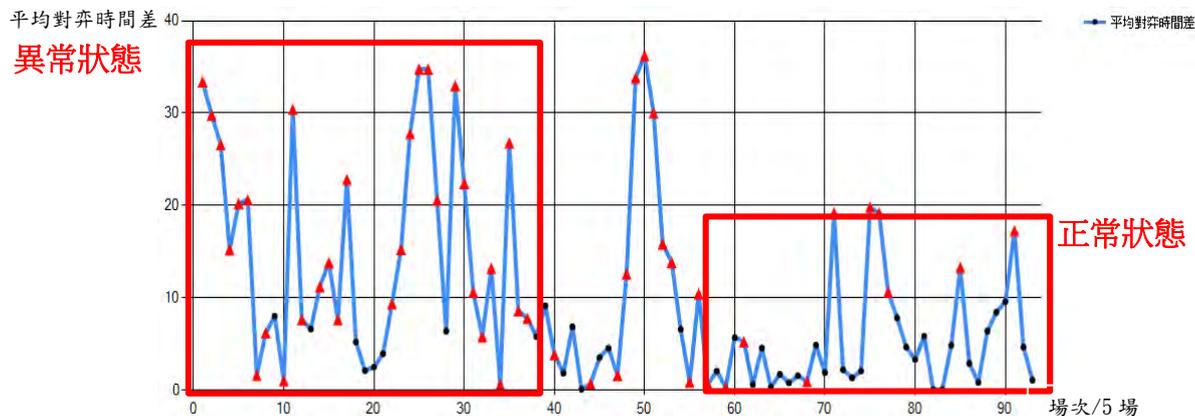


圖 22 對弈時間平均差

在與專家討論過後我們參考了 JONMAC 及 Mini-mental state Examination(MMSE 簡易智能測驗)並挑選了 5 項能用在黑白棋人機對弈系統的指標如：專注力(Attention)、判斷力(Judgement)、建構力(Constructed)等，並透過不同種類的錯誤分別給予不同的加權分數如下表：

	Calculation	Attention	Memory	Judgment	Constructed
Misplaced	0	1	0	2	1
Wrong again	0	2	1	3	1
No reverse	0	1	0	2	1
Give up	0	15	0	0	0
Time too long	0	8	0	6	8
Time too short	0	0	5	10	0
Doesn't know who is winner	0	0	8	15	0
Can not count pieces	10	4	0	0	4

圖 23 指標加權對應圖

在每場對弈結束後，系統將會統計分數並產生以下圖表：

- 一、狀態數值表：照護者可以透過此標分析使用者的認知功能狀態，黃色為同年齡層的標準值藍色為玩家的數值，若是藍色高於黃色則代表此位玩家可能需要多加注意。

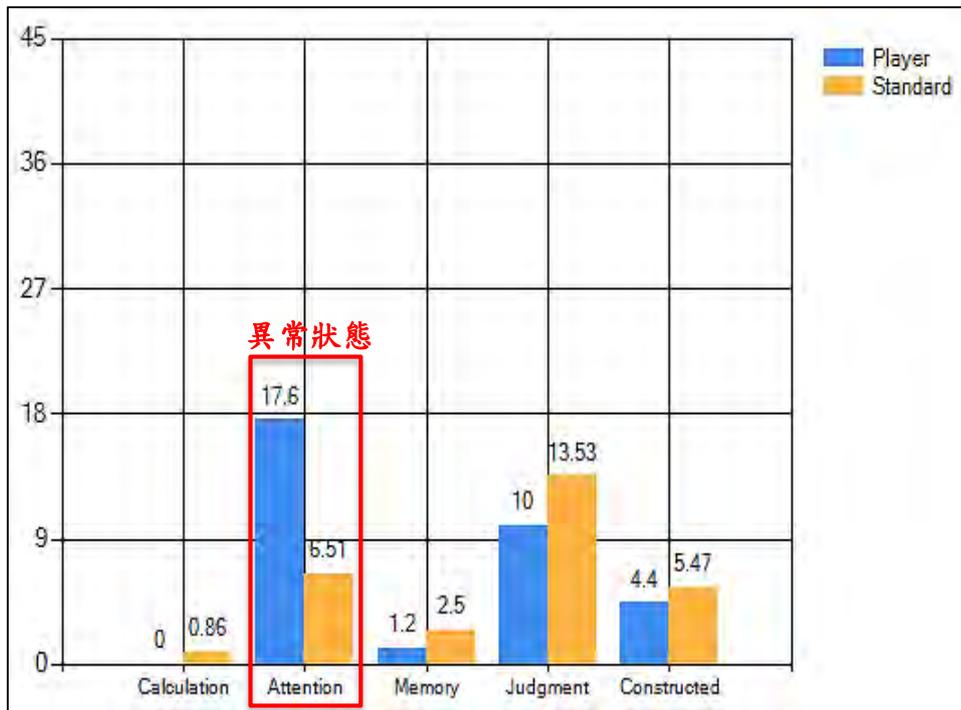


圖 24 狀態數值表

- 二、指標趨勢圖：照護者發現有些指標有所異狀，如藍色線在某天突然降於橘色線以下，便可以立即發現被照護者的狀況已經趨於異常，照護者可以據此給予關懷及協助。

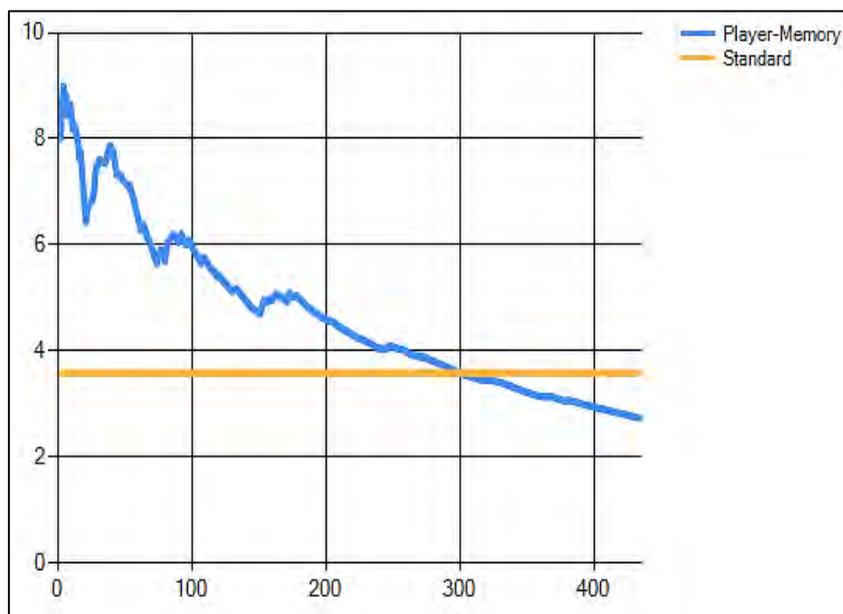


圖 25 指標趨勢圖

三、照護者分析系統：透過登入管理權限，讓照護者可以在這個介面上了解被照護者的整體資訊。

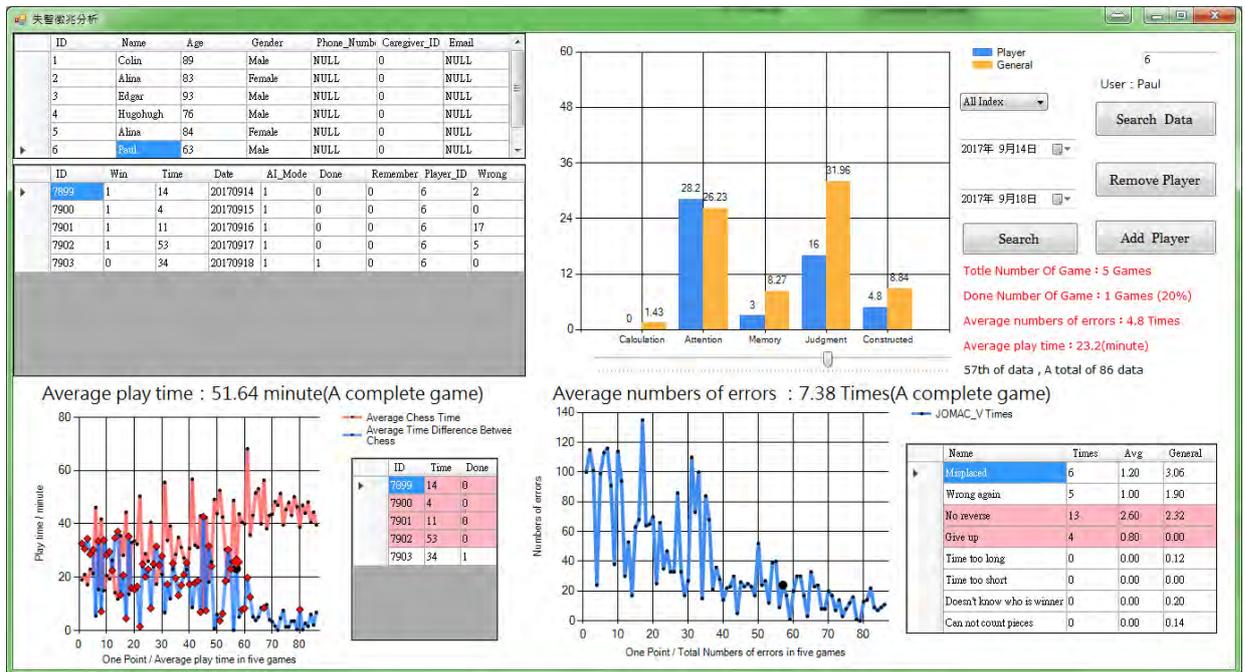


圖 26 照護者分析系統

實驗三：機器手臂測試

實驗一 使用 AI 伺服馬達版的機器手臂，進行 4 點 6 方向的來回夾取測試，以了解機器手臂的穩定性及精確的程度。

實驗說明：為瞭解機器手臂是否可靠，經調整結構穩定、彈力平衡力矩、減輕重量已達最佳化後，進行 4 點間 6 種來回取子、落子動作測試成功的次數，評估機械手臂重現率。

(一) 以玩家視角的位置，分別在棋盤上的四個角落標出四點，左上角為 A 點，以順時針的順序依序為 A→B→C→D。

(二) 每次進行實驗都以兩點反覆落子與取子為一次動作。完整完成視為成功，若中間有動作失敗，則視該次測試為失敗。每一個動作都進行 100 次，利用上方的網路攝影機記錄每一次的動作過程。

動作	第一代 R C 伺服馬達版		第二代 A I 伺服馬達	
	成功次數 (測試 50 次)	成功率	成功次數 (測試 100 次)	成功率
	A<-->B	45	90%	100
B<-->C	46	92%	100	100%
C<-->D	48	96%	100	100%
A<-->D	47	94%	98	98%
A<-->C	43	86%	100	100%
B<-->D	45	90%	100	100%

4 點 6 方向測試路徑規畫	測試結果
----------------	------

圖 27 六組測試的規畫及測試結果

測試結果：

- 1、以最後的實驗結果得知，手臂的重現率達 98% 以上。
- 2、都能夠完成 100 的反覆取動作，而在 A 點與 D 點有兩次取子失敗，原因在於夾子設定的力道不夠大，經由調校之後，都可以完成 100 取子。

討論：

- 1、與 RC 伺服馬達版的機械手臂的取子成功率實驗相比，使用 AI 伺服馬達後整體的穩定度和精度都有大幅的提升，成功率有 98% 以上的結果。
- 2、RC 伺服馬達版有抖動或是過載的現象，使用 AI 伺服馬達來控制，加上整體結構重量的減輕，將原本的機械手臂更加穩定，能夠完成精準的動作。

實驗四：人工智慧運作測試

(一) 觀察黑白棋程式是否會照規則落子，將權值顯示在版面上比對是否正確。

(二) 為了瞭解位置判斷、遊戲樹及 UCT 三種演算法所設計的人工智慧黑白程式與人對弈的獲勝率，邀集 20 位校內同學，包含新手與有玩過經驗的分別針對這三組程式各玩 5 次。

測量結果：

(一) UCT 演算法：這個方法由於透過經驗及勝率來做為落子依據，因此不容易掉入對手所設下的陷阱。以 UCB 公式計算最具探索價值的節點，從下圖左所示共有 0.18、0.68 及 0.25 三組 UCB 值，對畫框顯示最高的是 0.68，也就是最佳探索價值點為 6-3 的位置，因此下一步為右圖中的 6-3 點已被電腦落子，並取下兩個黑子。蒙地卡羅樹是透過隨機採樣的方式進行模擬，因此資料不足時等同於亂下，在此實驗中勝率會較低。

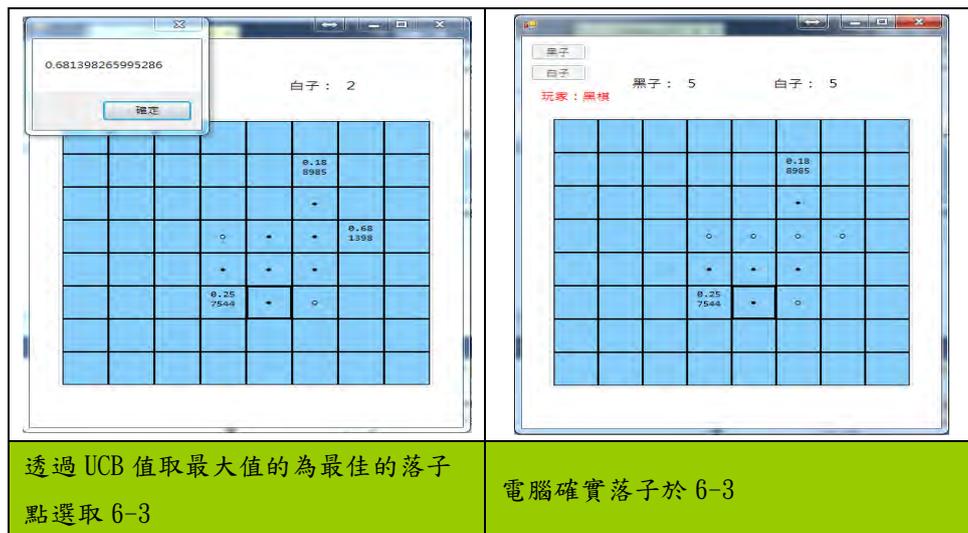


圖 28 觀察採用 UCT 演算法是否會照運算結果落子

(二) 單論翻棋加位置判斷：計算加權值的黑白棋 AI 也會按照所訂定的規則落子，而且它在與新手的對弈中勝率達到 85%，並且會以最安全的位置為落子的點。但也因此十分容易中有黑白棋經驗的人所設的陷阱，所以對於有玩過黑白棋的人勝率只剩下 60%。



圖 29 以位置判斷計算加全值的黑白棋 AI 是否會依照數量來落子

(三) 遊戲樹：透過多回合的分析能夠比對手想得更加深入因此有很高的機率，遊戲樹並不全然按照所訂定的規則走，而且推算到很多步以後的棋，也因此通常執行時間有會稍長，但是新手幾乎贏不了它，即使是有玩過黑白棋的人也時常會輸掉棋局。

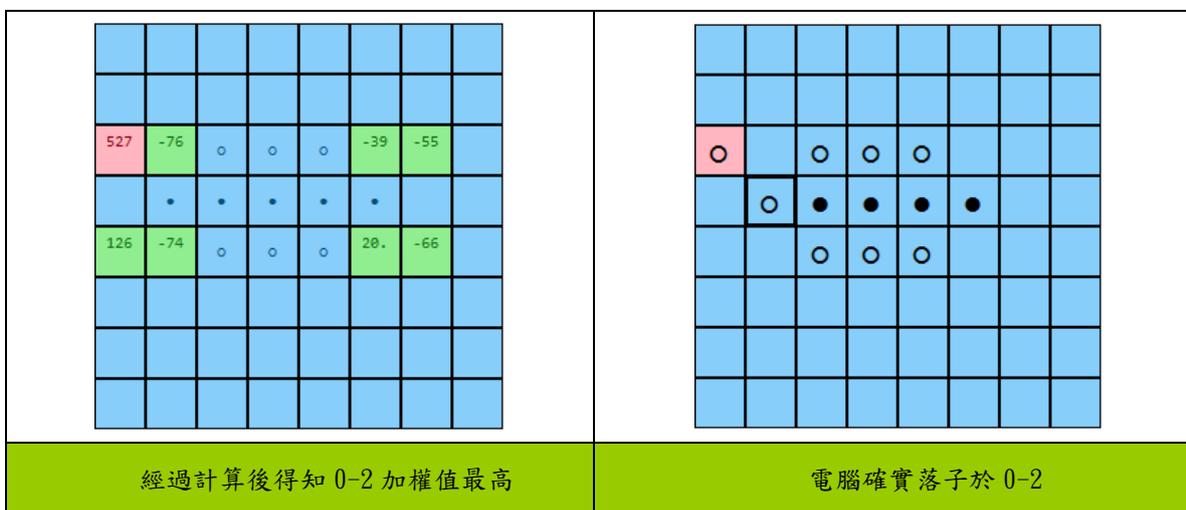


圖 30 以遊戲樹為基礎所撰寫的黑白棋 AI 是否會依照數量來落子

討論：

- (一) 以勝率來看，遊戲樹的勝率最高、UCT 其次、位置判斷最低。
- (二) 速度上在 UCT 模擬輪迴在經過各種優化後已經可以達到 2000 次以上的模擬圈數，不過由於它必須做資料的分析再經由計算才能得到結果，時間上仍是 UCT 最久。
- (三) 遊戲樹層數越高可能性越多，藉剪枝演算法剪去沒有獲勝機會的分支可使速度提升。
- (四) 由表 1 遊戲樹具有最高的勝率，由於 UCT 演算法需要經過長時間的學習才能有較為準確的數據能供演算法進行判斷，因此在資料不足的狀況下 UCT 的棋力不足。在這階段尚無法知道 UCT 是否可以勝過遊戲樹，須待更充分的學習才可以證實。

表四：人與黑白棋進行 100 次對弈的結果統計（20 人各玩 5 次結果）

使用何種方式運算	對手為第一次玩的人勝率	對手為有玩過的人勝率
單論翻棋加位置判斷	85%	60%
以遊戲樹(4 階)為基礎	100%	95%
UCT 演算法	70%	40%

實驗二 讓 UCT 與遊戲樹第四階（簡稱 T4）進行對弈，了解是否可以學習？評估系統是否在充分學習之後，可以勝過 T4。

實驗五：UCT 學習能力實驗

- (一) UCT 從第一步到最後一步的可能性極為龐大，若從頭學到尾學習，需要很多時間。
- (二) 實驗將使用 T4 下完前面的 N 步，以 T4(N) 表示，再將剩餘的棋步採用 UCT 進行探索及學習。藉以限制 UCT 的學習範圍，縮短學習時間。
- (三) 測試了 3 種範圍分別為先以 T4(20)、T4(30)及 T4(40)。為了測試大量棋局，使用 T4(N)+UCT 與 T4 進行了 160 場對弈，觀察其學習效率以選取較適當的學習範圍。
- (四) UCB 公式中的 C 值為 0.6，讓系統地學習慾望適中。

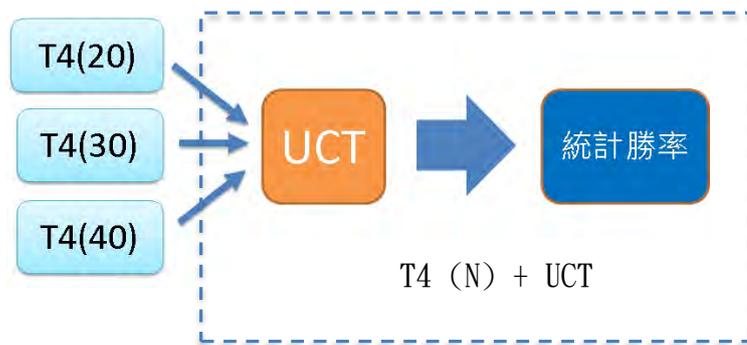


圖 31 評估各種不同組合勝率的實驗流程

實驗結果及討論：

分析 $T4(N)+UCT$, $C=0.6$ 的實驗統計 (如圖 22) 結果說明所示：

- (一) $T4(40)+UCT$ ：UCT 只剩下 20 步，可學習的步數不多，再加上刻意挑選的列式棋局可至勝的步數很少，因此在這個實驗中大約學習 20 場後就已經達到 100% 的勝率了。由於學習前面的佈局均交給遊戲樹處理，剩下 20 步讓 UCT 學習。雖然很快達到 100% 的勝率，但學得太少對於系統的提升來說幫助不大，也無法測試學習的成果。
- (二) $T4(30) +UCT$ ：在 50~60 場左右完成棋局探索並在 80 場後完成學習，達到 100% 的勝率。30 步為棋局中盤左右，學習時間約 70~80 場內就能夠完成學習，而學習的數量比較多，選擇此範圍做為下一個實驗的範圍。
- (三) $T4(20) +UCT$ ：大約 110 場才完成學習，學習時間較久，才可達到 100% 的勝率。剩下的 40 步進行 UCT 的學習，學習時間有點長且範圍有點大因此在這不做採用。

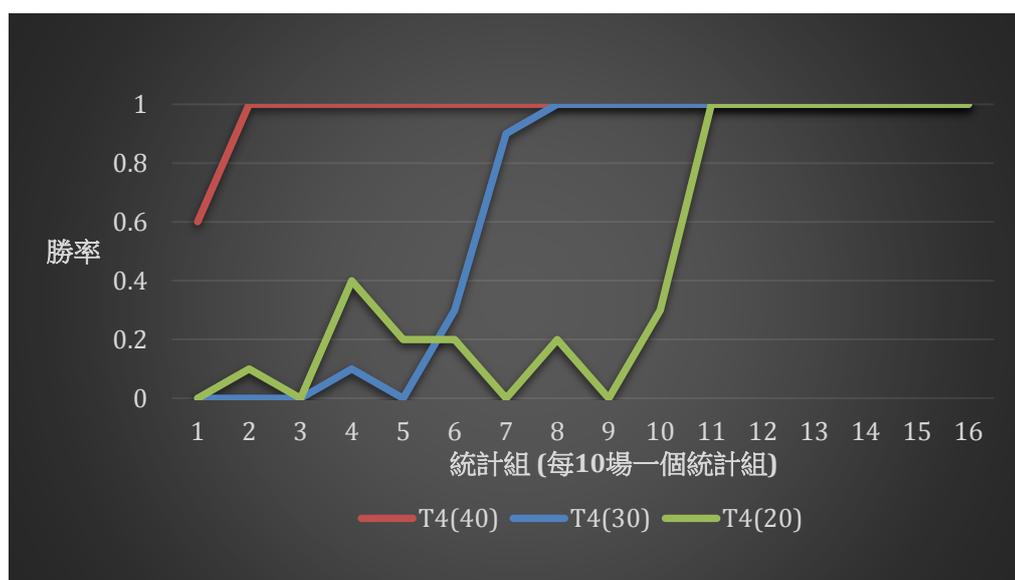


圖 32 使用 $T4(N)+UCT$, $C=0.6$ 的測試結果

實驗六：改變 C 值觀察其學習曲線以找出適當 C 值

- (一) 由於 UCB 公式中的 C 值，會影響系統的探索慾望，影響學習探索的深度與廣度。
- (二) 透過 5 種 C 值分別為 0.2、0.6、0.8、1 及 1.414 與 4 階遊戲樹進行 160 場的對弈。
- (三) 以 T4(30)+UCT 鎖定學習範圍從中局開始，觀察其學習曲線找出最適當的參數值。

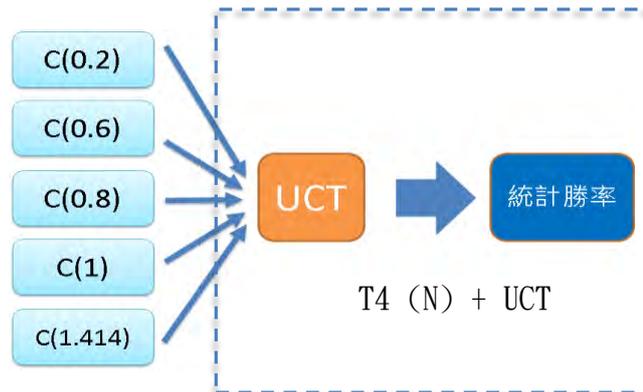


圖 33 評估 T4(30)+UCT 在不同 C 值學習效果的實驗流程

實驗結果及討論：

由圖 24 可比較各種 C 值在 T4(30)+UCT 上的學習效果，結論如下：

- (一) C=0.2：曲線中可以看到它很快得完成學習並達到 100%的勝率，因為他缺乏探索慾望，當發現了勝利的步數後便不會再去探索其他點。以學習的角度來講，探索慾望太低廣度狹窄，可能遺漏許多更好的點，這並不是一個好的範圍。
- (二) C=0.8：相較於 0.6 有更廣的學習寬度，加上學習效率不比 0.6 慢，因此 0.8 比較符合目前的演算法。

(三) $C=1$ & 1.414 ：這兩個範圍在 160 場的棋局中並沒有完成學習的趨勢，只是因為它的探索慾望強大，需要非常長的學習時間，雖然探索上會比其他來的周全但也會因此降低效率，因此目前不採用這個組合。

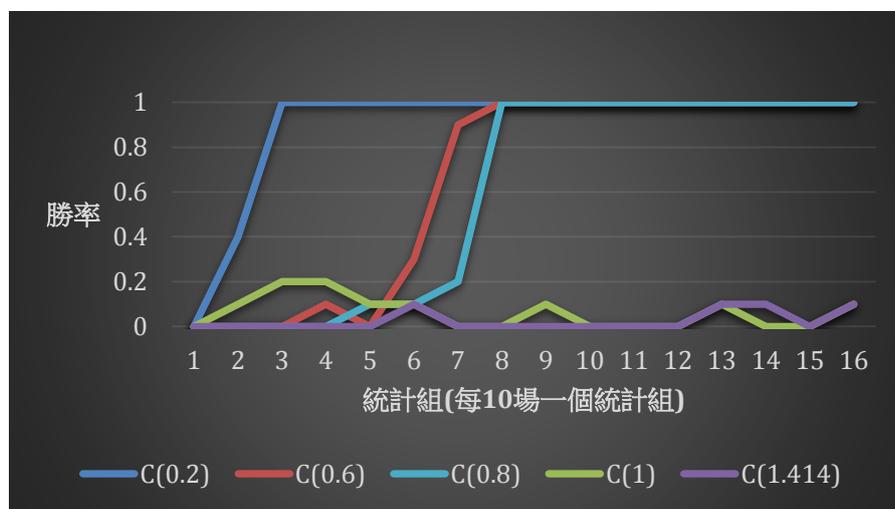


圖 34 評估不同 C 值對學習效率的影響

實驗七： 確認透過影像調整、識別及分析流程，是否可轉為規畫的棋局數位資料。

(一) 將影像透過 HTTP 的協定，從 IPCAM 上進行下載。

(二) 棋盤的四個角落依序點兩下，定出四點再推算出其他 60 點的位置。

(三) 調整彩度及明度之閾值使棋局上落子的分佈，與數位資料輸出結果一致。數位資料定義無子為 0，黑子為 1，白子為 2。

(四) 送出棋局資料，根據 AI 判別進行下一步。紀錄 100 步，觀察影像判斷的穩定度。

測量結果：以下為影像調整、識別及分析的操作流程。



步驟一：輸入 IP、選擇 AI 及玩家棋子並連線（左），等待後影像傳至畫面（右）。



步驟二：四點定位，並推算出其餘 60 點。

步驟三：調整閾值使輸出與棋局影像一致。



步驟四：點擊其按鈕將輸出資料送回到 AI 進行判讀。

之後進行 100 步的對弈，測試影像的穩定度，總共出錯了 2 次，此兩次的原因分別為：

(一) 人工定位的運算誤差。

(二) 閾值非位於可判斷區間之中，導致判斷錯誤發生。

討論：目前影像處理系統的誤差來源其中來自於定位的不精確，以及條閾值的可變性，未來希望能利用霍夫找圓進行定位及利用區間內相同結果的特性來自動條閾值，來減少這類人為誤差提高準確性。

實驗八：測試自動白平衡後，藉由彩度可正確判讀範圍大小，瞭解對色光變化的容忍度。

(一) 將棋盤上的影像攝影下來，並完成定位的程序。

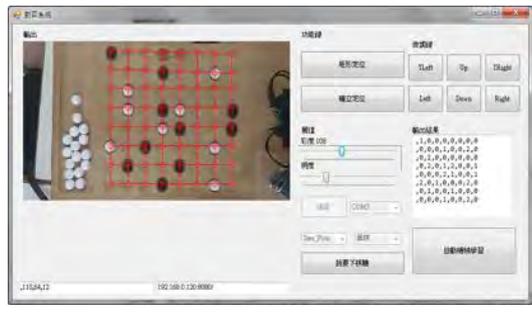
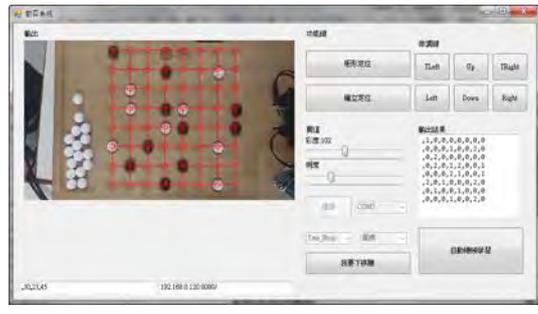
(二) 明度固定調動彩度，使棋局畫面與輸出陣列的結果一致，紀錄最小彩度與最高彩度。

(三) 將原始影像直接判讀（未白平衡），調整可正確判斷範圍的最小彩度及最高彩度。

(四) 把原始影像調整白平衡後進行判讀，調整可正確判斷範圍的最小彩度及最高彩度。

(五) 判別對色光變化的容忍程度：範圍越大，代表容忍度越高，越適合各種環境。

測量結果：

彩度	無白平衡	有白平衡
最低		
最高		
容忍度	108(最高) - 63(最低) = 45	102(最高) - 37(最低) = 65

討論：自動白平衡後確實能減少色光對於影像的影響，並提高可判斷區間之範圍。但自動白平衡並無法提升棋子與棋盤間的對比，只能恢復棋盤與棋子間原本的對比，所以在正常色光比例下提昇寬度的效果很有限。

實驗九：觀察採用直方圖拓寬提高對比，是否可提高白平衡影像對色光變化的容忍度。

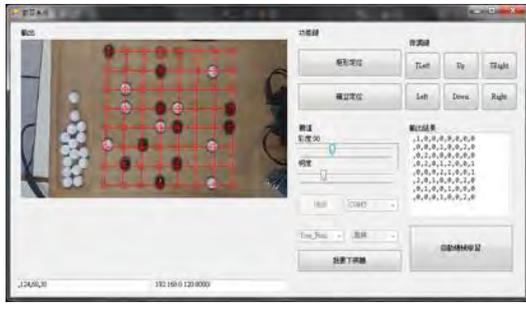
(一) 將棋盤上的影像攝影下來

(二) 將原始影像進行自動白平衡，觀察可判斷範圍的寬度

(三) 將原始影像進行自動白平衡，並加上直方圖拓寬強化，並觀察可判斷範圍之寬度

(四) 兩者數值進行比較來確立直方圖拓寬之效果

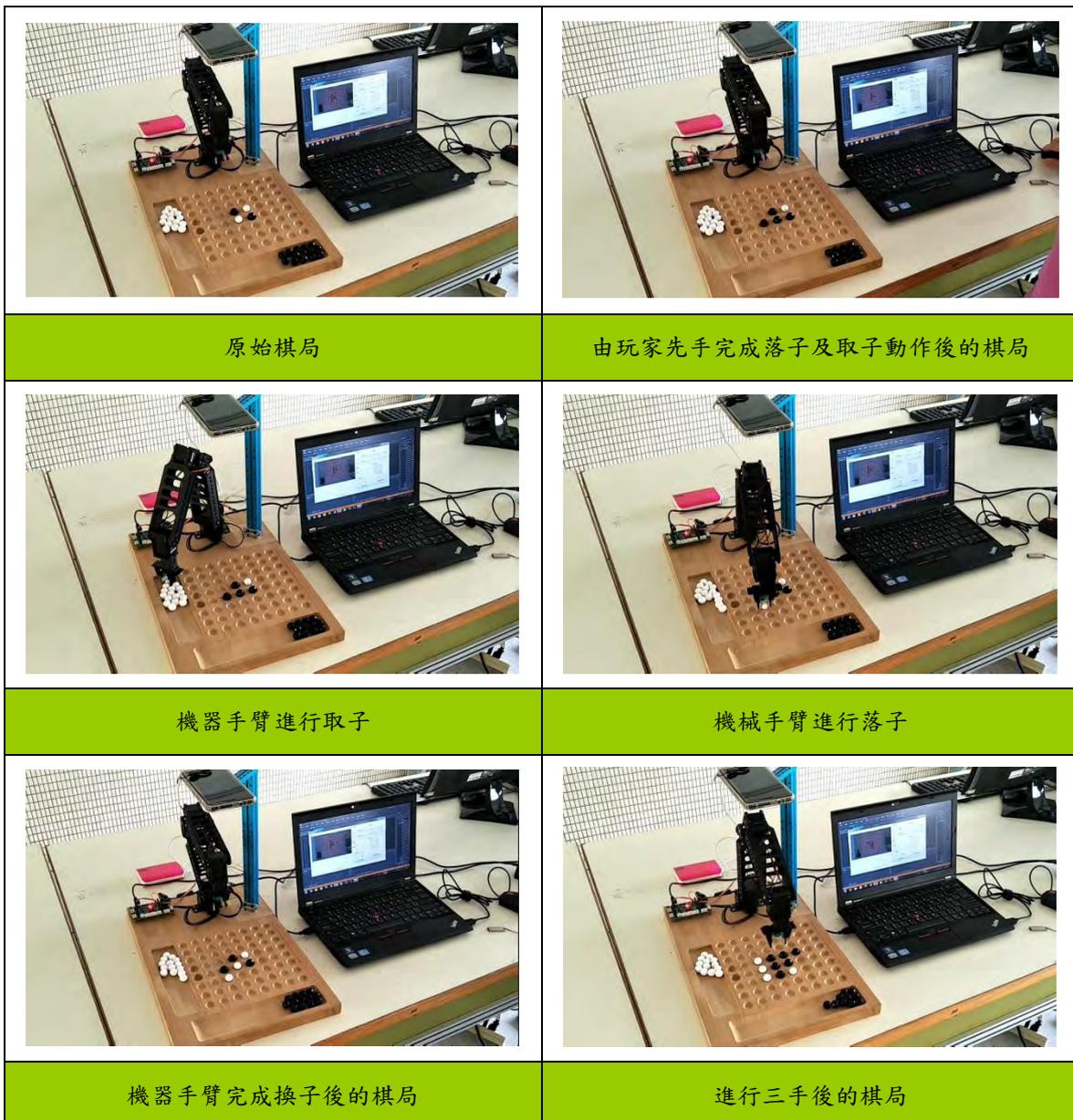
測量結果：

彩度	未直方圖拓寬	有直方圖拓寬
最低		
最高		
容忍度	102(最高) - 37(最低) = 65	90(最高) - 18(最低) = 72

討論：直方圖拓寬確實能使原本棋盤與棋子的對比加寬，但加寬的幅度很有限，此影像強化方法只能用在範圍小的地方，如為內容較廣之區域幾乎看不出強化成果，所以未來打算加上直方圖等化來進行進一步的影像強化，來提升判斷有效範圍。

實驗十：智慧黑白棋系統測試：統合所有系統，進行人機對弈的測試

- (一) 電腦透過 http 協定擷取攝影機影像，使用霍夫找圓以及顏色判斷的處理，將影像轉為數位資料傳送至電腦。
- (二) 將數據經過黑白棋程式的計算，並將計算出來的最佳位置傳送至馬達控制器進行定位。
- (三) 馬達控制器接收程式所傳的指令，以進行機械手臂夾棋和放棋的動作。



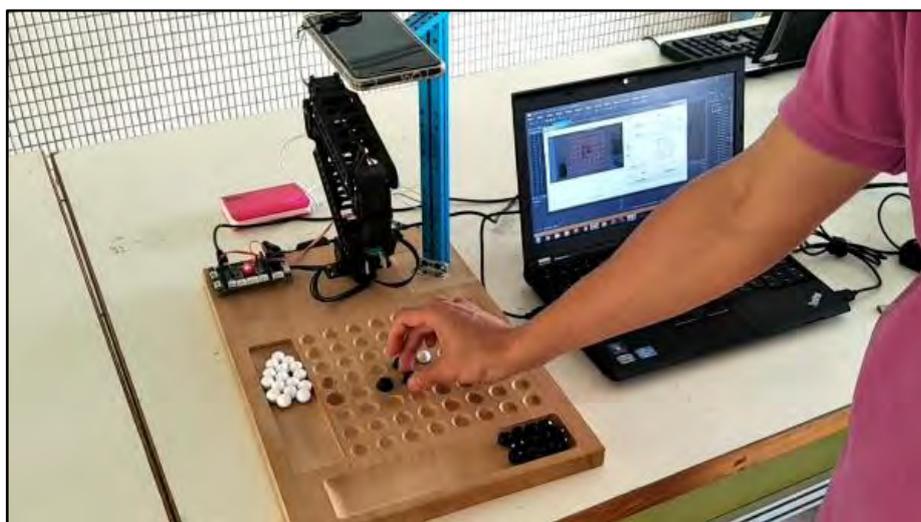


圖 35 人機對弈實況

總結一般使用者 25 次測試結果：

表五 25 位玩家各 5 場對弈之數據統計表

玩家	影像處理正確	手臂運行正確	AI 及手臂運行時間	總時間	AI 勝利	完成完整棋局
玩家一	V	V	16:54	39:21	V	V
			17:42	37:36	V	
	V		18:26	35:12	V	
	V	V	17:21	30:51	V	V
	V	V	19:05	28:32	V	V
平均(五場)	80%	60%	17:54	34:18	100%	60%
玩家二	V	V	17:36	30:35	V	V
	V	V	17:04	32:24		V
	V		19:23	28:42	V	
	V	V	18:42	34:28	V	V
	V	V	19:28	32:03	V	V
平均(五場)	100%	80%	18:27	31:38	80%	80%
玩家三	V		16:32	35:26	V	
		V	18:43	30:16	V	

玩家	影像處理正確	手臂運行正確	AI 及手臂運行時間	總時間	AI 勝利	完成完整棋局
	V		17:36	31:20	V	
	V	V	20:13	34:18	V	V
	V	V	17:25	29:14		V
平均(五場)	80%	60%	18:06	32:07	80%	40%
玩家四	V	V	16:32	25:42		V
	V	V	18:43	26:37		V
	V	V	18:26	30:24	V	V
	V		16:25	32:23		
	V	V	17:53	28:16	V	V
平均(五場)	100%	80%	17:36	28:40	40%	80%
玩家五 (線上玩家)	V	V	17:25	35:24	V	V
	V	V	16:38	32:54	V	V
	V	V	19:13	40:18		V
	V	V	16:25	41:23		V
	V	V	17:31	43:16		V
平均(五場)	100%	100%	17:24	38:39	40%	100%
總平均	92%	60%	17:53	33:04	68%	72%

結果與討論：

- (一) 在本次實驗中，影像的錯誤多半為人為調整閾值的不穩定性，辨別棋盤資訊的閾值是由玩家自行調整，所以就算經由影像強化後，容忍度已經調整的比原先還廣，也會因為人為調整不良而導致判斷錯誤，所以未來打算以自動調整閾值來去減少人為錯失的機率。
- (二) 以上所有手臂運行錯誤皆為取子通道設計不良導致的偶發性卡子，手臂在進行落子及取子時十分穩定，並無任何一次誤夾或誤放，因此需再次設計取子通道以減少運行錯誤。
- (三) 玩家四與線上玩家(玩家五)對於黑白棋有一定研究，因此 AI 勝率較低，在與線上玩家對弈時，發現線上玩家會以特定戰術佈局來提升後期機動性，我方 AI 在最後 3 場皆因

線上玩家已了解我方下法而導致輸棋，且戰術皆為同一種，從此可得知我方 AI 弱點在於盤末機動性掌握並不周全，導致後期遭到逆轉。

(四) 經過今次研究的改良結果，影像已經穩定於 9 成以上，且手臂落子準確度達 100%，在 AI 方面則需將棋局機動性考量在內，避免後期因敵方機動性過高而遭到逆轉。由目前整體完整度可得知，重新設計取子機構為當務之急，若改善此問題，則完成度可達 92%。

實驗十一：失智預防系統實際玩家測試

- (一) 透過與實際年長玩家進行對弈，蒐集一般標準數據外，一併蒐集受試者使用機械人的上手度。
- (二) 觀察受試者的實際狀況、背景及年齡等，與實驗時蒐集到的資訊做比對，觀察兩者間是否有關連。

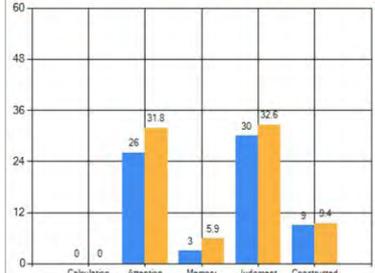
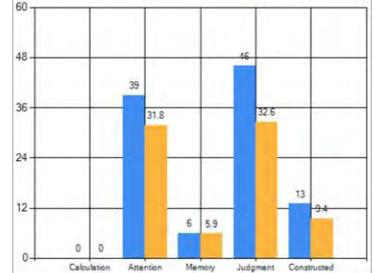
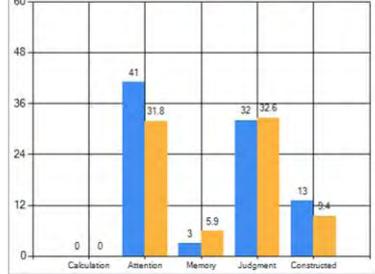
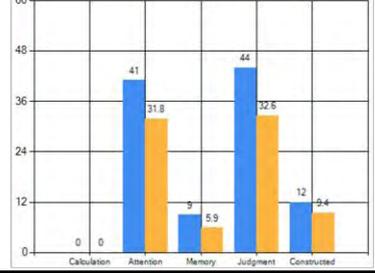
Player	Brain State Chart	Data
		<p>Total Number Of Game : 2 Games Done Number Of Game : 100% Average Wrong Times : 8.5 Times Average Game Time : 52(minute)</p>
		<p>Total Number Of Game : 1 Games Done Number Of Game : 100% Average Wrong Times : 13 Times Average Game Time : 27(minute)</p>
		<p>Total Number Of Game : 1 Games Done Number Of Game : 100% Average Wrong Times : 13 Times Average Game Time : 37(minute)</p>
		<p>Total Number Of Game : 1 Games Done Number Of Game : 100% Average Wrong Times : 12 Times Average Game Time : 35(minute)</p>

圖 36 玩家狀態統計圖

- **玩家 1**：這位玩家以前從沒玩過黑白棋，所以他花了不少時間在思考下一步該怎麼下，但因為他心思比較細膩，所以他比較少下錯，在所有受試者當中，他在狀態表上所顯示的數值是最正常的。
- **玩家 2**：因為該位玩家家裡有急事，所以在對弈過程中下得非常倉促，導致她總是下錯位置。從狀態表上可看出，她的專注力比同年齡的人還要弱。
- **玩家 3**：這位玩家平時下圍棋的經驗。所以在理解黑白棋的規則上比較容易也比較快速。也因此很快就能決定要下的步，但他總是將圍棋的規則與黑白棋的規則搞混，以至於忘記將棋子翻成自己的。所以從狀態表可看出他的專注力較常人弱
- **玩家 4**：這位玩家沒有玩過相關棋類，所以花了較多的時間去熟悉規則，因此經常下錯子，在狀態表裡可看出他的狀態比同年齡的人還弱。

肆、結論與應用

一、結論

本研究改良第一代的黑白棋人機對弈系統，第二代在人工智慧、視覺及機械手臂方面的研究成果如下：

- (一) 透過類神經網路來辨識非線性特徵的影像其成功率在沒嚴重干擾影響下近乎 100%，且學習遠比先前自動校正的速度快上許多。
- (二) 將對弈中的資訊儲存至資料庫加以分析卻時能在資料中觀察出異常狀況，在康寧大學訪查時，分析方向也得到證實。
- (三) 針對 JOMAC 的大腦狀態評估項目蒐集資料、至做檢測小遊戲等方式讓使用者在對弈於樂的同時也能檢察其近似狀況，照護者也可透過手機及網頁隨時看到使用者近況，並在異常時給予關懷及協助。

(四) 對弈中思考對於失智症的延緩本身就有一定功效，在加上對弈系統中的人工智慧會自動分析玩家初始棋力，當玩家棋力上升時會以類似闖關方式的形式提高 AI 棋力。

(五) 透過需求訪談，設計最後的小遊戲，觀察期使用者的計算能力已達 JOMAC 的 C，並透過其他資料分析加上照護者實際判斷狀況能夠有效。

(六) 利用本系統能讓使用者享受對弈樂趣，照護者也可在使用者玩樂之時觀察棋變化，以以往的檢測表為例，失智症患者在接受測量訪問時事會有壓力的，但透過遊玩的方式，在使用者不知情下就可取得需要的資訊以力診斷，也不會讓使用者感到有壓力。

二、討論

在經過多次體驗及實驗下發現以下幾點問題：

(一) 在一開始訓練用的小遊戲太過困難

再經由實地操作後，的確因為多加了正負號的關係變的困難許多，為減少運算時的困難度，可以針對數值產生方式做一些特殊的設計，讓計算式變得較簡易些。

(二) 經過類神經網路學習的影像分析在極度干擾的情況下會誤判

造成誤判的原因基本上是因為除光影極度差異外，還有一些物體干涉鏡頭，才導致錯誤，不然其實在各種環境下、色調下基本上都能將近 100% 的判斷出來。

(三) 機器人下棋時的安全疑慮

AI 馬達有偵測電流流量，藉此或許可以設計碰觸到使用者時，解除動力的設計，避免敲擊到使用者造成對於使用者的傷害。

(四) 為何不在每個棋目上安置感測器而是用影像來進行判別？

如在每個棋目上安置感測器，每副棋盤將安置高達 64 個感測器，且只能用於單一遊戲上，成本較高且自由度也較低，若以影像來進行判別，適用於任何棋類遊戲，不只成本低且自由度也遠比安裝感測器高上許多，因此決定利用影像進行處理判別。

(五) 當使用者忘記規則時的應對處理

本系統有自動偵錯功能，當使用者忘記或下錯時，系統會透過電腦的方式提醒及告訴使用者正確的規則，並透過語音的方式進行教導。

伍、參考文獻

鍾國亮(2012)。影像處理與電腦視覺(第五版)。台北市：台灣東華書局股份有限公司。

陳頌威(2013)。淺談老人失智症。新竹馬偕紀念醫院。

陳閩雄(2003)。一個應用托勒密定理的隨機圓形偵測演算法。國立臺北科技大學資訊工程系碩士論文。

陳俊嶧(2010)。一個蒙地卡羅之電腦圍棋程式之設計。國立交通大學 資訊科學與工程研究所碩士論文

黃士傑(2011)。應用於電腦圍棋之蒙地卡羅樹搜尋法的新啟發式演算法

張任佐、李昀龍(2015)。具有手眼協調能力的高爾夫機器球童。中華民國第 55 屆中小學科學展覽會。

傅中玲(2008)。台灣失智症現況。台灣老年醫學暨老年學雜誌。

取自 <http://www.tagg.org.tw/DOWN/%E9%9B%9C%E8%AA%8C/3-3%E6%9C%9F/3-3-2.pdf>

莊順斌(2005)。以改良式 Sobel 邊緣偵測法與密度分群為核心之產品外觀檢測系統—以車牌辨識為例。國立屏東科技大學碩士論文。

成和平(2004)。百玩不厭的黑白棋《國內第一本黑白棋攻略專書》。宇河文化出版有限公司。

周冠羽(2016)。國立交通大學_電控工程研究所_應用於機器人之基於影像人員活動偵測_論文研讀心得。取自 <http://coolmandiary.blogspot.tw/search/label/EmguCV+%E9%96%8B%E7%99%BC>

王小科、王軍(2015)。王者歸來-C#完全開發範例集。佳魁資訊。

Cameron Browne(2012)。A Survey of Monte Carlo Tree Search Methods。IEEE

【評語】 100010

該作品進行對弈機器人之改良，使其在運用層面有新的應用，針對未來人口老年化與失智症之可能，研究作品搜尋失智症分析之量化指標與大腦功能狀態評估，整合機器人改良研究與黑白棋遊戲應於預防及延緩失智症之發展。建議未來可與大型醫院合作，針對實際失智症之病患予以提供協助並收集可行性評估之數據。