

# **110-1 人工智慧**

## **麻將 AI 設計與實作**

### **小組專案報告**

班級、姓名與學號：

資工二 林奕廷 110590004

資工二 吳宥駒 110590066

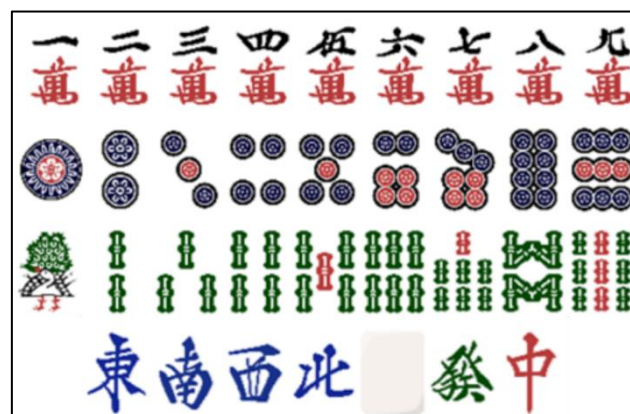
## 一、引言

麻將為一種多回合手牌制的不完美資訊多人遊戲，起源於十九世紀末期，目前在世界上有超過千萬的玩家。作為一種富決策與技術性及機率的遊戲，對於 AI 有許多挑戰性，除了因有四名玩家且牌局隱藏資訊極大，還有非常複雜的遊戲規則，其中多樣的勝利牌面也讓在遊戲過程中的組牌方式具有極大的彈性。

目前在國際間有非常多的麻將種類，我們在此專案中主要針對的是日本的四人麻將，其最主要的特色在於一翻起胡，較於其他較普遍的如台式、中式麻將等零翻起胡的規則更具挑戰及策略性。此外在日本日麻此遊戲也擁有專業的職業競技聯賽。

## 二、麻將規則簡介

日麻中有 34 種牌，包括數字牌萬、筒、索 1 至 9 序數牌及 7 種字牌，每種牌有 4 張總共 136 張牌。



▲麻將牌圖示

遊戲中四名玩家圍繞方形桌四邊圍坐，每人初始有 13 張僅自己可見的手牌。玩家會按照順序從牌堆中取出一張牌，之後打一張牌按順序至於己方的桌面，抽牌者在抽牌後可以選擇是否進行以下幾種行為：

### 1. 自摸：

若玩家的手牌能組成勝利牌型，宣告自摸並將手牌攤開進行計分。

### 2. 立直(Riichi)：

若玩家手牌在打出一張牌後差一張及能組成勝利牌型(聽牌)，可以宣告立直，並放置一根立直棒在桌面上，在這之後的摸牌若不是能使自己自摸的牌，只能直接打出，不能與手牌交換。

在一名玩家打牌後其他玩家可以選擇是否鳴牌，鳴牌分為四種：

1. 吃牌(Chow)：

限定打牌玩家的下一位玩家(下家)，若手牌中的數字牌能與前一位玩家(上家)打出來的牌組成順子(三張數字連續花色相同的牌)，將對方打出的牌拿走並將順子翻開置於桌上。

2. 碰牌(Pon)：

所有除打牌玩家的手牌若能與打出來的牌組成刻子(三張一樣的牌)，將對方打出的牌拿走並刻子翻開置於桌上。

3. 槓牌(Kan)：

所有除打牌玩家的手牌若能與打出來的牌組成槓子(四張一樣的牌)，將對方打出的牌拿走並將槓子翻開置於桌上。

4. 榮和(Ron)：

所有除打牌玩家的手牌若能與打出來的牌組成勝利牌型，可以宣告榮合並將手牌攤開進行計分。

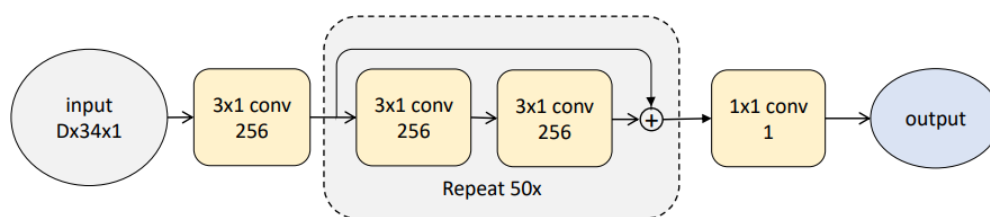
基礎的日麻贏牌牌型為四組面子一組雀頭，面子可以為順子或刻子，雀頭為任意兩張一樣的牌，但日麻其困難之處便在於牌型除了滿足此條件外還要有【役】。如「斷么九」是一種有一翻的役種，限制牌面上不能存在任何的「字牌」或是「一或九的數字牌」，若牌型滿足此條件則可獲得一翻。日麻中約有三十種常見的役種，每一種都有相對應的翻數，翻數越高組成難度也越高，在計算分數時得到的分數也越高，並且多種役種可以疊加計算。若一局遊戲進行到牌池中無剩牌，且無人獲勝，則該局流局。牌局通常總共會有八局，若任何一名玩家分數小於零，遊戲會提早結束，八局結束後統計分數並暫照排名給予段位升降，通常前兩名為勝利，後兩名為失敗，其中最前與最末會有最多的獎勵與懲罰。完整麻將規則可參見(<https://zh.wikipedia.org/zhtw/日本麻将规则>)。

### 三、文獻探討

如今市面上已存在許多基於麻將理論知識建構的專家系統 AI，並且在競技平台上也能到達 PR90 左右的段位；而使用機器學習的 AI 並不多，且表現距離人類頂尖玩家的穩定段位仍有一段距離。

近年有一個微軟開發的 AI Suphx (Super Phoenix) 到達了目前人類無法達到的極高穩定段位。我們深入研究了其論文，其實現主要分為幾大步驟：利用高段位玩家牌譜監督式學習訓練每個行為各自的模型、增強學習、全局預測、先知教練等。

其中監督式學習的模型在各種麻將的機器學習 AI 都有類似的實踐，因此我們認為這是一個設計日麻 AI 的一個很好的起手式。



▲ Suphx 之 CNN 架構示意圖

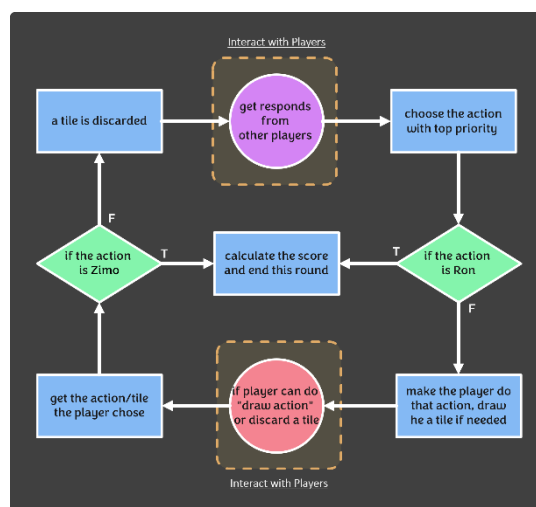
## 四、專案規劃

本專案規劃完成兩種日本麻將 AI，分別為利用貪婪演算法作為行為準則的 AI，以及使用卷積神經網路完成的 AI，並且在完成後對其表現進行分析。專案中需完成的項目主要分為幾部分：

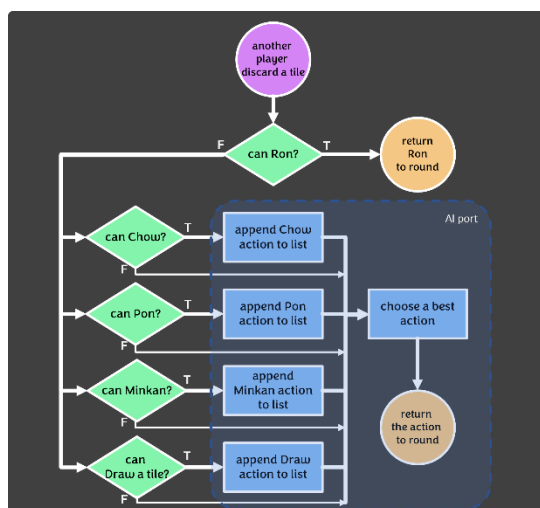
- 本地的日麻環境架設
- 貪婪演算法的設計
- 訓練資料的收集
- 神經網路模型的設計與訓練
- 對 AI 對戰後的結果分析

## 五、專案內容

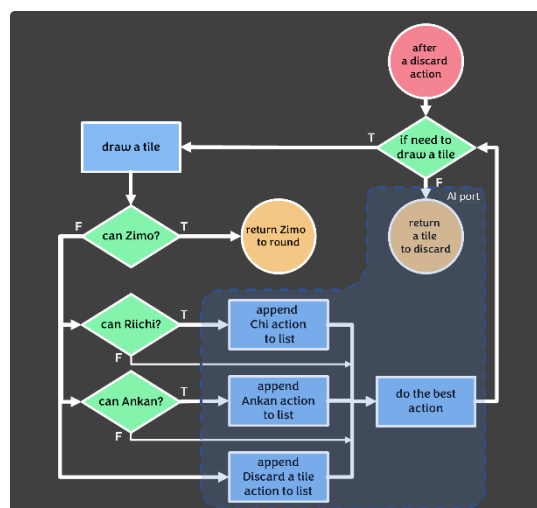
我們使用 Python 建構了一個完整的日本麻將遊戲環境，並基於所有的日麻行為提供接口，讓 AI 能夠對接進行遊戲模擬。此環境運作流程圖如下：



▲ 對局流程圖



▲ 玩家-對他人丟牌之流程圖



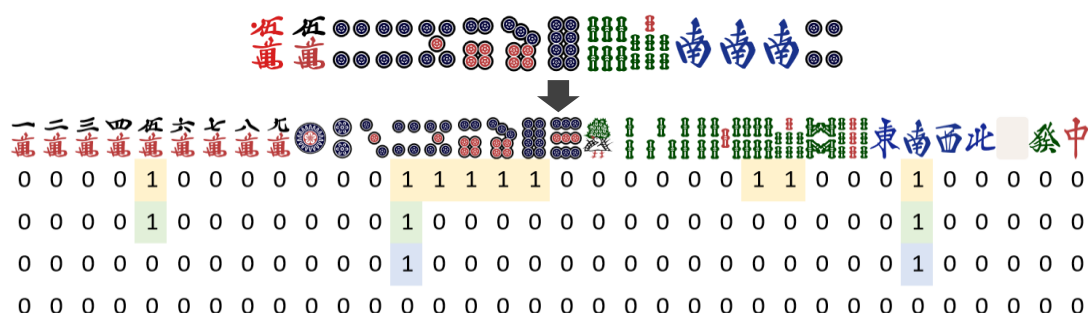
▲ 玩家-輪到自己的流程圖

並設計了一款計算向聽數(還要幾張牌才能聽牌)的演算法，此演算法能夠用於判斷「出哪一張牌」能使當前手牌更接近勝利牌型。由於日本麻將擁有非常多種的勝利牌型，所以對於每一種牌型，必須設計分別的演算法進行計算。

為了取得足夠多且足夠專業的訓練資料，我們選擇從日本擁有最多玩家且歷史最悠久的日麻競技平台【天鳳】爬取對局資料。我們選擇從段位排名前200(PR 0.01)的玩家爬取所有歷史對局，並將資料以 SQLite 資料庫進行整理。

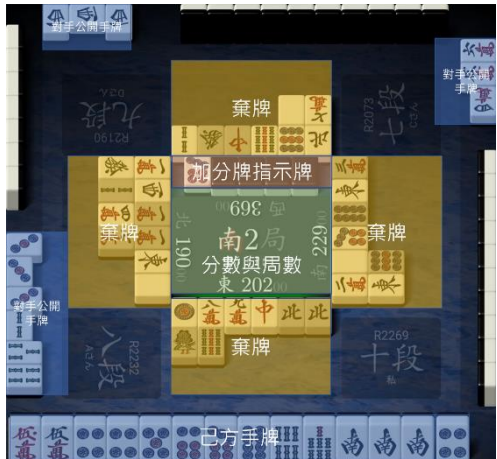
由於麻將不同其他傳統的桌上型競技遊戲，如：西洋棋、圍棋、象棋等具有制式化的場地以及資訊完全的性質，所以要將牌桌大量且複雜的資訊進行正規化是具有挑戰性的事。我們將以以下方法紀錄牌局的資訊，並作為模型的訓練與測試資料。

日本麻將是一種基於「牌」的遊戲，並且牌總共有 34 種，所以我們決定將所有資訊編碼為  $34 \times N$  的格式，例如以下為將手牌編碼的方式：



一~九萬	一~九筒	一~九索	七種字牌
索引值：0~8	索引值：9~17	索引值：18~26	索引值：27~33

玩家也能夠從牌桌取得許多訊息，我們將這些資訊轉換成總計維度  $34 \times 121$  的資料：



▲牌桌顯示的資訊



特徵	維度
手牌	4
己方公開手牌	4
對手公開手牌	$4 \times 3$
己方棄牌	20
對手棄牌	$20 \times 3$
所有公開牌面	4
己方立直狀態	1
對手立直狀態	$1 \times 3$
寶牌	4
己方分數	1
對手分數	$1 \times 3$
場風	1
自風	1
對手自風	$1 \times 3$
<b>總計</b>	<b>121</b>

▲編碼成  $34 \times 121$  格式

在專案中我們基於日麻的四種基本操作建構模型，分別為棄牌、吃、碰、立直，由於「槓」此行為在日麻中是一個多數時弊遠大於利的行為，導致在高段位的對局中此行為出現的次數極少，所以我們並沒有訓練「槓」的模型、爬取資料時也將其資料過濾掉。

行為	10000 局對局資料量		
	總資料量	訓練	測試
棄牌	1,634,987	100,000	10,000
吃	161,479	100,000	10,000
碰	152,111	100,000	10,000
立直	57,217	45,000	12,217

▲爬取的資料比數

考慮到輸出維度的差異，設計出以下模型架構：

	棄牌	碰	吃	立直
輸入維度	34 × 121			
輸出維度	34	2		

▲四種模型輸入輸出維度

Input, 34 × 121
3 × 1 Convolution, 256
Batch Normalization
3 × 1 Convolution, 256
Batch Normalization
3 × 1 Convolution, 256
Batch Normalization
3 × 1 Convolution, 256
Batch Normalization
3 × 1 Convolution, 32
Batch Normalization
Dense, 256
Batch Normalization
Dropout, 0.5
Output, 34

▲打牌的模型

Input, 34 × 121
3 × 1 Convolution, 64
Batch Normalization
3 × 1 Convolution, 256
Batch Normalization
3 × 1 Convolution, 256
Batch Normalization
3 × 1 Convolution, 32
Batch Normalization
Dense, 256
Batch Normalization
Dropout, 0.5
Output, 2

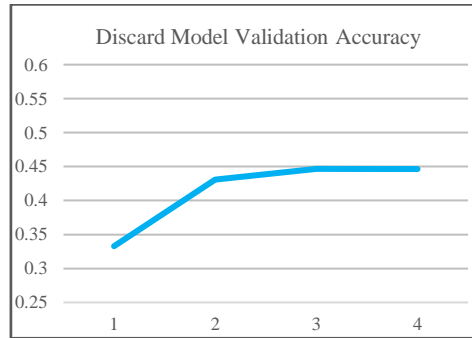
▲吃、碰、立直模型

我們使用 Google Colab 平台進行模型訓練，平台免費提供 Tesla T4 及完整的 Python 線上環境使用。

對於 AI 的成效指標，主要分為牌效、合法預測、胡牌率、放銃率、自摸率、聽牌率、平均排名、流局率等，在 AI 設計完成後使用這些指標進行結果分析。

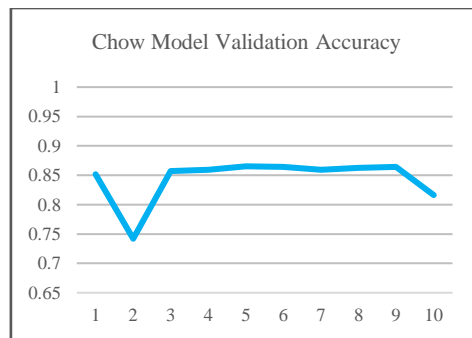
## 六、專案結果

從模型的訓練結果中可以看到，棄牌模型在極少的 epoch 訓練後得到約 45% 的正確率，雖然並不是非常高，但在考慮出牌並非經常有最佳解，並且輸出選項有 34 種的狀況下，這是一個不錯的結果。

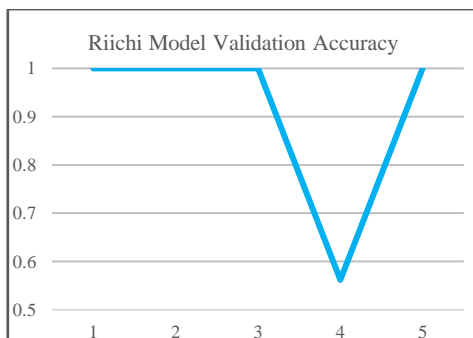


▲棄牌動作的 accuracy

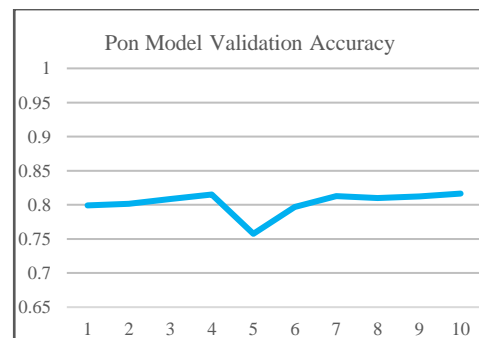
而其餘三個模型在 10 個 epoch 的訓練後都有不低的正確率，但因資料量過少，所以不排除過擬合的可能性。



▲吃動作的 accuracy



▲立直動作的 accuracy

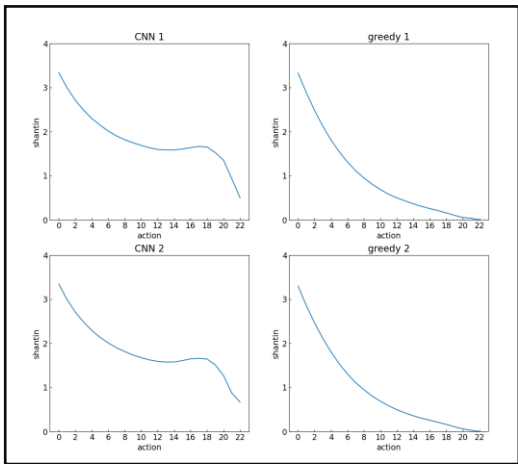


▲碰動作的 accuracy



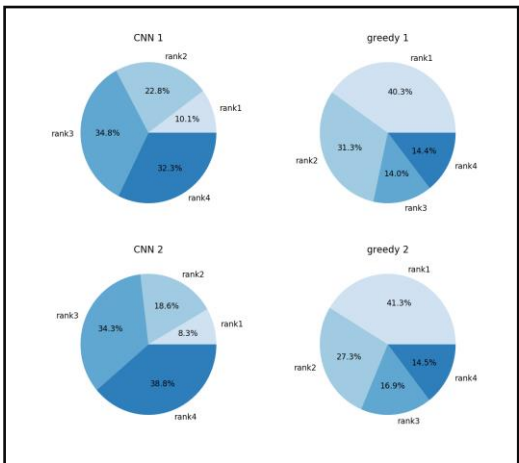
我們讓兩種 AI 各兩名進行遊戲模擬，經過約 1000 場的模擬後，以下為幾個主要能夠進行分析的指標。

表一為兩種 AI 在隨遊戲進行中的向聽數變化之平均，可以看見兩種 AI 在遊戲過程中，向聽數都有降低的趨勢，尤其是貪婪 AI 因算法設計，下降的尤為快速。雖然 CNN 的表現在向聽數接近 1.7 時產生小幅度的反彈，而且多數時後向聽數都大於 1，但向聽數的降低仍代表 CNN AI 具有一定的攻擊性。

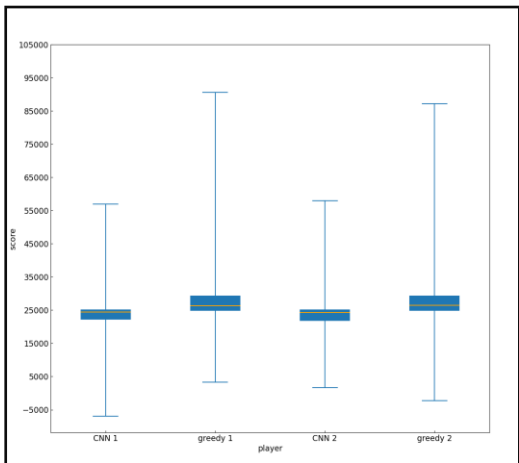


▲表一 每個動作的平均向聽數

在日麻中，排行在一二位的玩家都算勝利，而三四位的玩家則為輸家，從表二能很明顯的看出貪婪 AI 的表現明顯壓制了 CNN AI，但在表三中卻能發現兩者平均分數分布是非常相近的，可以推測兩者實際實力差異並不大。

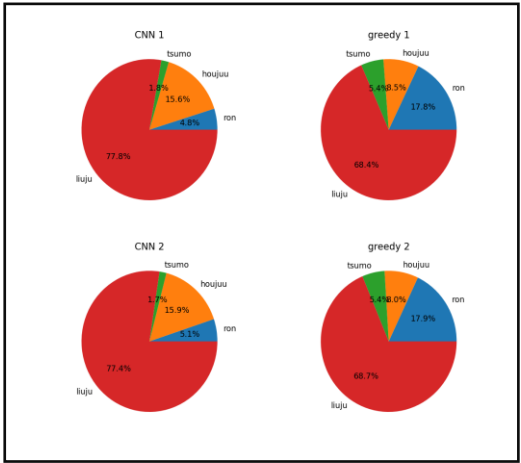


▲表二 名次分布

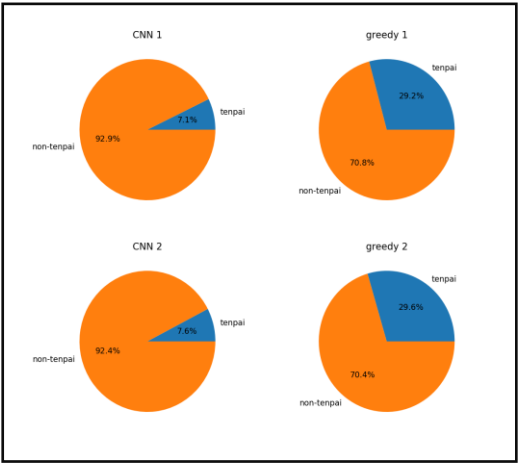


▲表三 分數分布

表四同樣能看到貪婪 AI 在贏牌率與放銃率都是宰制 CNN AI 的，但是值得注意的是在對戰中 AI 的流局率極高，代表 CNN AI 成功對純攻擊導向的貪婪 AI 進行了一定的抵禦。表五能看出的也是貪婪 AI 在進攻能力對於 CNN AI 的高度壓制。

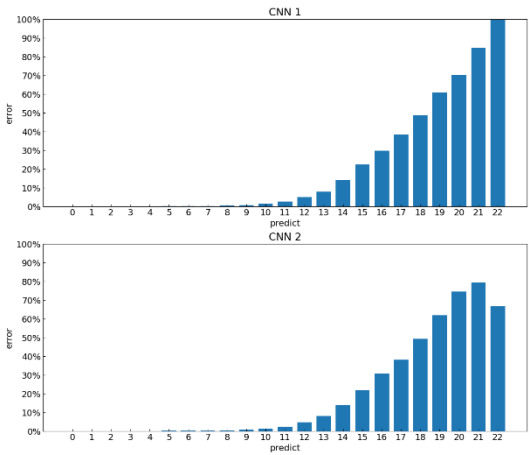


▲表四 狀態統計



▲表五 聽牌統計

表六中能看到作為 CNN AI 的一個重大缺陷，因為對於規則沒有確實的概念，又因為牌局在中後期遊戲資訊越加複雜，此 AI 在牌局中後期容易產生不合法的預測。此外訓練資料的數量也是隨著牌局進行而減少，這樣的資料不平衡也是造成大量錯誤預測的可能原因。



▲表六 模型預測非法率

## 七、結論與展望

在本專案中，我們實作了兩種日麻 AI，一種基於卷積神經網路，另一種基於演算法。以 CNN 架構的 AI 在對局中有表現出對於日麻規則的理解，並展現出一定的進攻性，但同時因對於遊戲概念仍有模糊之處，導致其在末局容易出現非法預測。基於向聽數演算法的 AI 對於規則是清楚的，並且外加算法上的設計，使的此 AI 具有更強的進攻性。

此專案的未來展望主要針對於 AI 的強化，在 CNN 的部分包括加入更多特徵向量在訓練資料並使用更多資料訓練、使用更深的卷積神經網路架構(如 ResNet)、使用強化學習對模型進行加強，以及加入全局遊戲的策略規劃等。在演算法 AI 中可以對防禦機能進行加強，並加入以降低速度換取高分的相關策略。最後讓此兩種 AI 上線與人類玩家對弈，瞭解這兩種 AI 在真實環境下的表現情況。

## 八、參考資料

1. X. Zhao and S. B. Holden, “Building a 3-player Mahjong AI using deep reinforcement learning,” arXiv preprint arXiv:2202.12847, 2022.
2. J. Li, S. Koyamada, Q. Ye, G. Liu, C. Wang, R. Yang, L. Zhao, T. Qin, T.-Y. Liu, and H.-W. Hon, “Suphx: Mastering Mahjong with deep reinforcement learning,” arXiv preprint arXiv:2003.13590, 2020.
3. Shingo Tsunoda. Tenhou Manual. <https://tenhou.net/man>. Retrieved April 22, 2021.
4. S. Gao, F. Okuya, Y. Kawahara, and Y. Tsuruoka, “Supervised learning of imperfect information data in the game of Mahjong via deep convolutional neural networks,” The 23rd Game Programming Workshop of the Information Processing Society of Japan, 2018.