

靜宜大學  
專題報告

# 機械手臂-象棋 對弈機器人

資工四A	411134523	吳珮婷
資工四A	411134484	馬聖宸
資工四A	411134387	陳羿廷

# 目錄

1 動機

2 目的

3 成果演示

4 流程圖

5 象棋

6 機器手臂

7 程式碼流程

8 遇到的問題

車

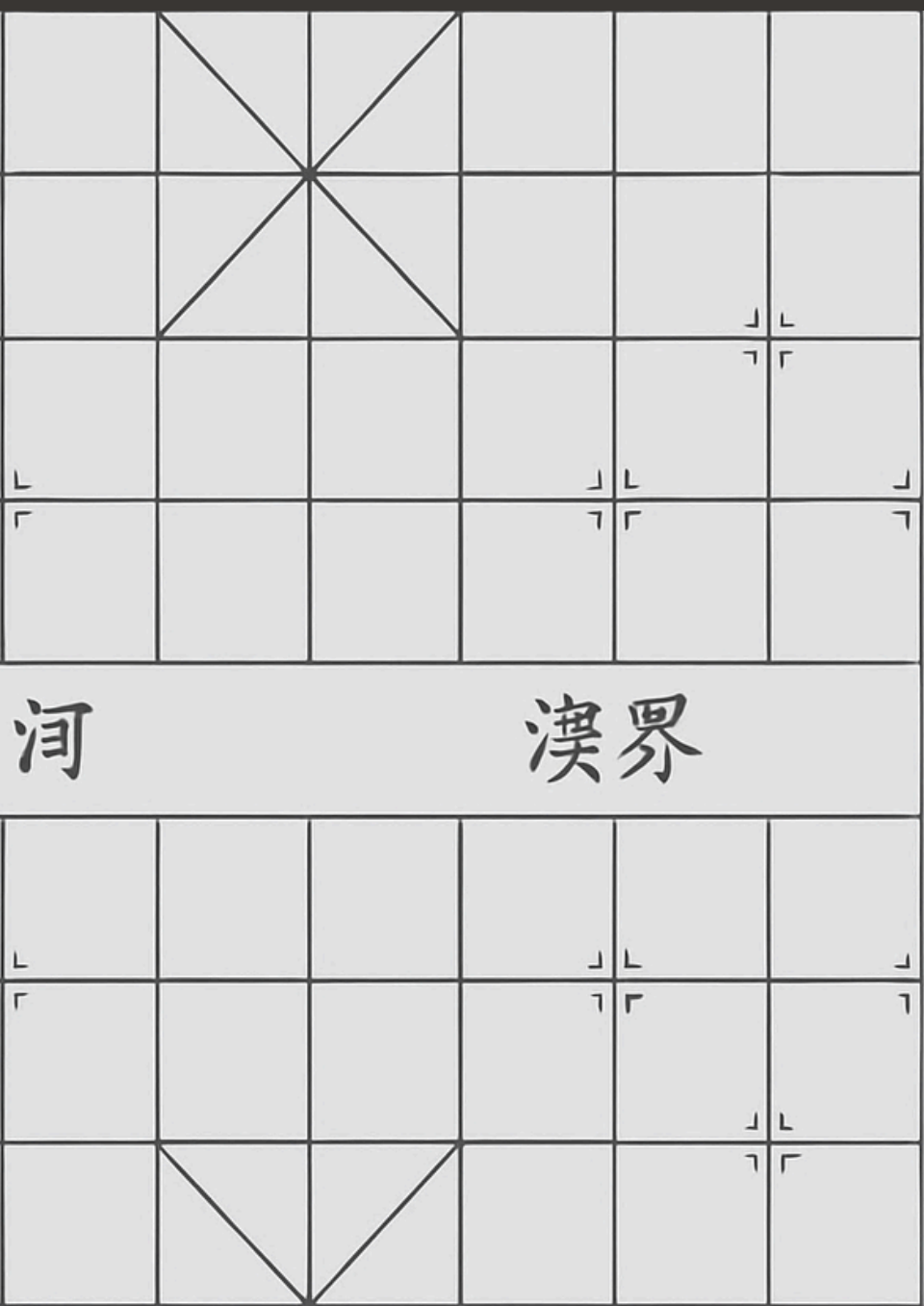
馬

象

士

將





車

馬

相

仕

帥

## 動機

運用人工智慧以及機械手臂的結合，探索智慧機械的應用潛力；提升象棋的互動性，讓使用者可以實際在桌上對弈；推廣Yolo影像辨識，讓人工智能可以辨識的更準確，不僅可以用在辨識人臉，也可以用在棋盤上。

# 目的

讓更多人可以實際的去下棋，而不是只能用線上的軟體進行對弈，也能讓更多對象棋有興趣的人使用，並且作學習。

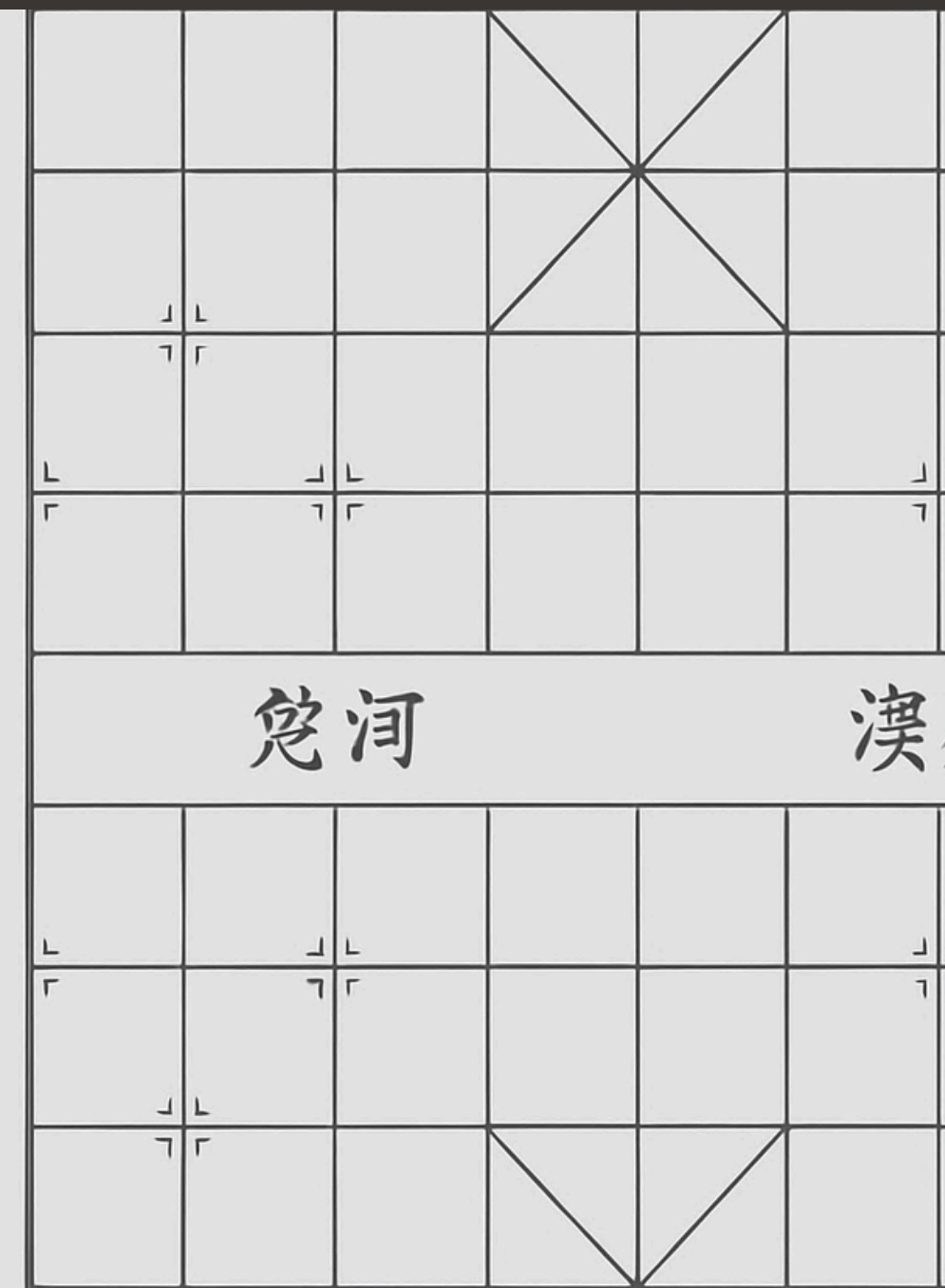
車

馬

象

士

將



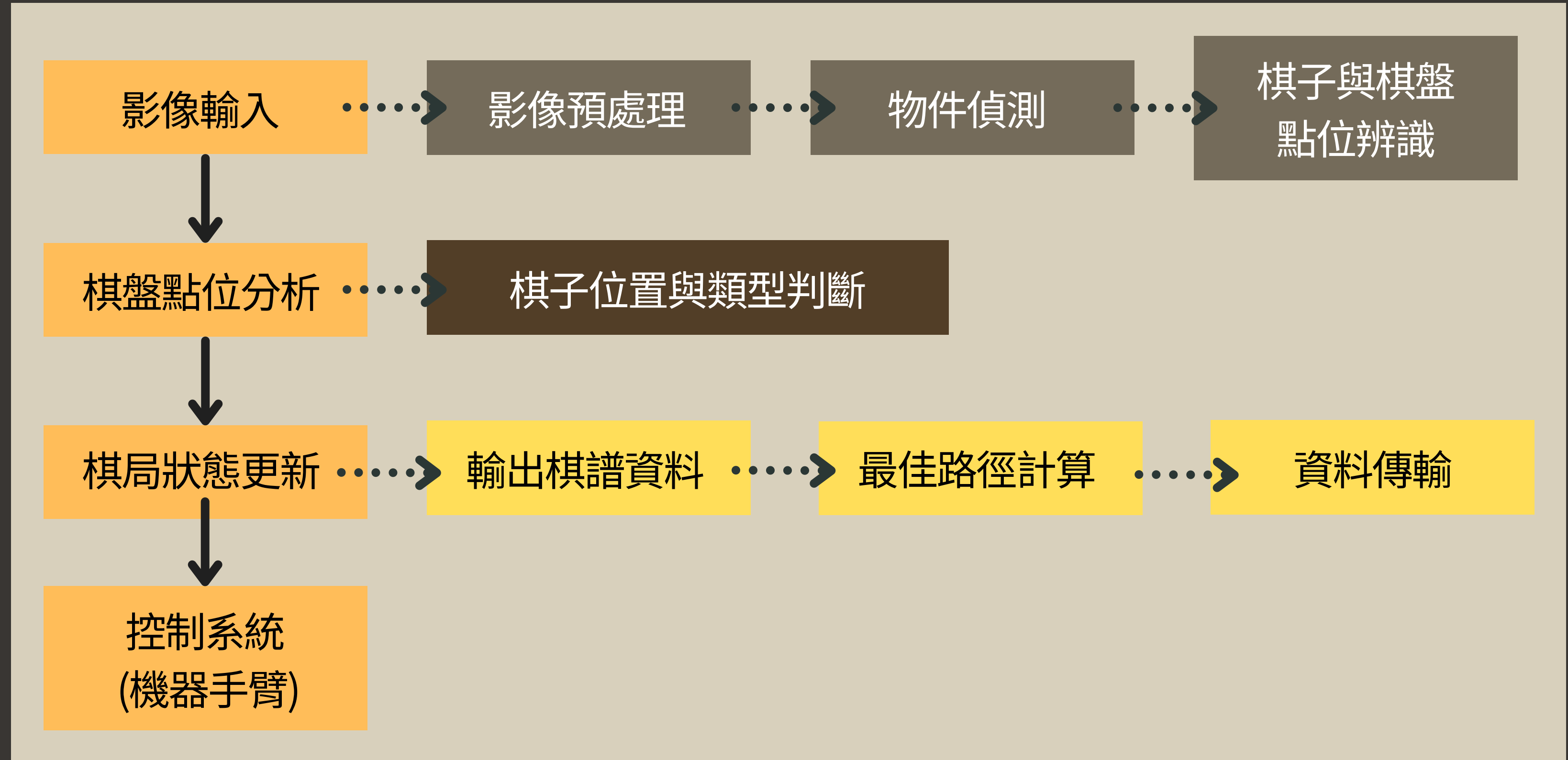
# 成果演示

## 工業機械手臂 ER8-720-MI



### • 影片連結

# 流程圖



# 象棋

→ **OpenCV**

影像處理 (Image Processing)

→ **YOLO**

物件偵測 (Object Detection)

→ **PIKAFISH**

使用核心

# 機器手臂

→ **連接機械手臂**

**ER8-720-MI**

→ **手臂運動軌跡**

平面化移動

→ **記錄點座標**

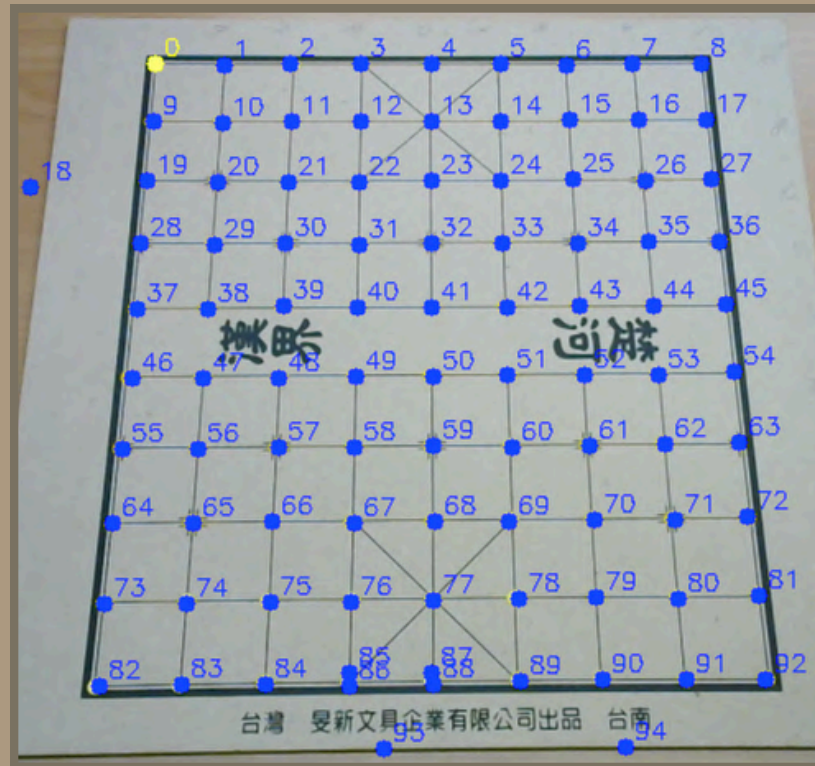
移動座標

# 象棋

漢

## OpenCV

影像處理 (Image Processing)



- 貼黃點，用OpenCV去找

### 棋盤位置想法:

每個點貼黃點再用OpenCV辨識找出座標，再去對每一列做排序，進行微調

### 微調方法:

使用IJKL進行校正、AD切換上下點、X刪除點、N新增點、S結束微調。

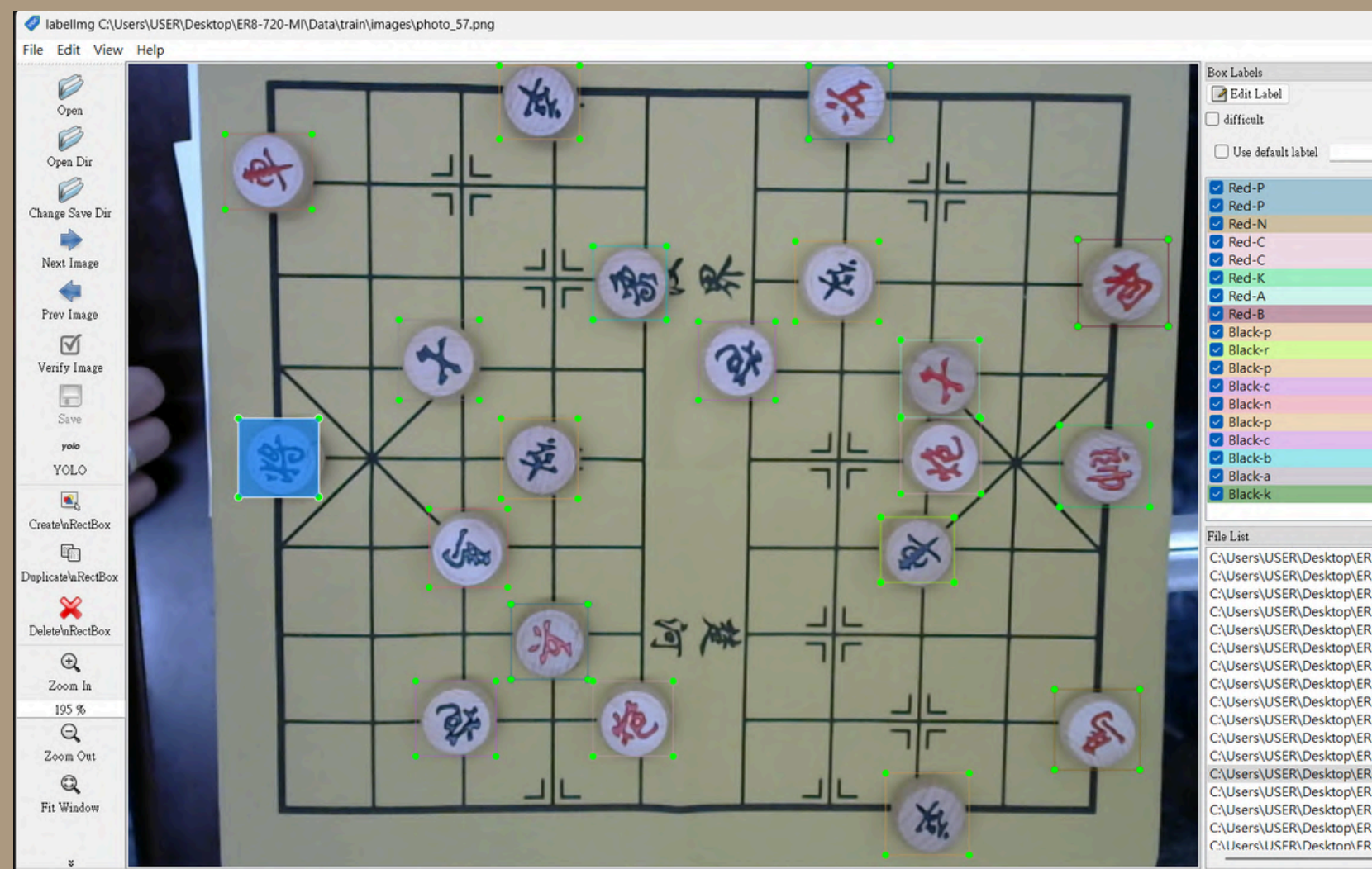
結束後輸出90點可手動紀錄到陣列，並用\*按鍵在微調時取代所有點。

# 象棋

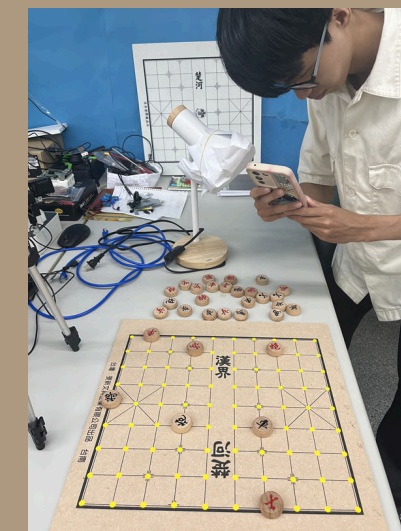
漢

# YOLO

## 物件偵測 (Object Detection)



- 11版(ultralytics)
- 拍照
- 標點(LabelImg)
- 訓練 8次(約600多張)



# 象棋

海

## PIKAFISH

使用核心

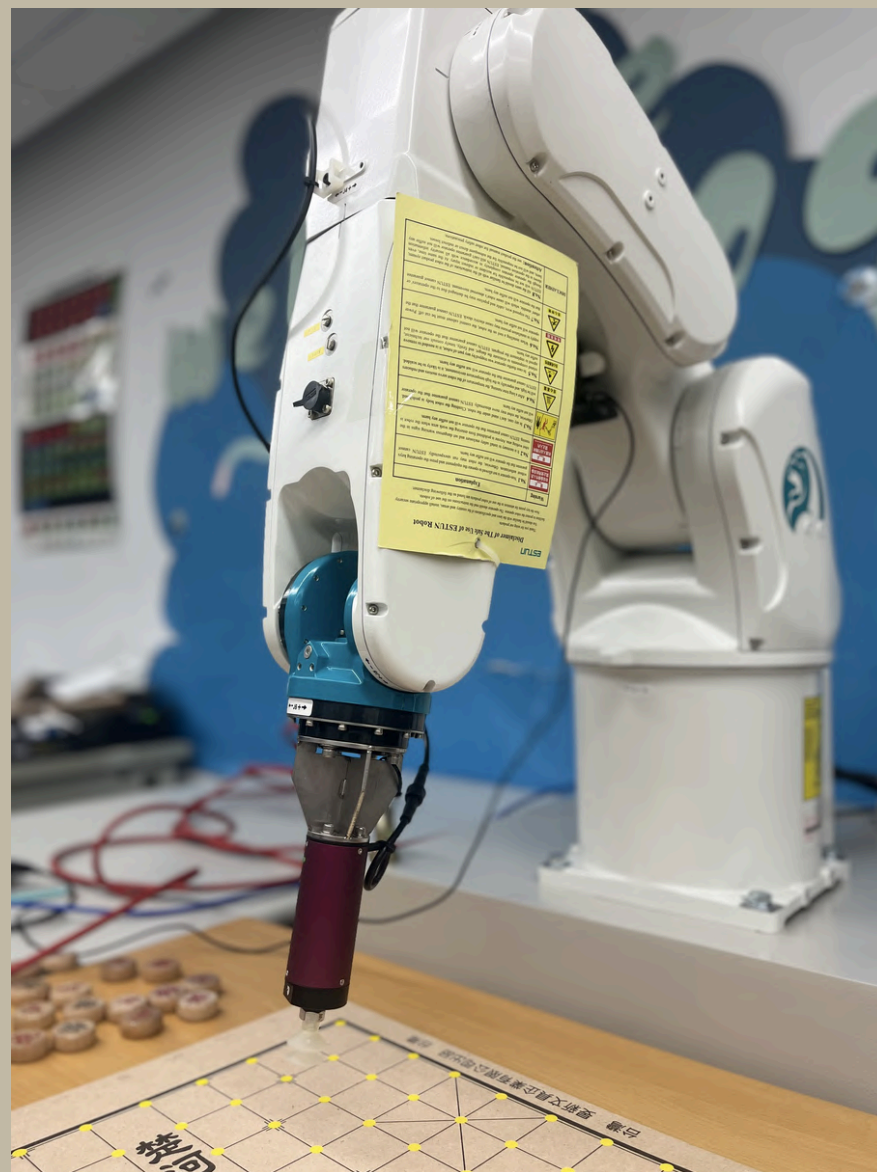


- **Pikafish 是一款專為中國象棋設計的  
開源對弈引擎**
- **Alpha-Beta 剪枝演算法**
- **Iterative Deepening（逐步加深）**
- **Transposition Table（轉換表）**
- **多執行緒處理**

# 機器手臂

澳

工業機械手臂  
ER8-720-MI



- 連接機械手臂
- 手臂運動軌跡
- 記錄點座標

# 機器手臂

## 連接機械手臂



The image shows a network configuration window with two main sections. The first section is for IP configuration, with the option '使用下列的 IP 位址(S):' (Use the following IP address(S)) selected. It contains three input fields: 'IP 位址(I):' (IP address) with the value '192 . 168 . 60 . 65', '子網路遮罩(U):' (Subnet mask) with the value '255 . 255 . 255 . 0', and '預設閘道(D):' (Default gateway) with the value '192 . 168 . 60 . 68'. The second section is for DNS configuration, with the option '使用下列的 DNS 伺服器位址(E):' (Use the following DNS server address(es)) selected. It contains one input field: '慣用 DNS 伺服器(P):' (Preferred DNS server) with the value ' . . . '.

☐ 自動取得 IP 位址(O)

☒ 使用下列的 IP 位址(S):

IP 位址(I): 192 . 168 . 60 . 65

子網路遮罩(U): 255 . 255 . 255 . 0

預設閘道(D): 192 . 168 . 60 . 68

☐ 自動取得 DNS 伺服器位址(B)

☒ 使用下列的 DNS 伺服器位址(E):

慣用 DNS 伺服器(P): . . .

電腦連接網路線，並將IP改65(機械手臂連接IP)，預設閘道68(手臂本身IP)，使用socket套件，將電腦改為host端，手臂為client端。

# 機器手臂

## 手臂運動軌跡



先使用三點標定法將棋盤的方向紀錄於手臂，並設定手臂系統座標為用戶系座標，即可實現手臂在棋盤上平面移動。

# 機器手臂

## 記錄點座標

```
1 {  
2   "RY": [413.65, 371.86, 329.33, 287.83, 247.12, 204.81, 163.46, 121.87, 80.01, 38.17],  
3   "RX": [16.44, 57.88, 99.50, 141.02, 182.52, 224.37, 265.31, 306.81, 347.91],  
4   "RZ": [180],  
5   "Deg": [90, 0, 0],  
6   "outside": [492.62, 221.34],  
7   "initial_user": [131.03, 194.72, 534.43],  
8   "initial_six": [-3.90, 10.48, -4.34, -0.04, 83.26, -1.45]  
9 }
```

改成平面化不再需要設定90個位置，只須設定17個點(行列)、吃子後放的位置、回原位的座標。

在開始執行前，會先使用找點的程式，會跑17個點只須把棋盤對準即可。

# 程式碼流程

1.抓黃點

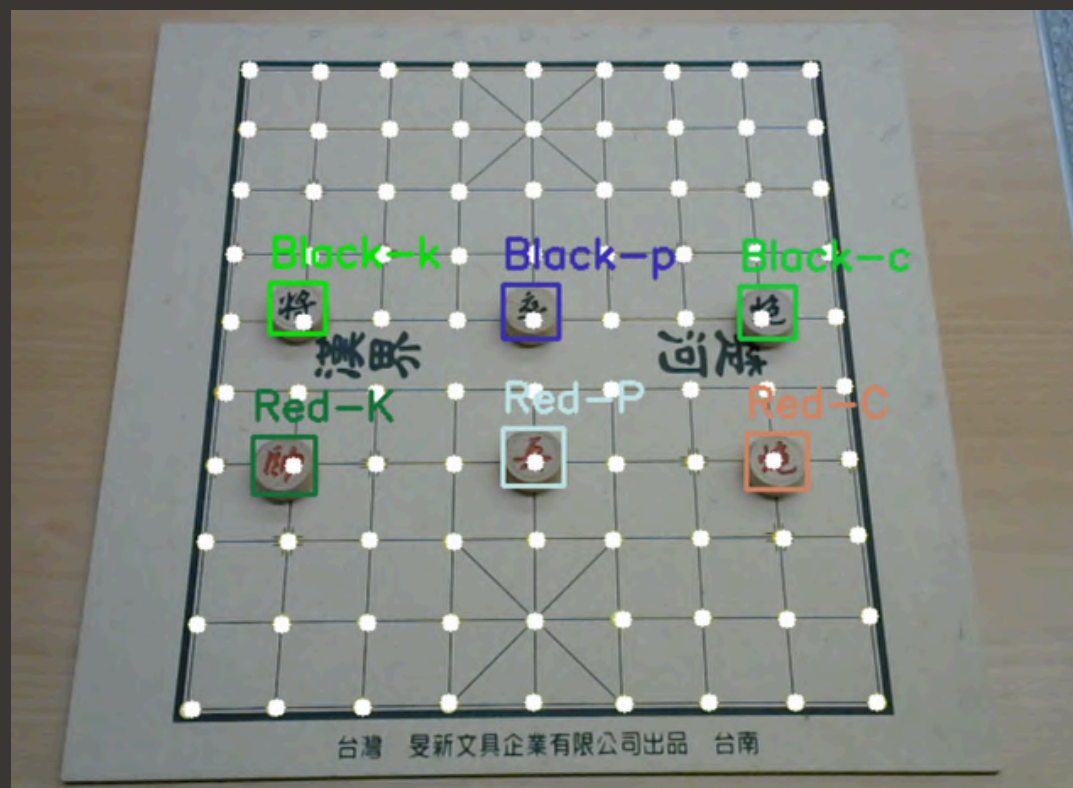
2.微調

3.YOLO辨識

4.轉FEN格式

5.計算最佳解

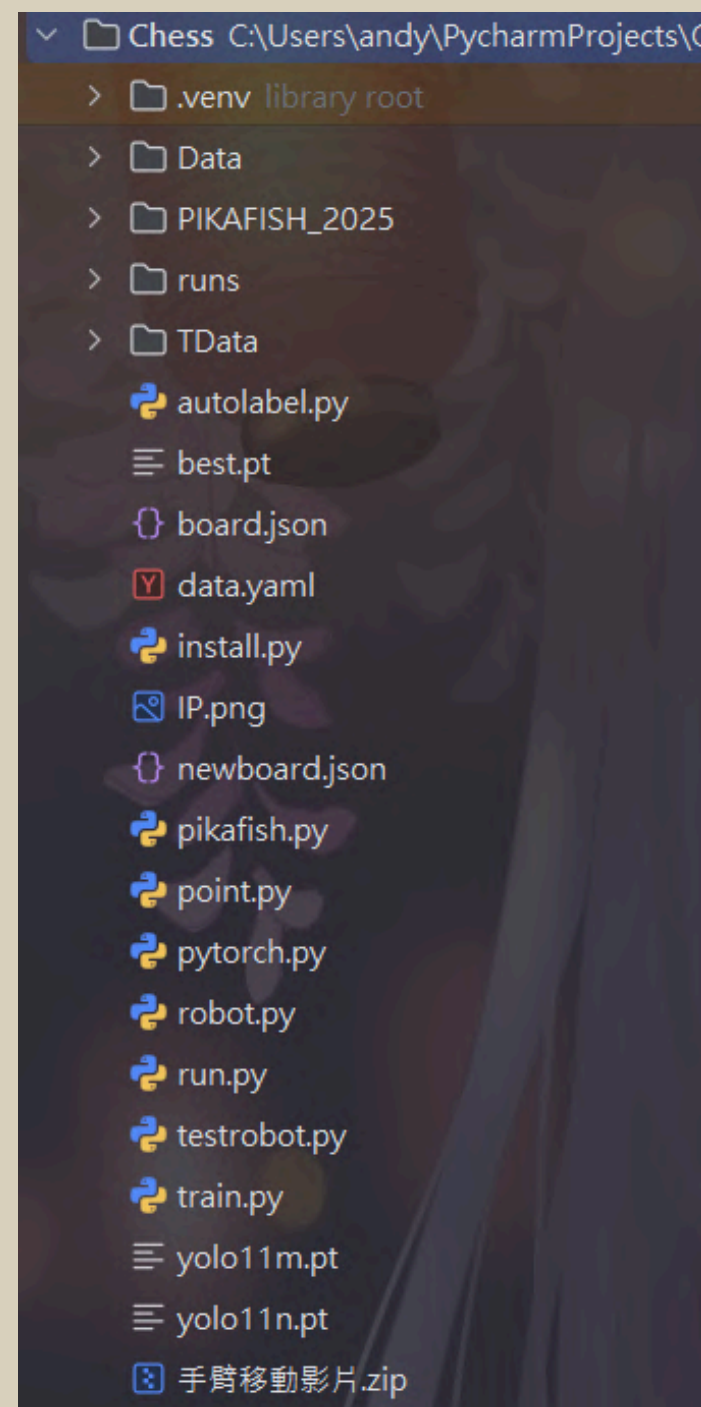
6.傳給機器



Red : (none)  
Black : | e9f9

當前棋盤狀態：

9/9/9/9/1k2p2c1/9/1K2P2C1/9/9/9



autolabel.py 半自動標點

train.py 訓練YOLO

run.py 主程式碼

point.py 算出棋盤位置

pikafish.py 最佳路徑解

newboard.json 手臂座標

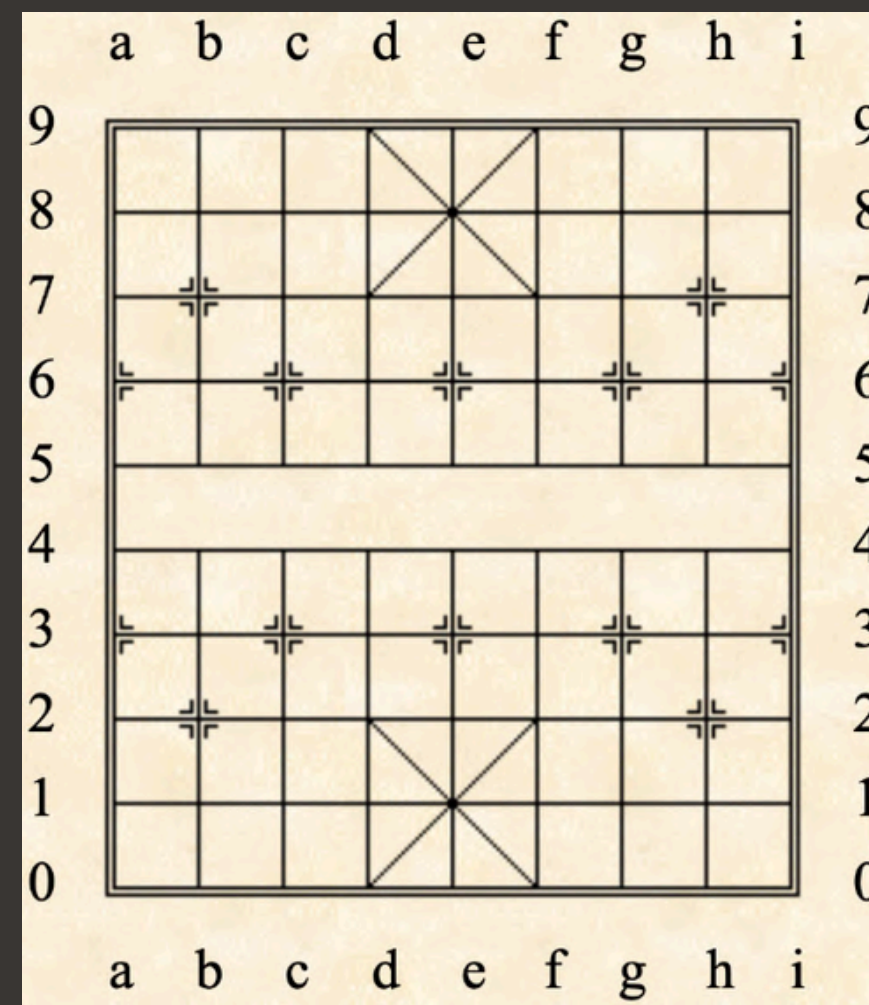


轉成棋譜格式(FEN)  
 讓程式判斷每個棋子位置  
 紅方為大寫，黑方為小寫  
 每列用斜線隔開  
 若是沒棋子  
 則用連續幾個空格數字代表  
 9/9/9/9/9/9/9/9/9



補充

FEN格式



rnbakabnr/9/1c5c1/p1p1p1p1p/9/9/P1P1P1P1P/1C5C1/9/RNBAKABNR

# 遇到的問題

Q1：算棋盤位置時，排序的Y軸不一定一樣會出問題

A1：設定成10以內的都算一條水平線

Q2：手臂無法連接，網路上文獻少，並且吸盤訂製很久、手臂電池又沒電

A2：查看說明說一步一步慢慢試

Q3：吸盤可調節吸力開與關，但是棋子字體有凹洞，會破真空。

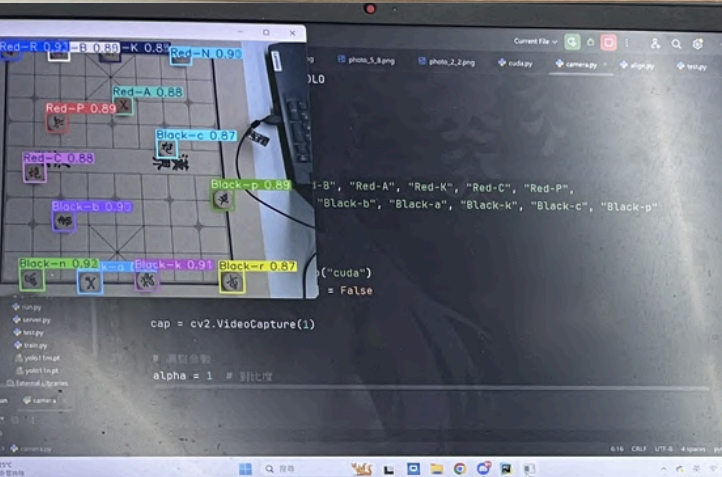
A3：使用膠帶把表面貼起

Q4：吸盤沒有放氣功能，棋子離開吸盤需要一段時間

A4：把吸盤從兩層改為一層

Q5：奇異點問題，手臂走平面時，走的每步都有特定計算方式，手臂軸平行的話會不規則方式大反轉，容易導致機器壞掉。

A5：墊高高度，規避奇異點位置





謝謝大家