

中原大學資訊工程學系

112 學年度 專題實驗書面報告

基於 YOLOV7 的自動射擊機器人

專題組員：

資訊四甲 10927141 黃詳諺

資訊四甲 10927143 王胤迦

資訊四甲 10927144 陳建成

指導教授：

張元翔博士

中華民國一百一十二年十二月

## 摘要

現代射擊遊戲越來越普遍，而遊戲玩家總是想要能在射擊遊戲上面取得更好的表現，隨著科技的演進，利用程式製作遊戲外掛進行射擊，為了達到能夠自動射擊目標的目的，我們首先需要做的事情是蒐集數據，透過螢幕錄影，遊戲畫面後擷取影像，整理所擷取的影像並透過軟體標物件的座標(如圖一)，整理出一個數據集之後便可以透過 YOLOv7 訓練模型，檢測生成的模型的準確度後，便可以透過模型來設計演算法，達成自動射擊的目的性。

本研究採用 YOLOv7 模型來做為主要影像辨識的模型，接著先錄製一段遊戲影片，利用 PotPlayer 其自帶的每幀截圖功能，將一分鐘的影片每秒擷取一張圖片，來收集資料集，將其訓練成用來辨識 AimLab 球體的模型，接著將遊戲畫面截圖送入模型預測，預測結果後計算滑鼠座標，將其拖曳至目標上射擊，最後是希望能訓練一個能辨識任何物件的機器人，將其套用在任何需要滑鼠點擊的程式上。

本研究結果只要將套用的模型稍微修改，就能實現各種滑鼠點擊類型的自動操作。

關鍵字:物件辨識、YOLO、射擊機器人

## **Abstract**

Modern shooting games are becoming increasingly prevalent, and gamers always seek ways to improve their performance in shooting games. With the evolution of technology, using programs to create cheats for shooting becomes a common practice. To achieve the goal of automatic target shooting, the first thing we need to do is collect data. Through screen recording, capture images from the game screen, organize the captured images, and extract object coordinates through software marking (as shown in Figure 1). After compiling a dataset, we can then use the YOLOv7 model for training. After evaluating the accuracy of the generated model, we can design algorithms using the model to achieve the purpose of automatic shooting.

This study adopts the YOLOv7 model as the primary image recognition model. First, record a segment of the game video and use the screenshot function of PotPlayer to capture one image per second from the one-minute video to collect the dataset. Train the model to recognize AimLab spheres. Then, input game screen screenshots into the model for prediction. After obtaining the prediction results, calculate the mouse coordinates and drag them to the target for shooting. The ultimate goal is to train a robot capable of recognizing any object and apply it to any program that requires mouse clicks.

By slightly modifying the applied model, this research demonstrates the feasibility of achieving various types of automatic mouse-click operations.

**Keywords:** Object Recognition, YOLO, Shooting Robot

# 目錄

第一章 緒論.....	1
1.1 研究動機.....	1
1.2 研究目的.....	1
1.3 報告架構.....	1
第二章 文獻討論.....	3
2.1 YOLO 模型介紹 .....	3
第三章 研究方法.....	4
3.1 使用設備.....	4
3.2 影像擷取.....	5
3.3 標記物件.....	7
3.4 操作遊戲.....	9
3.4.1 計算座標.....	9
3.4.2 移動滑鼠.....	9
3.4.3 討論不同的射擊間隔差距.....	10
第四章 實驗結果與分析.....	13
4.1 測試資料比較.....	13
4.2 實際應用於遊戲影片中 .....	14
4.3 將射擊機器人實際應用於遊戲中 .....	15
第五章 結論與未來展望.....	17
參考文獻.....	18

## 表目錄

表 3-1 硬體規格 .....	4
表 3-2 軟體規格 .....	5
表 3-3 測試結果比較表 .....	12

## 圖目錄

圖 3-1 PotPlayer 程式示意圖 .....	6
圖 3-2 遊戲畫面截圖 .....	6
圖 3-3 LabelImg 手動標記物件 .....	7
圖 3-4 YOLO 影像儲存格式 .....	8
圖 3-5 資料集範例圖 .....	8
圖 3-6 AimLab 畫面圖 左上方座標為(0,0) .....	10
圖 3-7 test1 測試結果 .....	10
圖 3-8 test2 測試結果 .....	11
圖 3-9 test3 測試結果 .....	11
圖 4-1 300epoch 資料集訓練結果 .....	13
圖 4-2 要辨識的球體物件 .....	14
圖 4-3 要辨識的物件 .....	15
圖 4-4 Aim Lab 格網射擊遊戲開始畫面 .....	16
圖 4-5 位居網格射擊的排行榜第二名(截錄自 2023/11/23 止) .....	16

# 第一章 緒論

## 1.1 研究動機

隨著不同類型的射擊遊戲的推出，吸引了許多遊戲玩家加入遊玩，但是總會有那麼一群在遊玩過程中受挫的玩家，轉而投向了使用外掛，在短時間內能夠讓自己的遊戲技術提高許多，雖然這些玩家能夠從中獲取了成就感，但是他們所使用的方式是不被遊戲公司所允許的，不僅僅提高了遊戲伺服器崩潰的可能性，也大大影響到其他玩家的遊玩體驗。

正好近幾年 yolo 影像辨識的技術越來越成熟，讓我們想嘗試透過影像辨識，設計出演算法，能夠在不影響伺服器運行的情況下，和使用外掛的玩家制衡。

## 1.2 研究目的

現今在各種遊戲當中使用最頻繁的外掛當屬傳統的遊戲外掛，遊戲在執行過程中會把程式放置到記憶體內，而傳統的外掛正是利用了這個原理，透過從玩家電腦的記憶體中，讀取遊戲中敵人位置，但是這種方法不僅容易被遊戲公司所偵測出，在撈記憶體資料也困難了許多，並且隨著每一次的遊戲載入記憶體，記憶體位置可能會有所變動而必須重新計算偏移量；同時，這樣的行為是不被遊戲公司所允許的，多數的遊戲公司會對頻繁的讀取記憶體的行為給與警告，屢犯的話就會直接封禁玩家的遊戲帳號。

因此，利用模型檢測的方式並實現相對會更容易，也不需要更新資料，並且相對較難被系統偵測出來，取而代之的可能是對於硬體設備上的要求，在載入模型偵測的過程當中會使用到大量的資源，設備如果過於老舊，可能在影像辨識的速度上就會落於他人，也就達不到迅速擊倒敵人的效果。

## 1.3 報告架構

本篇論文共有六個章節，以下條列各章節，並加以說明與介紹。

第一章 緒論：介紹本文的研究動機、研究目的和論文架構

第二章 文獻回顧與相關知識：介紹 YOLO 模型

第三章 研究方法：說明本篇論文資料準備與辨識流程

第四章 實驗結果與討論：對本文物件辨識結果、實驗結果作分析並加以討論

第五章 結論與未來展望：對本文物件辨識模型做結論，並提出可能優化方法或其他可延伸方面之討論

第六章 參考文獻：列舉本篇論文之研究期間，所參考的相關文獻及網站。



## 第二章 文獻討論

### 2.1 YOLO 模型介紹

YOLO (You Only Look Once) 是一種實時目標檢測算法，其主要特點是能夠在一次前向傳遞中同時檢測圖像中的多個目標。這使得 YOLO 在處理實時應用場景時具有較高的速度和效率。

以下是 YOLO 模型的一些關鍵特點和步驟：

1. **多尺度檢測：** YOLO 將輸入圖像劃分為一個固定的網格，每個網格單元負責檢測該單元內的目標。這使得 YOLO 可以處理不同尺度的目標，而不需要多尺度的圖像金字塔。
2. **單次前向傳遞：** YOLO 的主要優勢在於它只需要一次前向傳遞就能夠完成整個檢測過程。相比之下，一些傳統的目標檢測方法可能需要多個階段或多次前向傳遞，導致速度較慢。
3. **輸出格式：** YOLO 輸出的是一個包含目標邊界框及其對應類別概率的張量。每個邊界框通常由四個坐標（左上角和右下角的坐標），一個目標置信度（該邊界框包含物體的概率）和各個類別的概率。

## 第三章 研究方法

本研究目的是設計一個遊戲用途的自動射擊機器人，因此要先訓練該遊戲適用的模型，再來為了要預測，必須先將圖片擷取送入模型中，接著要計算座標，讓滑鼠移動到目標上射擊。

### 3.1 使用設備

研究過程中，由於經費的問題，是透過筆記型電腦來進行實驗過，雖然在效能上略輸桌上型電腦的設備，仍是足以支撐整項實驗；而整個實驗流程是透過 Anaconda 在本機架設虛擬環境，並使用到了 YOLO 團隊在 GitHub 上供開發者使用的 YOLOv7，並使用了 Pytorch 框架安裝 CUDA、Torchvision，需要特別注意的是，安裝的軟體彼此之間存在著相容性的問題，在開始實作之前仍須檢查，以免在過程中需要經歷卸載、重新安裝的過程，以下是我們所使用的硬體、軟體的規格。

作業系統	Windows 11 家用版--22H2
中央處理器 CPU	AMD Ryzen 9 5900HX with Radeon Graphics 3.30 GHz
圖形處理器 GPU	NVIDIA GeForce RTX 3060 Laptop GPU
記憶體 RAM	16 GB DDR4

表 3-1 硬體規格

Anaconda 版本	conda 22.9.0
開發環境	Visual Studio Code
Python 版本	Python 3.9.17
PyTorch	1.11.0
Torchvision	0.12.0
CUDA	11.3
OpenCV-python	4.8.0

表 3-2 軟體規格

## 3.2 影像擷取

首先必須先處理資料集的來源，由於在 AimLab 中每種模式執行的時間皆為 60 秒，而我們透過錄影的方式可以取得一段影像，藉由「PotPlayer」軟體的幫助，可以選擇每間隔固定秒數或是每一幀，擷取影片的畫面，本研究透過秒為單位，使用 PotPlayer 分割影片中的影像，也就是圖 3-1 的示意圖，可以從影片中得到大約 60 張的遊戲畫面，每張畫面裡會有三個球體，正是三個所需偵測的物件，如圖 3-2。



圖 3-1 PotPlayer 程式示意圖

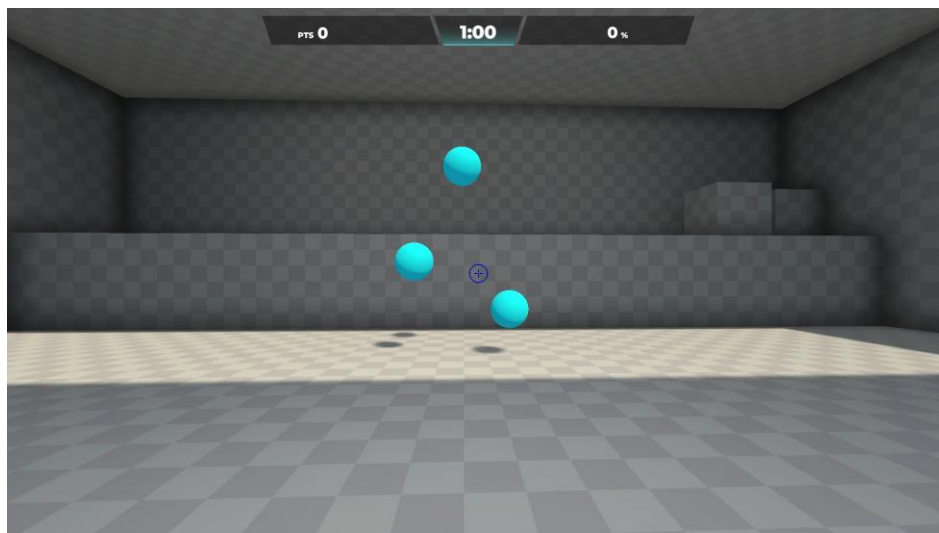


圖 3-2 遊戲畫面截圖

### 3.3 標記物件

本步驟算是整個研究過程中最為繁複的，在完成 3.2 章節的影像擷取後，需要透過「LabelImg」軟體，將所偵測到的三個物件，手動標記其所在位置，如圖 3-3 所示，並以 YOLO 的影像格式儲存，而格式會如同圖 3-4 所示，其中欄位分別代表的意義是物件的種類編號、中心點 X 座標(經標準化)、中心點 Y 座標(經標準化)、方框寬度(經標準化)、方框高度(經標準化)，由於所辨識的物件只有球體，所以物件的種類編號只會是 0，代表球的意思；最後將所有的切格圖片檔以及儲存的格式檔存放到資料夾中，這樣所需的資料集便搜集完成。

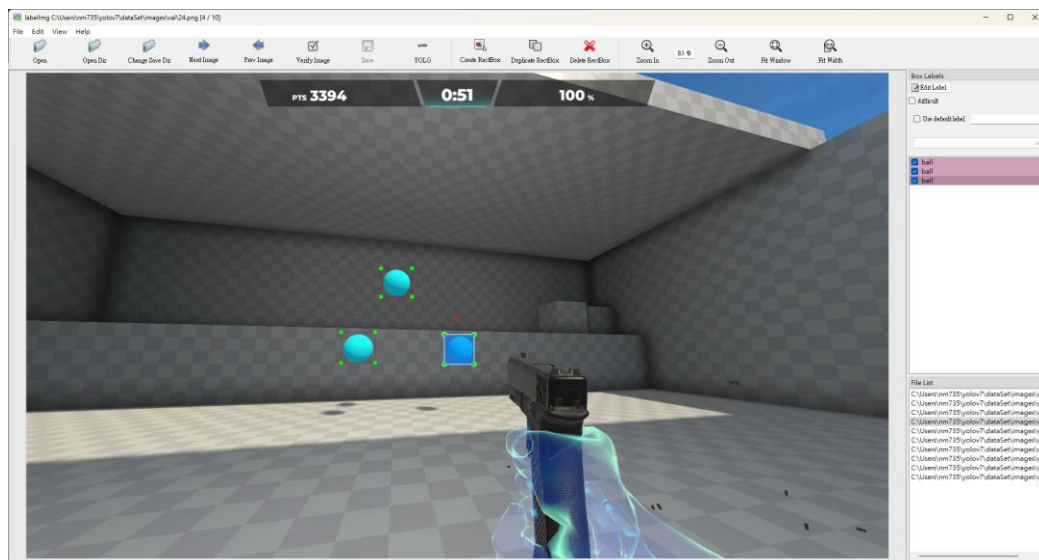


圖 3-3 LabelImg 手動標記物件

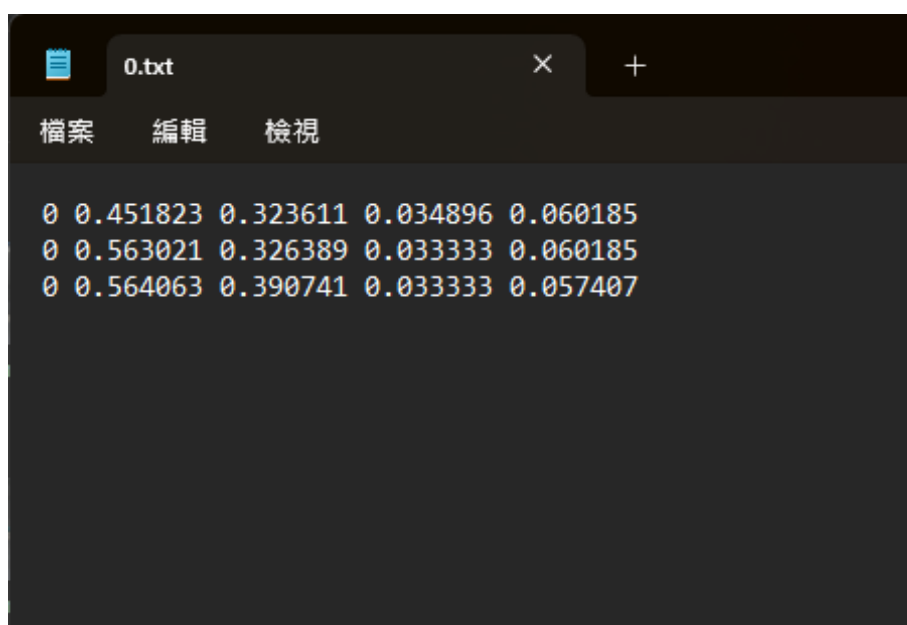


圖 3-4 YOLO 影像儲存格式

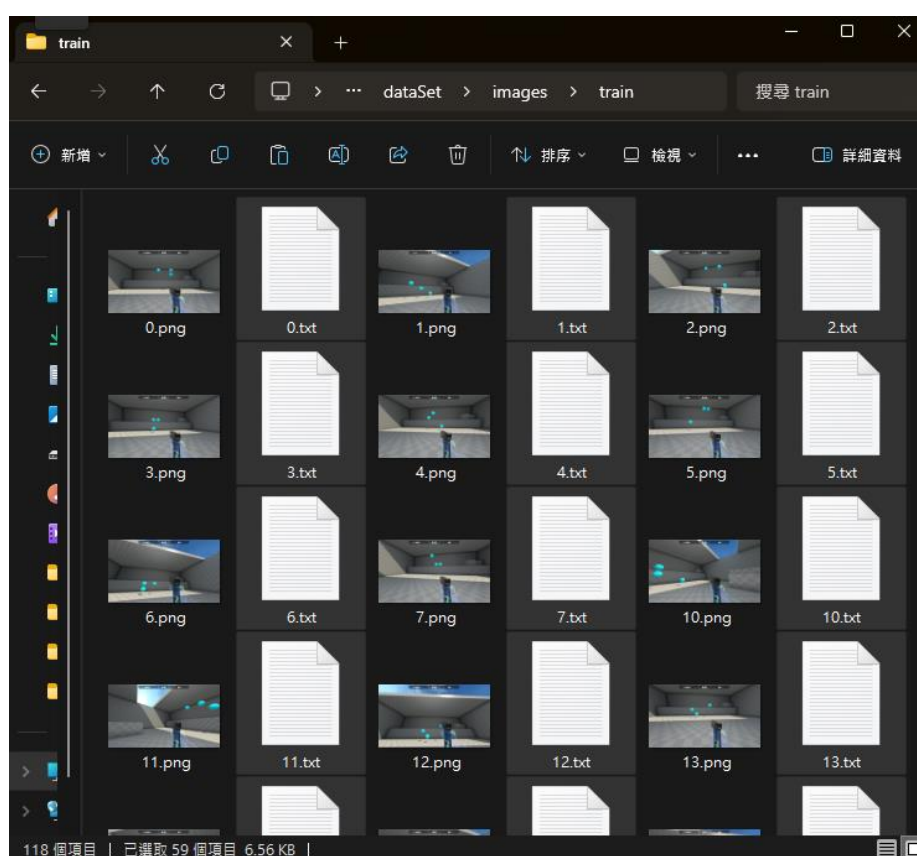


圖 3-5 資料集範例圖

## 3.4 操作遊戲

透過 YOLOv7 官方所提供的模型以及程式碼，我們可以輕鬆地透過導入資料集，即可訓練出最佳的模型，也就是 best.pt 權重檔，訓練後的模型結果可以參閱圖 4-1。

### 3.4.1 計算座標

本研究是在 AimLab 執行期間，擷取螢幕畫面並丟進模型判斷，可以得到物件的 X 座標範圍、Y 座標範圍，也就是  $x_{min}$ 、 $x_{max}$ 、 $y_{min}$ 、 $y_{max}$ ，同時計算出物件的”CONFIDENCE\_THRESHOLD”，也就是信心閾值。

而在執行過程中，不見得只會有球體是圓形的，若是分數有出現 0，分數也可能被誤判為物件，因此我們需要設定信心閾值，本研究是採取當信心閾值大於 0.6 時，才能被當作是我們要偵測的物件，如此可以大大降低誤瞄的產生。

接著就需要處理 x、y 座標的問題，由於之前已經得到了 x、y 座標的範圍，那中心點的位置便得以計算得出。

### 3.4.2 移動滑鼠

在 3.4.1 中，得出了三個物件在畫面中的位置，街上來需要計算的是，三個物件的中心點，以達到迅速射擊的目的。

由於本研究是使用 win32api 進行鼠標的位移，因此在計算偏移量需要特別注意，在 Windows 中，win32api 的座標系統通常會將 (0, 0) 定義為螢幕的左上角，如圖 3-6 的左上角。也就是說，這個座標系統的原點位於螢幕的左上方，x 軸向右遞增，y 軸向下遞增。

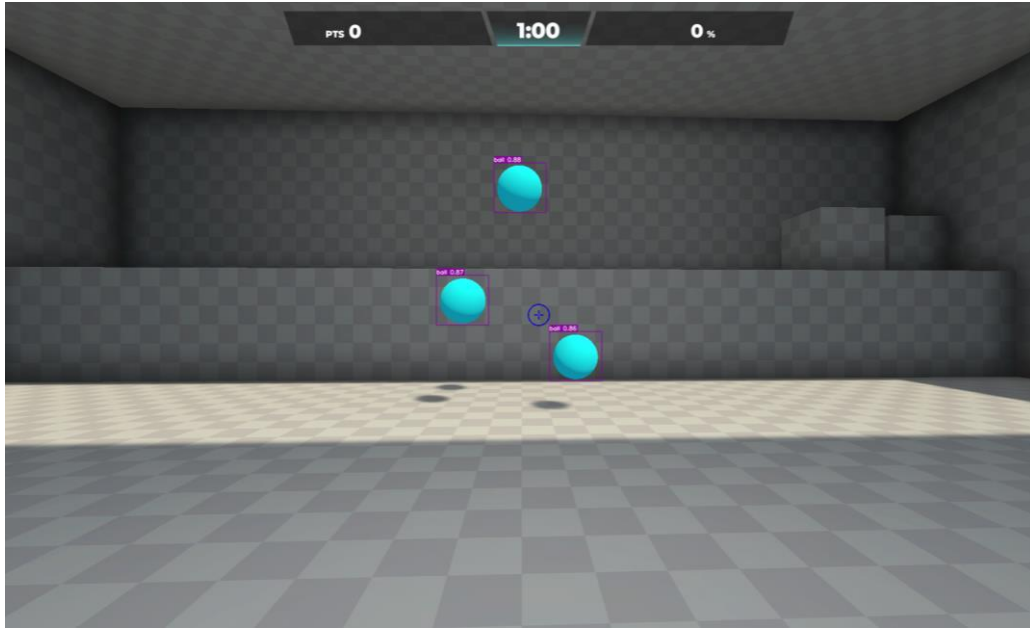


圖 3-6 AimLab 畫面圖 左上方座標為(0,0)

### 3.4.3 討論不同的射擊間隔差距

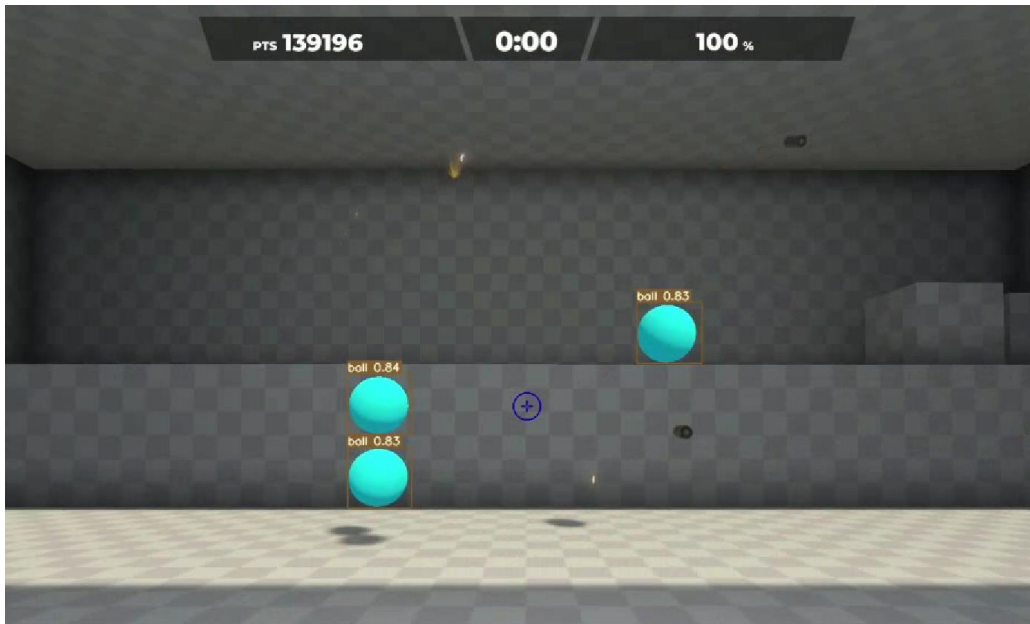


圖 3-7 test1 測試結果



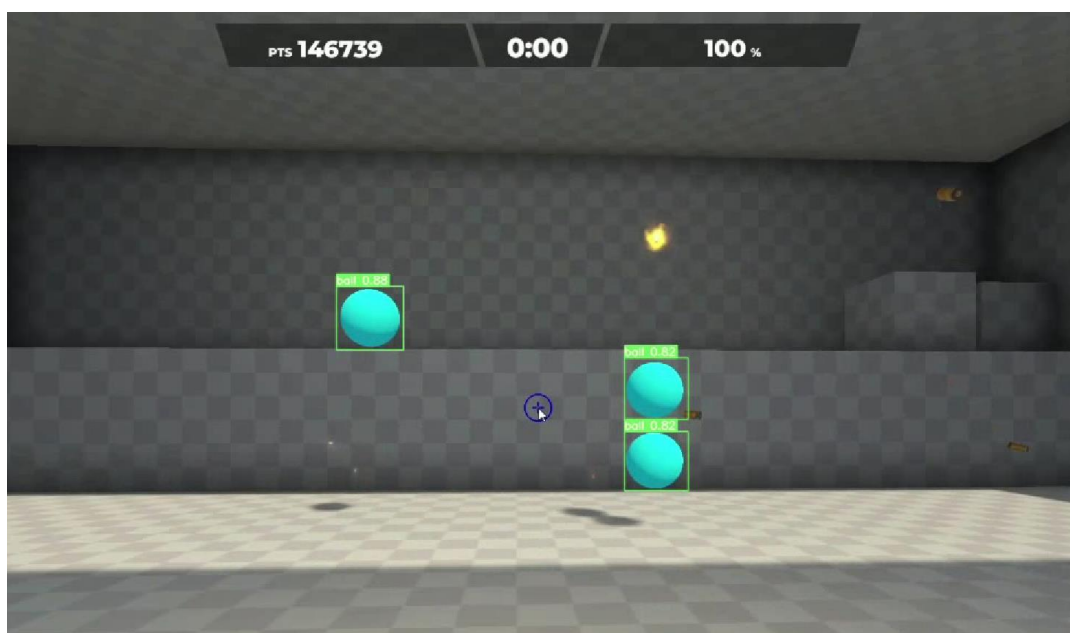


圖 3-8 test2 測試結果

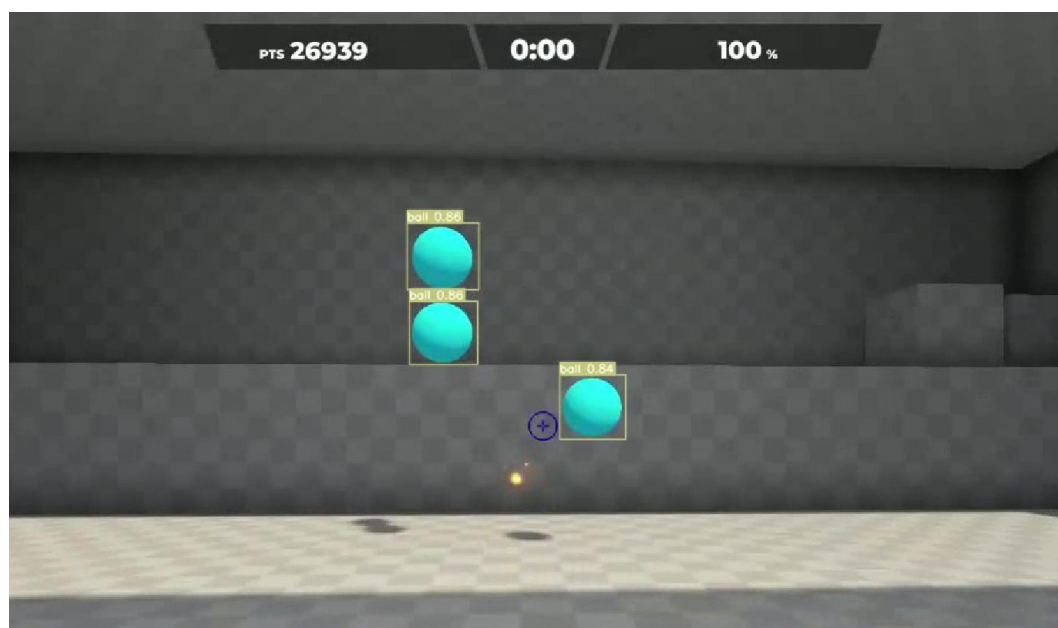


圖 3-9 test3 測試結果

	test1	test2	test3
射擊延遲	延遲 0.001 秒	延遲 0.005 秒	延遲 0.5 秒
遊戲得分	139196	146739	26939
命中率	100%	100%	100%

表 3-3 測試結果比較表

從這個表格當中我們可以看出，不同的延遲會影響遊戲得分，延遲時間長也就變向影響了射擊速度，因此 test3 分數較低；但是值得注意的地方是，在延遲時間為 0.001 秒的 test1，得分反而還不如延遲更久的 test2，這是因為前文所提到的判定問題，延遲過短而造成遊戲無法偵測到其中一些射擊動作。

## 第四章 實驗結果與分析

### 4.1 測試資料比較

我們事先標註好的 60 張照片放在一個資料夾中，之後用程式將其中的文字檔和圖片檔分開，經過 300 回合的訓練後，其訓練狀況如圖 4-1。

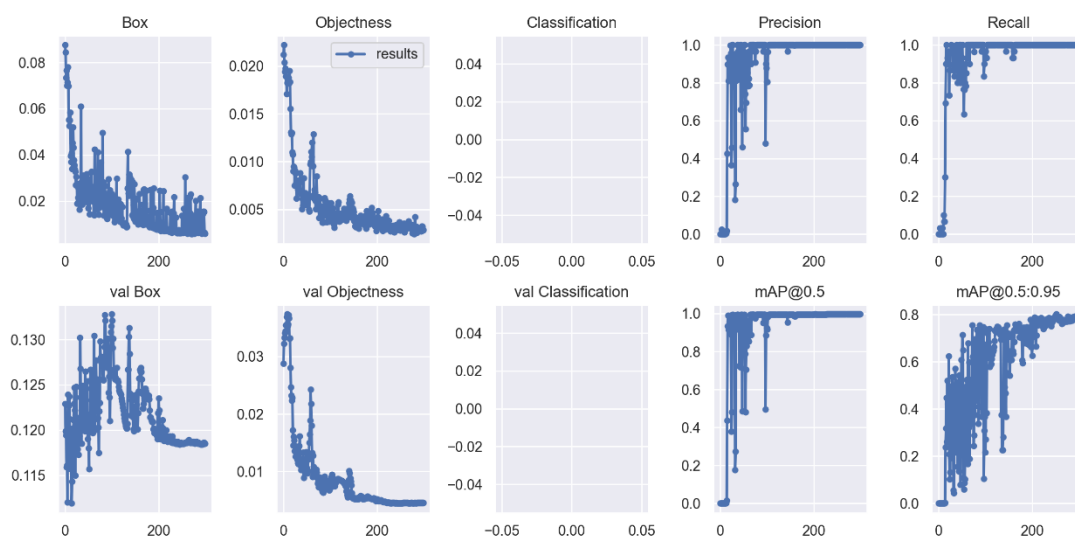


圖 4-1 300epoch 資料集訓練結果

由圖 4-1 中可看出，其結果顯示到了後面的回合後，所有圖片都能辨識成功，其 Precision(精確度)維持在 1，其原因可以歸咎於這款遊戲要辨識的物件很單純，像是圖 4-2 這樣只有一顆藍色的球體，也幾乎不會有其他變形，因此不需要擴增資料集就能以 300 回合訓練好。

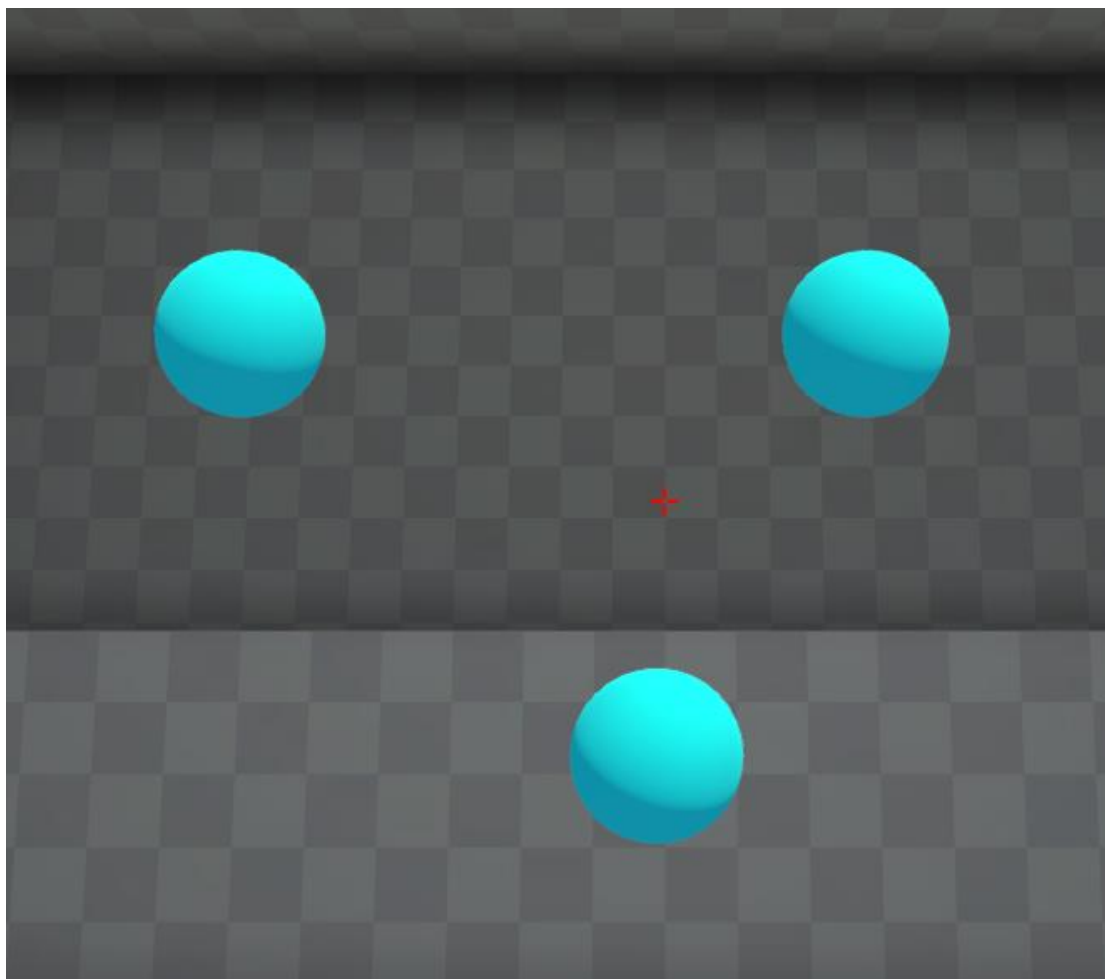


圖 4-2 要辨識的球體物件

## 4.2 實際應用於遊戲影片中

我們首先錄製一段自己實際操作的影片，接著用 YOLOV7 的測試檔案，測試結果如圖 4-3。

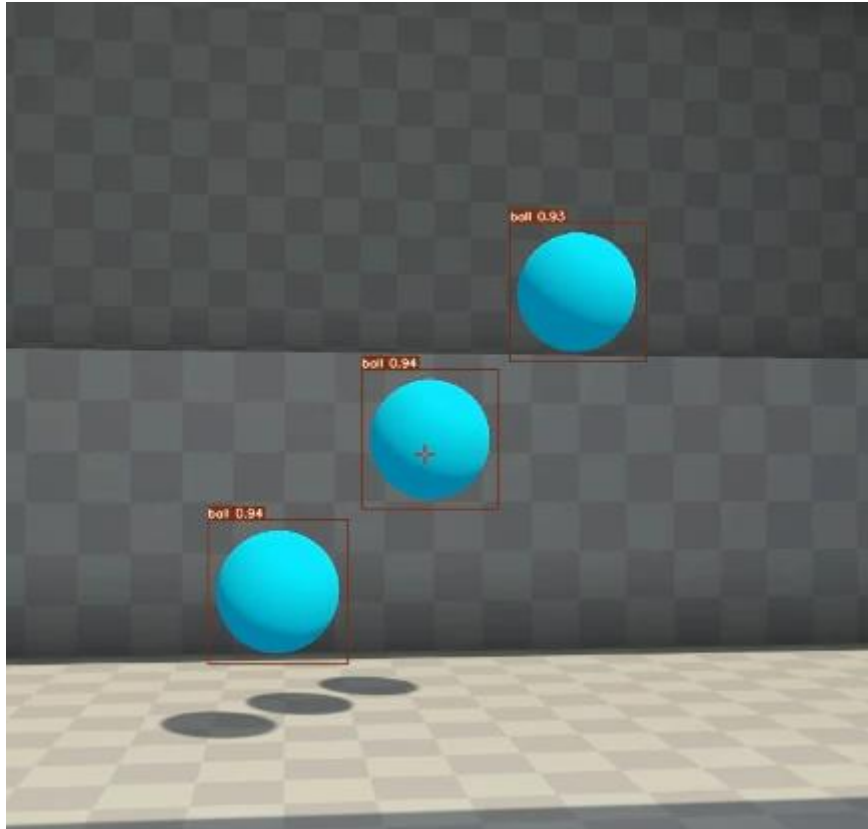


圖 4-3 要辨識的物件

從圖中可以看出，每個物件其信心度都有高達 0.9 以上，結果顯示我們的模型訓練得很成功。

### 4.3 將射擊機器人實際應用於遊戲中

我們先手動開始遊戲，遊戲開始畫面如圖 4-4，之後程式會將遊戲畫面擷取，送入模型預測，再由預測結果去計算座標位置，得出相對座標，其結果如圖 4-5，由圖中顯示我們訓練的機器人其得分位居榜 2，可見我們的機器人訓練得相當成功，特別是在物件擷取、座標計算、移動滑鼠的部分，圖上顯示我們的準確度高達 100%，意思是每次射擊都有命中目標，並不會發生任何的射擊失誤。

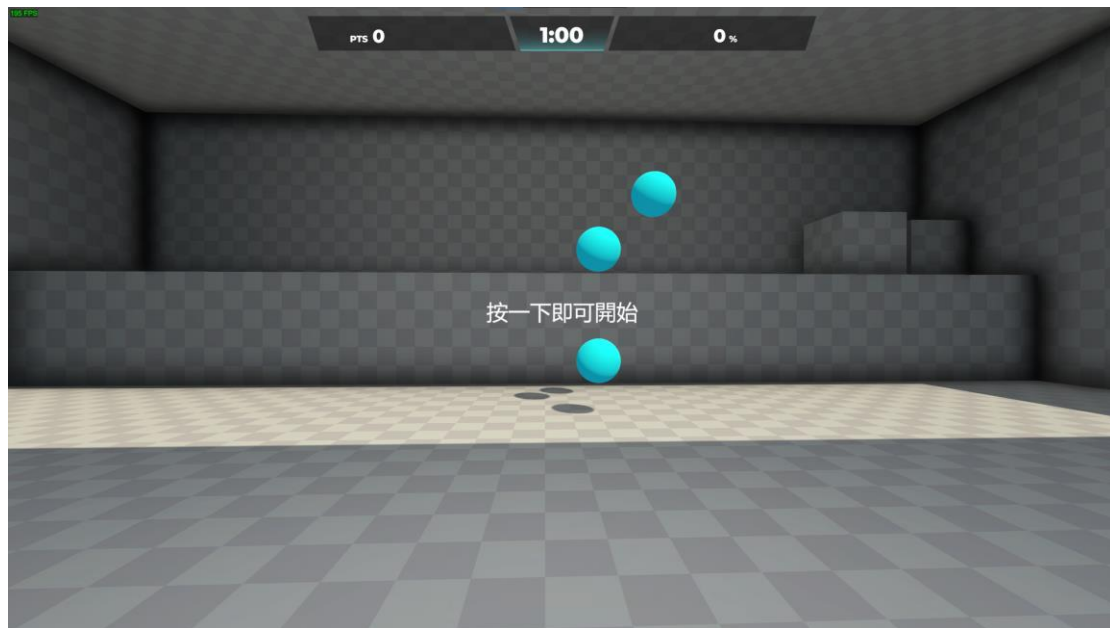


圖 4-4 AimLab 格網射擊遊戲開始畫面



圖 4-5 位居網格射擊的排行榜第二名(截錄自 2023/11/23 止)

## 第五章 結論與未來展望

傳統的外掛多數與記憶體掛勾，而對此遊戲官方也有不少應對方法，但是利用影像辨識的方式，本身並沒有對遊戲程式動手腳，這使官方要機械化的辨識會很困難；更甚者，若是我們將畫面擷取後，使用不同的主機進行計算，並以物理的方式控制滑鼠，偵測這種外掛的難度又會更加提升，所以我們認為這種新型的外掛是需要被多加關注的，因為這可能會很大程度的危害到射擊遊戲的生態。

在未來，利用影像辨識並移動設備進行瞄準，也許可以套用到軍事上，實時截取當下的場面，當看到有敵方進入視野後就以最快的速度瞄準到對方並進行射擊等等，當然以此為延伸就有諸如能否準確判斷是敵是友等困難存在，如何避免誤擊我們認為就是對基於影像辨識所製作的自動射擊機器人的最大挑戰。

## 參考文獻

- [1] 湯仁愷, “應用 YOLOv5 和 CNN 深度學習技術於車牌辨識研究”, 碩士論文, 國立台北科技大學, 2023.
- [2] 朱閔聖, “基於 YOLO 影像辨識之自動化軟體測試系統”, 碩士論文, 國立台北教育大學理學院資訊科學系, 2023.
- [3] Lasse Kurz. (2021). YOLO: You only ~~live~~ look once. Retrieved from <https://blog.cronn.de/en/ai/2021/07/28/yolo-object-detection.html> (2023)