摘要

人工智慧盛行在我們的時代，啟能悄悄從它身邊走過, 偶然的接觸到人工智慧，竟然完全無招架之力的被那科技深深吸引，開啟短暫但我相信還有續集的探索之旅。

一、卷積神經網路(Convolutional Neural Network, CNN)：一種用於圖像辨識和分類的深度學習模型。MNIST 手寫數字辨識和 Cifar-10 圖片判斷都是基於卷積神經網路實現的。

二、YOLO 技術：這是一種物件檢測技術，可以實現快速而準確地檢測圖像中的物體。可製作判別貓狗位址和自動瞄準外掛都是基於YOLO實現的。

三、3D虛擬人物控制：MediaPipe 是 Google Research 開發的一個跨平台的機器學習框架，可以實現從相機、影像等輸入中擷取關鍵信息並進行處理。進一步使用 MediaPipe 模型操控3D虛擬人物。

四、神經風格轉換(Neural Style Transfer)：將一張圖像的風格轉移到另一張圖像上的技術。

五、生成對抗網路(Generative Adversarial Network, GAN)：這是一種利用兩個神經網路模型相互競爭來生成逼真的圖像的技術。GAN 的變體 CycleGAN 可以更加快速地實現風格轉換，GAN 這種技術在許多領域都有廣泛的應用。

前言

一、研究動機

世代的轉換代表的是成長、變動，更是恐慌。農業時代進階到工業時代、工業時代進階到AI時代，都代表著一群企業及人的沒落，同時也創造了另一批企業及人的興起，而現在正值AI時代要進階到人工智慧的時代，不想被這波洪流淹沒，則要加緊腳步迎頭趕上，也正是時機讓我們有個創造的舞台！處在這樣的時間及轉捩點，對我來說是非常興奮的。

二、研究目的

(一) 研究人工智慧所擁有的有趣技術
(二) 辨識系統的應用
(三) 虛擬人物的控制
(四) AI繪圖原理

三、人工智慧 (Artificial Intelligence, AI)

1956年，人工智慧被確立為一門學科，半世紀間經過許多起起落落。如今電腦的運算能力約為 30 年前的 100 萬倍，且近10幾年大數據的快速發展，人工智慧重新過來，許多先進的機器學習技術成功應用於社會中的許多問題。

四、機器學習 (Machine Learning, ML)

而在AI底下有個分支，也就是這次的主題機器學習。從 1980 開始蓬勃興起。機器學習之所以能興起，也歸功於硬體儲存成本下降、運算能力增強、大數據的發展。而機器學習中又有4類的學習方式，分別為監督學習、半監督學習 、無監督學習 、強化學習。

五、深度學習 (Deep Learning, DL)

深度學習又是機器學習的分支，深度學習能自動提取資料特徵，其能力遠遠甩開其它演算法。深度學習參考人腦神經概念，用程式還原神經網路的構造，人工神經網路架構分為 輸入層(input layer)、隱藏層(hidden layer)、輸出層(output layer)。輸入層是資料進入系統的入口，而隱藏層是處理資訊的地方，隱藏層從輸入層或其他隱藏層取得輸入。人工神經網路可以有大量的隱藏層。每個隱藏層分析前一層的輸出，進一步處理，並將其傳遞給下一層，重複直到輸出層，而最終的計算結果就會顯現在輸出層，也就是預測結果。

六、神經網路種類

(一) 前饋神經網路 (Feedforward Neural Network, FNN)

是最古老的神經網路之一，最簡單的神經網路模型，資料經由輸入層通過隱藏層到輸出層，神經元之間沒有連接迴路存在。

(二) 卷積神經網路 (Convolutional Neural Network, CNN)

卷積神經網路通常用於圖片辨識，模仿人類大腦的認知方式，觀察由細微的事物到整體特色。卷積層 (Convolution Layer) 使權重的減少、池化層 (Pooling layer) 壓縮圖片，以此更高效率的判斷圖片。

1. 卷積層 (Convolution Layer)

將輸入的圖像劃分為若干個矩形區域，對每個子區域以相同權重運算，最後加上激勵函數。神經元運算中無須每個輸入都要一個權重，我們稱 共享權值 (Shared weights)，可大幅減少權重數量，藉此減少運算時間。

2. 池化層 (Pooling Layer)

一個壓縮圖片並保留重要資訊的方法，取圖片範圍內最高或平均當做輸出，常用的有最大池化 (Max pooling) 與 平均池化 (Average pooling)。

3. 扁平層 (Flatten Layer)

將多維的輸入壓扁為一維輸出，常用在從卷積層到全連接層的過渡。

4. 全連接層 (Fully Connected Layer)

連接最基本的神經網絡。

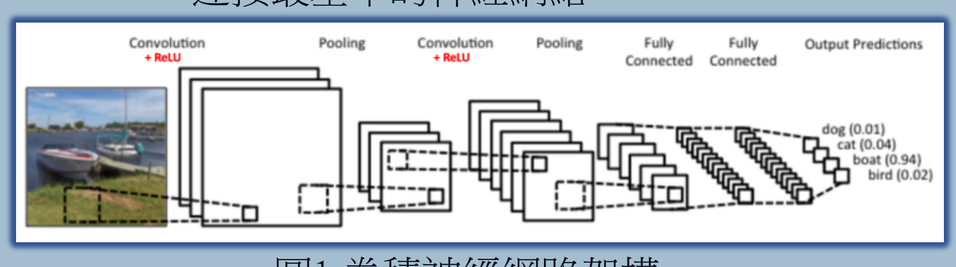

圖1 卷積神經網路架構

(三) 遞迴神經網路 (Recurrent Neural Network, RNN)

最常被用來處理時間和序列相關的問題。與使用前饋類神經網路不同的是，循環類神經網路具備前一層事件的「記憶」，並附加到目前層的輸出內容。

(四) 長短期記憶網路 (Long Short-Term Memory, LSTM)

是進階的遞迴神經網路，解決許多問題。長短期記憶網路會透過三個控制閥(輸入閥、遺忘閥、輸出閥)來決定將什麼資料保存(記憶)下來，而什麼記憶又該捨棄(遺忘)。看似不錯但也因為家入了許多內容導致參數變多，訓練難度提升了不少。

(五) 生成對抗網絡 (Generative Adversarial Network, GAN)

生成對抗網路是種非監督式學習，主要是兩個相互競爭的神經網路 生成網路 (Generative Network) 與判別網路 (Discriminative Network)。生成網路生成圖片，目標騙過判別網路，判別網路判斷是否與資料相同，目標提升鑑定水準，這樣一來一回的對抗促使兩邊互相成長。

七、YOLOv8

YOLO (You Only Look Once) 是一種物件偵測方法，目前共推出8個版本。YOLO 的主要優勢是其快速的運算速度，能夠及時處理圖像。YOLOv8剛好在2023登陸，既然是最新版本，運算成本應該較低，因此選用 YOLOv8。

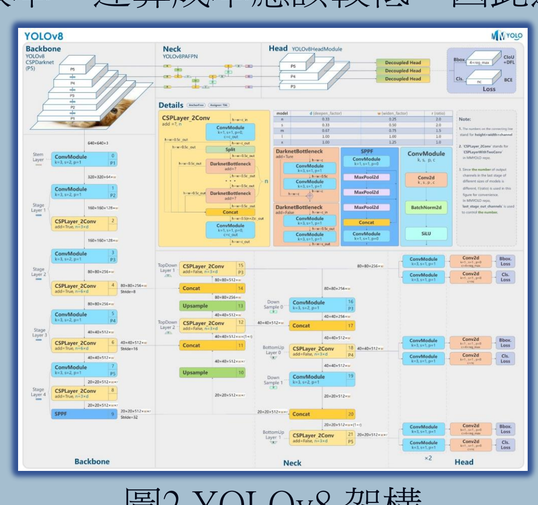

圖2 YOLOv8 架構

研究設備及器材

一、硬體設備

(一) 桌上型電腦
作業系統：Windows 10
CPU：Intel Core i7-12700K
GPU：NVIDIA GeForce RTX 3060
記憶體：32 GB
(二) Logitech C310 HD 網路攝影機

二、軟體工具

(一) Python 3.9：程式語言
(二) C#：程式語言
(三) Unity：遊戲引擎
(四) CSGO：射擊遊戲(測試用)
(五) Anaconda：虛擬環境
(六) Kaggle：數據建模和數據分析競賽平台
(七) Roboflow：線上圖片標註

3D虛擬人物

藉由程式設計，我創造出自己的虛擬人物，這個虛擬人物就可以隨我控制，一個人體結構，就可以有40個控制點，著實令人興奮不已，過去在 虛擬實境 (Virtual Reality, VR) 的遊戲裡，控制虛擬人物大多都以VR穿戴裝置實現，總是需要手把或其他工具來操作這個虛擬人物。使用 MediaPipe 後，發現是否只用一台攝影機，我個人的任何動作都可控制這個虛擬人物，不需任何的手把或工具。並且我相信仍有更多使用的空間。

(一) MediaPipe

MediaPipe 是 Google Research 發表的開源專案，可支援多種語言，擁有許多辨識功能，這次實驗主要使用臉部網路 (Face Mesh)、人體偵測 (Pose)、手部偵測 (Hands)。這些偵測模型可抓出身體各部位，只使用一個鏡頭，並且輸出三維位置。人體偵測原理是訓練時以三維當標籤，臉部網路則是偵測幾個點後再將3D圖形套上。

圖15 MediaPipe專案類別

(二) 人體動作偵測

只能控制虛擬人物太無聊了，因此設計動作偵測，創造互動式小遊戲。希望做出特定動作，角色就可發射子彈，攻擊目標。

1. 資料集

使用 MediaPipe 偵測點位，而動作是需要時間完成的，所以需要在相同時間內完成動作，並將時間內所有偵測到的點當作資料集。

收集的資料集為，發射動作與無動作。

圖16 發射資料

圖17 無動作資料

2. 模型製作

剛好動作與時間序相關，正好可使用長短期記憶網路，可記憶以前重點事件並輸出給下個神經元，最後以密集層連接。

3. 實際執行

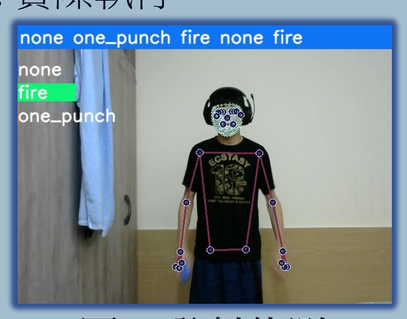

圖18 發射偵測

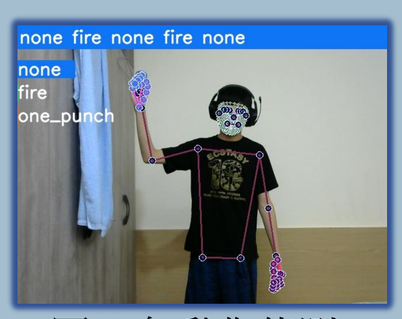

圖19 無動作偵測

(三) 歐拉角 (Euler angles)

物體在三維空間旋轉的方法，三個旋轉軸分別為翻滾(Roll)、俯仰(Pitch)、偏擺(Yaw)。運用臉部網路偵測點，

推算頭部翻滾、俯仰、偏擺。

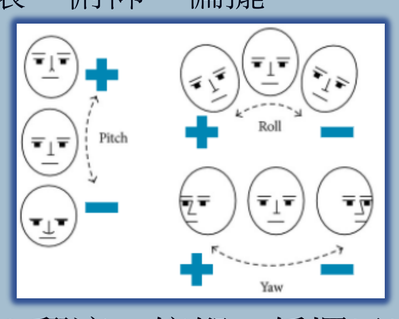

圖20 翻滾、俯仰、偏擺示意圖

(四) 卡爾曼濾波器 (Kalman Filter)

是一種高效率的遞歸濾波器，能夠從包含雜訊的測量中，排除雜訊。MediaPipe 偵測中很難完全無雜訊，卡爾曼濾波器這時就可很好的發揮其作用。

(五) 傳輸控制協定 (Transmission Control Protocol, TCP)

將 MediaPipe 偵測完人體位置後，Python 傳送至 Unity 中的所需工具。傳輸控制協定會在兩個端點間建立連線確保雙方的溝通順暢，其中要求位置(IP)、連接埠(Port)。

(六) Unity

最初不知要使用何種方式呈現虛擬人物，一度嘗試用 Python 建3D模型，但難以執行，後來發現 Unity，簡直與我的需求完全符合。Unity 為2D和3D的遊戲引擎，語言為 C#(完全沒碰過 全部重頭學起)。

(七) 3D虛擬人物模型

大部分虛擬人物皆需要錢，Unity 官方有免費釋出一個人物模型 Unity-Chan，有免費肯定用啊。

圖21 Unity-Chan

(八) 物理骨 (PhysBones)

由 VRChat 開發，在 Unity 中模擬頭髮、衣物、配件物理飄動功能。

(九) 製作過程

版本1：先在 Python 中使用 MediaPipe 人體偵測，再用傳輸控制協定將偵測資料傳給 Unity，使每個傳輸資料控制小方塊，完成後就可簡單看出人體架構了。

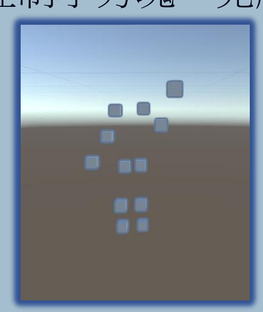

圖22 Unity影像

圖23 現實影像

版本2：將 Unity-Chan 人物模型套入，但3維位置無法控制角色關節活動的，Unity 中有指令可使一個3維位置指向另一個，藉此完成角色控制，最後推算並套用頭部角度。實測發現全身有嚴重震動，卡爾曼濾波器加入後優化許多，但移動速度就相對慢一拍。

圖24 Unity影像

圖25 現實影像

版本3：只有控制角色就稍微無趣些，因此加入第一人稱視角的小型射擊遊戲，遠處放上些許目標物，偵測人體動作判斷是否射擊，藉此擊倒目標物。手指可以表達許多事物，因此也將手指偵測位置套入，但手指相對精細，距離稍遠可能偵測不完全。

圖26 Unity影像

圖27 現實影像

(九) 實際執行

第一人稱射擊小遊戲展示。

圖28 Unity影像

圖29 現實影像

總結

真的很訝異我能夠做出控制虛擬人物程式，以前以為遙不可及的，現在卻在我手中。過程中訪查了上百上千個網站來學習，為了達到這個功能，為此還特別學了一種程式語言。雖然這技術還有需多可改進的地方，像是偵測的準確度，可以使用雙攝影機加上自己設計的模型，也許就可判斷更加準確。

這種技術的應用對多個領域帶來了實際的改進。從VR到元宇宙，這種精確控制的程式為人們提供了更方便、更簡單、更直觀的虛擬體驗。隨著這些技術的進一步發展和普及，可以期待看到更多的人從中受益。