

# 2022 年臺灣國際科學展覽會 優勝作品專輯

作品編號 190003

參展科別 電腦科學與資訊工程

作品名稱 以分散式邊緣運算網路架構實現智慧機器人代理系統之研究

得獎獎項

就讀學校 國立臺灣師範大學附屬高級中學

指導教師 魏宏宇

作者姓名 蕭徹

關鍵詞 機器人(Robot)、YOLO、邊緣運算(Edge Computing)

## 作者簡介



我是師大附中蕭徹，從小就對機械與電腦很感興趣，進到附中的資料班後，又有更多的機會與資源讓我可以鑽研各類程式。我從小就是位電玩迷，在各類遊戲中幾乎都會出現機器人這樣的角色，所以我總是夢想著這些機器人出現在現實生活中，在我遇到困難時幫助我。所以就在國中時接觸到機器人格鬥後就對人型機器人有了一定的了解，也發現其中的問題。流暢度與速度的不足，才會展開這次的研究。

## 摘要

本研究企圖建構一個運用邊緣運算技術(Edge Computing)之人工智能機器人代理人(AI Agent)，並將之運用於實體人型格鬥用機器人的研究開發中。

在此以人型格鬥機器人做為場景需求使用設定目標，運用彈性化模組，加上分散式、嵌入式即時網路技術來降低系統設計的複雜度，整合通訊協議與深度學習 YOLO 影像演算法，進一步運用 ZMP 運動控制理論，以及多維感測器融合技術(sensor fusion)，融合陀螺儀(GYRO)、加速儀(accelerometer)、CMOS Sensors、FSR(Force sensing resistor)作為人形機器人智慧平衡基礎，再藉由圖形識別做為預測辨識以及智慧姿態 ZMP 控制技術作為攻防策略判斷。

整體系統藉由 AI 晶片與嵌入式系統網路作為整合。透過網路即時傳輸環境資訊與指令，使機器人可以知道高層的指令目的資訊。值得一提的是本系統網路設計建構依照仿生哺乳類動物的分層式架構。神經系統將反射以及即時控制交由智慧代理人軟體作為即時演算與控制來達到高性能與彈性發展的需要，未來可用在高等擴充性的人形機器人使用，包括格鬥機器人，人形機器人工地建材搬運、具自平衡醫療外骨骼機器人.....等，使人與機器人能並肩工作，提升人與機器人整合互動。

## 英文摘要

Distributed embedded system design is essential for successful intelligent robotic implementations. The system integration has become more and more complex due to the requirement for size, power saving, weight, and functionality. The distributed edge computing architecture was largely adopted because of its expansion capability and the ease of software/hardware development. This research focuses on the scenario of the Ultimate Battle Humanoid Robot. We applied flexible modular design concepts and distributed embedded real-time network architecture. The proposed edge computing robot architecture can reduce the complexity of traditional ones. It's now easier to build up communication integration for the sensors and improve system efficiency.

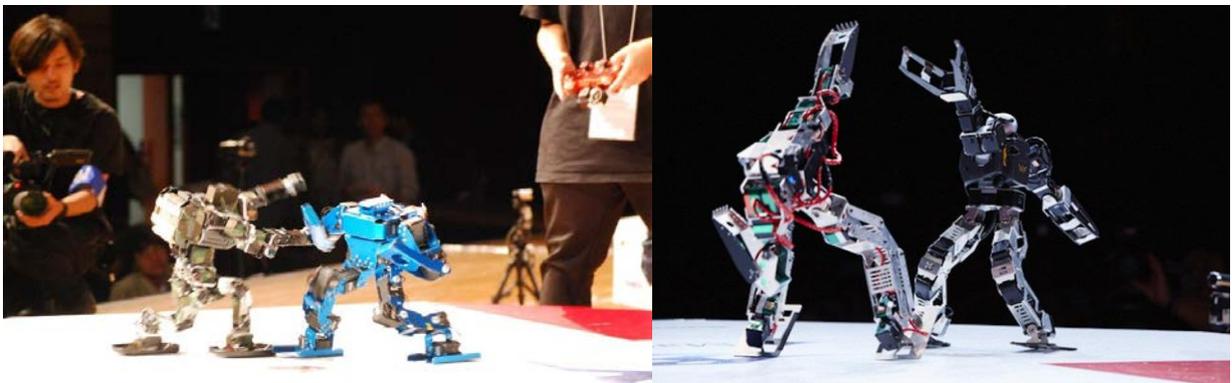
For the robotic AI technology, we attempted to integrate robotics sensor fusion systems, including IMU (Inertial measurement unit), ACC (Accelerometer), CMOS Sensors, FSR (Force Sensing Resistor), and intelligent ZMP (Zero Moment Point) motion control.

The robot's vision AI integrates a tiny YOLO algorithm to support tracking and attacking target. The overall Edge computing system is integrated by the AI chip and the embedded system network. This study further presents a novel distributed embedded platform using an embedded network as the communications backbone, with three layers: 1) control and sensing layer, 2) gateway layer and 3) Internet layer. It's very much alike to the human nervous system. In the future, we picture that our robot edge computing platform can be easily integrates with home IoT network through TCP/IP. People will be able to combine many kinds of IoT Edge computing technology with intelligent Robots.

# 壹、前言

## 一、研究動機

人形機器人的競技舞台[1]一直是一個很熱血且吸睛的科學戰鬥遊戲，滿足了許多機器人與科學迷的熱情，仿生機器人的格鬥夢想是一直以來很科幻動漫迷的夢想。時代的推演格鬥機器人技術的發展推演著機器人基礎科學與關鍵零組件的進步與發展。這競技場由於 F1 賽車帶動整個汽車工業發展一般帶動機器人技術不斷的進步。



圖(1) ROBO-ONE 雙足機器人格鬥比賽舉行

然而現有格鬥機器人都是藉由實體遙控即時對戰連線概念來實現。操作者忙碌的觀察機器人的狀態，再透過遙控器提供每個操控指令，這樣的方式在人工智能輔助操控上較為不足。但是，一直以來要設計一個具備複雜性能力的人工智能技術代理人機器人其實不容易。過去要設計此人工智能代理機器人困難瓶頸在於下列幾點：

- (一) 人形機器人動力結構系統與仿生機構設計難度高，必須考慮附載大扭矩以及平行性的問題。
- (二) 人形機器人系統結構複雜，肢體分散於身體部位，馬達控制制動元件網路系統複雜即時控制以及感測回授運算複雜整合不易。
- (三) 由於控制技術複雜且多樣，一般運算平台運算越強大功耗、耗能就越大，能量電池容量越高成本也越高。系統設計要實現小體積、低工耗、可彈性泛用不容易。

基於上述原因，一個彈性的系統平台架構非常重要。過去的控制技術僅止於馬達控制指令的接收與執行。然而從許多科幻電影與遊戲中我們發現，人類對於人工智能的代理操作期



圖(2)《泰坦降臨 2》

待絕對不僅於此，人們設想未來能夠有一個值得信賴的隨行 AI 夥伴，例如近期坊間有一人氣遊戲《泰坦降臨 2》[2]，它是一款第一人稱的機器人冒險遊戲，遊戲設定的特別之處是玩家可以在遊戲中召喚一部巨大的機器人-泰坦，並且操縱它與其他人對決。它同時也具備智慧，可以主動告訴你敵人的所在位置，並分析環境中有什麼危險，是玩家的武器兼戰友。

人工智能代理引擎若能成功主管機器人的控制，例如：步行的平衡控制、摔倒爬起、閃躲、甚至自動操作武器瞄射敵人、解除敵人的追蹤鎖定及系統自我保護，這些都可以讓人與機器人成為互動更密切的夥伴關係。

研究者為機器人的深度玩家，深深為此類型遊戲著迷，並發想與遊戲結合，創造出一款具人工智能代理人的新型 AR 機器人戰鬥遊戲。

## 二、研究目的

為了開發並實現格鬥機器人代理人技術，探討分析過去文獻，研究者整合出下列幾個解決問題的方向，作為本研究之研究目的。

### (一) 分散式即時嵌入式系統

歷史上在很多機器人技術工程的相關研究提到[3]，嵌入式系統設計對於成功的智能機器人實施至關重要。構建可以執行複雜任務的機器人需要大量的計算能力和系統集成工作[4]-[6]。其中，必須考慮的問題之一是：考量人形機器人的體積與重量，這樣的複雜智能機器需要什麼樣的嵌入式系統？歷史上著名 SONY SDR-4X 需要 60 多個處理器和每秒超過 22.6 億條指令（MIPS）的計算能力 [6]。計算架構本質上是分散式架構的，因為它不太可能將所有原始信息轉儲到單個 CPU 中。信號互連、控制和信息處理應像機械結構一樣模塊化。若沒有精心考慮與設計，整個嵌入式系統可能難以開發、維護和擴展。因此，分佈式嵌入式系統可能是一個有利的選擇。大多數分佈式系統在其處理單元之間具有一定的網絡拓撲。

### (二) 以 TCP/IP 為基礎的機器人嵌入系統網路邊緣運算單元

1998 年 5 月，Journal of Internet Computing 發表了關於嵌入式以太網路技術的特刊，重點介紹了嵌入式 Internet 技術的最新發展和工業應用 [7]。該期刊指出傳統台式 PC 環境的諸多限制。相對而言，嵌入式系統，則允許人們運用手機、PDA 等手持終端設備，或工廠的自動化機器控制器等設備 [8]、[9]，透過以太網無線網路，發展出一套全新的機器人嵌入式系統通信網，在未來極具優勢。

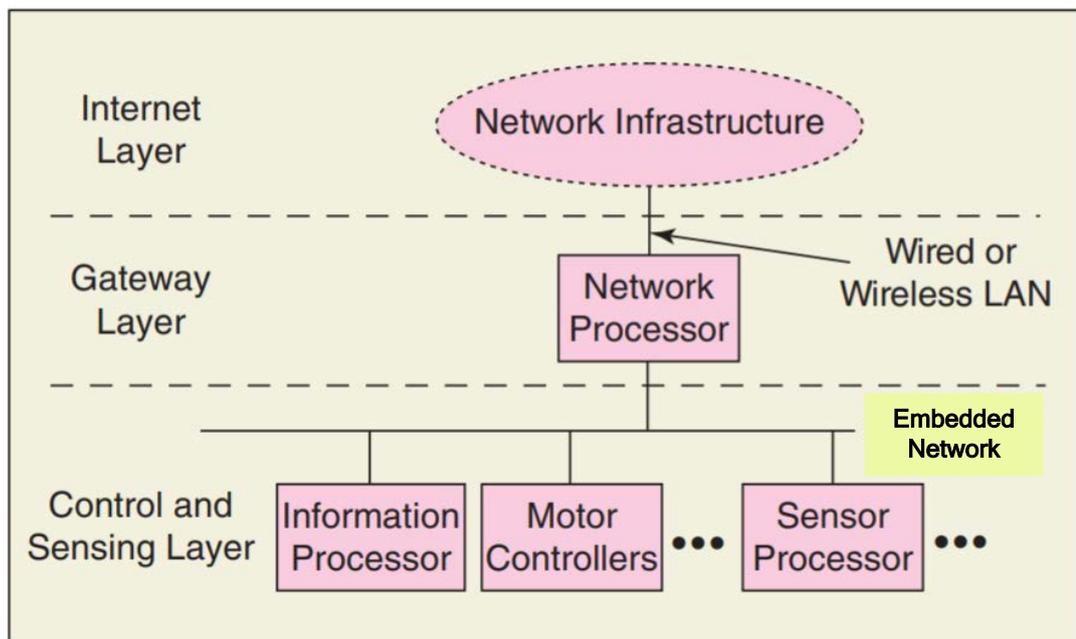
以以太網路技術作為機器人嵌入系統之溝通主幹的好處如下：

1. 傳輸控制協議/互聯網協議 (TCP/IP) 多年來已被證明是穩定的，並且也是被公開運用的。
2. 當機器人與接入網路（公共網路）和家庭網路互連時，設計者必須非常重視相關的安全問題。從歷史上看來，IP 位址的設定已經仔細考慮過這些問題，而且 IP 技術有能

力解決這些問題。

3. 豐富的應用程序和工具使以太網在統一系統接口和降低系統複雜性方面非常有效。例如，它可以用作並行處理中的通信主幹 [10]-[12]。最新技術已將以太網帶寬推高至 Gb/s/Tb/s 範圍 [13]，導致建議將其納入學術和工業領域的片上系統 (SoC) 設計 [14]-[16]。在頻帶寬和系統成本方面，以 IP 為基礎的乙太網/無線網路為嵌入式應用提供了極具競爭力的解決方案。

下圖示意性地描述了本研究所提出的模型。



圖(3) 《機器人分散式網路模型》

機器人系統由三層組成[3]：互聯網層、網關層、控制和傳感層。顯然，這種結構模仿了當今網絡基礎設施的拓撲結構，其中控制和感知層就是所謂的 Intranet。但是，不同之處在於必須為 RT 控制強制執行 RT 消息傳遞（如電機同步）。實施建議的 H RTP 以提供 RT 消息傳遞。重要的是，通過保持 TCP/IP 兼容性，機器人系統可以輕鬆地在該層內擴展（例如通過添加另一個處理器或傳感器）。

此外，互聯網與控制層和傳感層之間的機器人內部通信變得透明（例如網關層中的簡單橋接功能）。藉此可以藉由 TCP/IP 實現高通透的即時通訊控制網路。

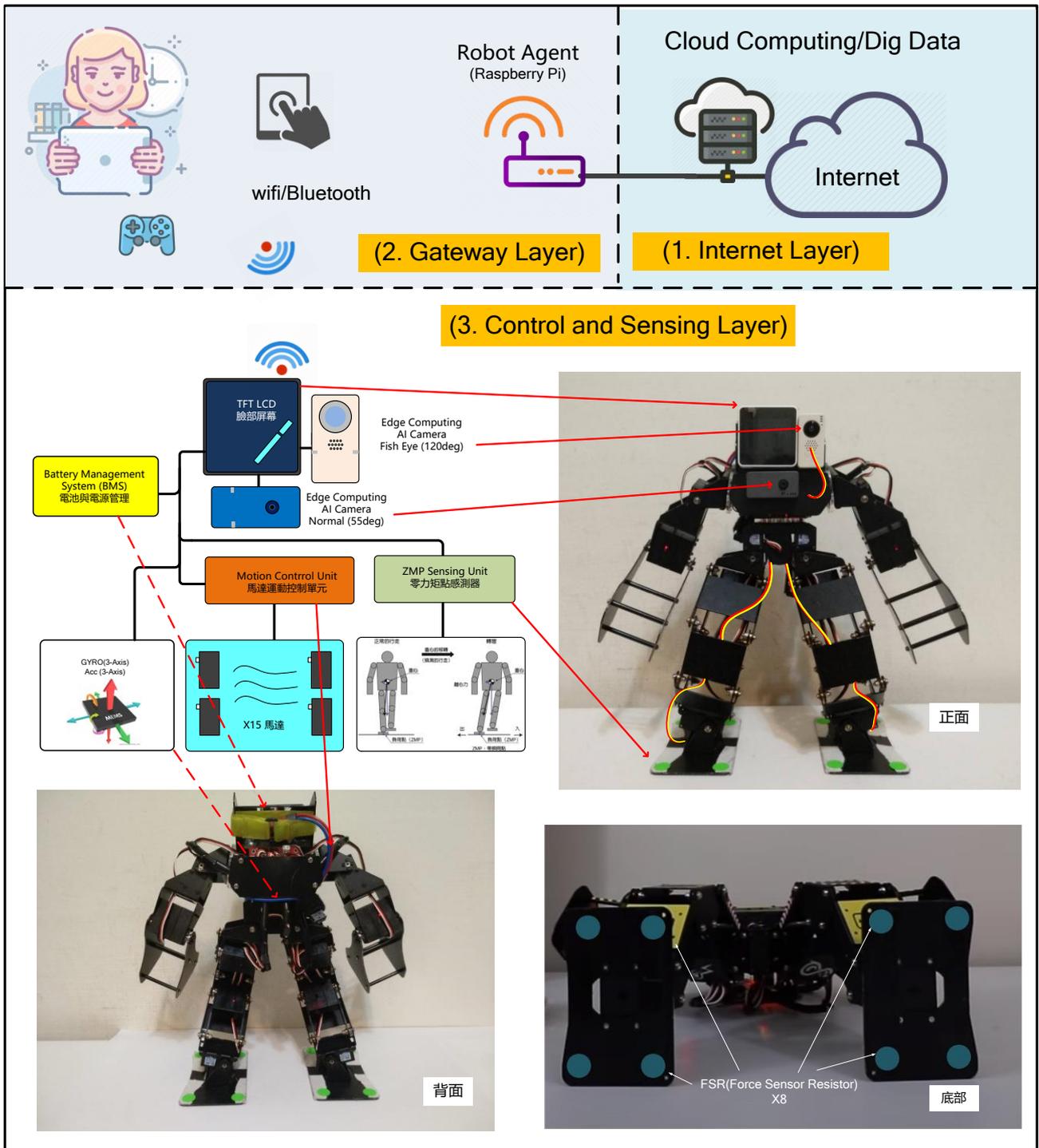
### (三) 分散式多維感測器融合技術(sensor fusion):

在此自主格鬥機器人系統上，研究者使用分散式多為感測器融合技術，包含以下項目：

1. 陀螺儀(GYRO)模組
2. 加速儀(accelerometer)模組
3. FSR(Force sensing resistor)模組
4. ZMP 作為人形機器人智慧平衡基礎模組
5. CMOS Sensors 結合深度學習 YOLO 影像演算法之辨識

人形機器人機構設計對於人形機器人步態有很大的影響，由於關節較多，因此關節與關節的連接緊密程度，會影響到機器人的步態的穩定性與流暢性，如果機構連結的鬆散會造成一定的程度晃動，在機器人行走時會導致重心不穩定，加上機構的比重讓機器人整體的重心位置高度改變，或是機構的形變也會讓行走上造成許多困難。為了做平衡補償以及格鬥過程的跌倒自主爬起在此使用陀螺儀 GYRO 做角速度偵測以及加速規感測做姿態補償。我們對於平衡引源使用 ZMP 演算法藉由腳底壓力感測器是為了計算出零力矩點，機器人的零力矩點定義為機器人運動時，存在一點其反作用力、慣性力的淨力矩和為零的點。由於機器人在平衡狀態時，零力矩點等於壓力中心點，本研究以壓力感測器的感測值來計算零力矩點。

上述之陀螺儀、加速儀、FSR、ZMP 計算模組、CMOS Sensors 結合深度學習 YOLO 影像演算法模組皆以前述之嵌入式分散式網路模型概念實現，如下圖所示。有關其細節將於後續章節介紹。



圖(4) 《機器人分散式網路模型》

## 貳、研究方法

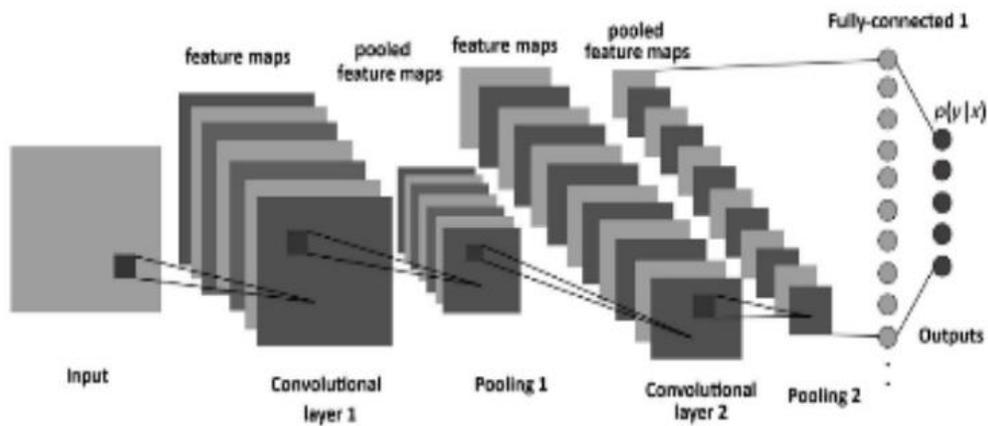
### 一、ZMP 作為人形機器人智慧平衡基礎模組[17]

腳底壓力感測器是為了計算出零力矩點，機器人的零力矩點定義為機器人運動時，存在一點其反作用力、慣性力的淨力矩和為零的點。由於機器人在平衡狀態時，零力矩點等於壓力中心點。

### 二、CMOS Sensors 結合深度學習 YOLO 影像演算法之辨識

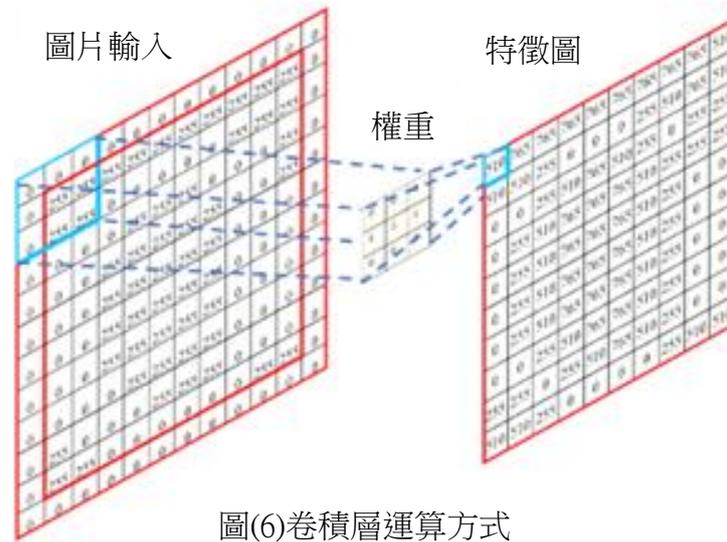
為了使機器人能夠藉由視覺系統找到敵方的機器人，在此加入物體偵測模型，用以辨識物品在圖中的位置，現有的物體偵測方法是用 CNN（卷積神經網路 Convolutional Neural Networks）[18]演算法辨識影像做到分類，並搭配深度學習模型來確定物品位置。

(一)CNN：CNN 的過程分為卷積層 (Convolution)、池化層 (Pooling)、平坦層 (Flatten)、全 連接層(Fully Connected Layer)、輸出層 (Output)這五層，流程如圖(5)：

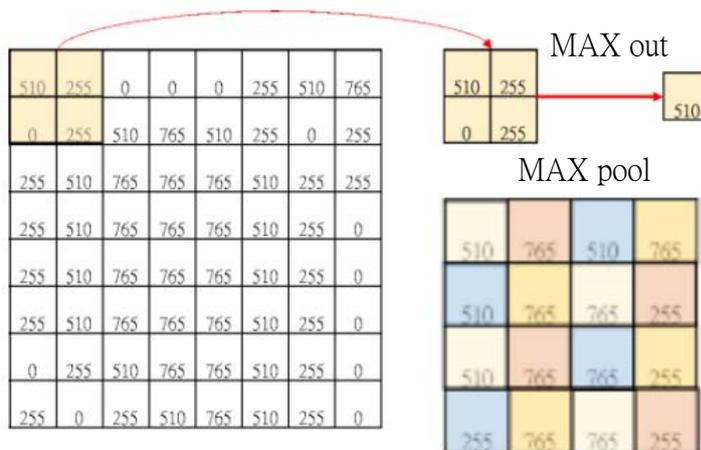


圖(5) CNN 流程示意圖

1. 卷積層：先將圖片二值化，為了保持圖片輸出與原有大小相同，在圖片的最外圍補上 0，之後依序取輸入與權重大小相同的矩陣，進行矩陣相乘運算，輸出特徵圖。特徵圖中將保留輸入圖片明顯的特徵。

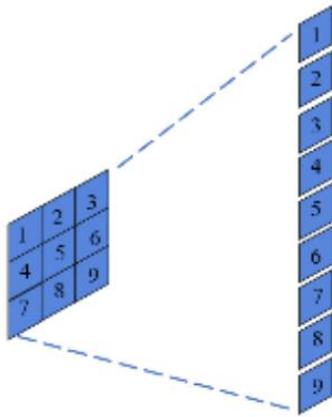


2. 池化層：池化目的在於減少資料量、保留明顯特徵、減少過擬合（overfitting）的發生，將圖片分成多個小塊，保留每塊的最大值輸出，輸出將使特徵圖降維。

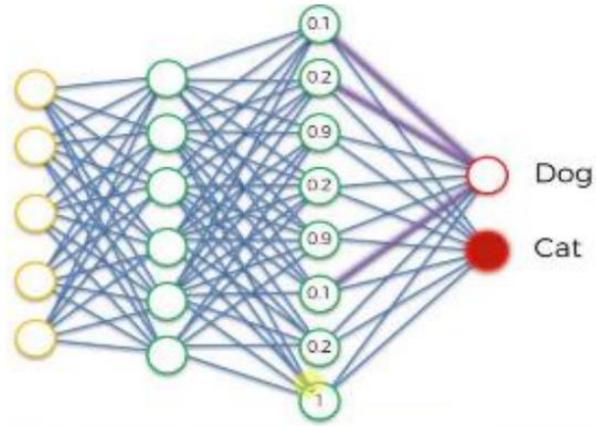


圖(7) 池化層運算方式

- 平坦層：為了連接全連接層，將特徵圖轉為一維陣列。
- 全連接層與輸出層：將平坦後的結果連接至神經網路，進行運算並輸出可以得到辨識的權重。



圖(8) 平坦層運算方式

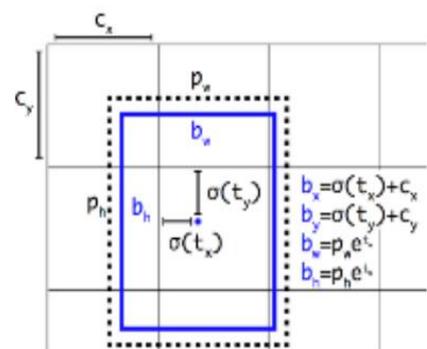


圖(9) 全連接層運算方式

深度學習模型：常見的深度學習模型大致上分為兩種，運用預先篩選（Region Proposal）的 RCNN（Regions with CNN features）與運用迴歸分析（Regression Analysis）的 YOLO（You only look once），經過比較兩種類型的模型，採用 YOLO 模型，因 YOLO 比起 RCNN 辨識更快速且辨識準確度並不差，使用的效能較少，因此較適合使用 YOLO 模型進行辨識。

(二) YOLO 模型：在 YOLO 模型中只有卷積層與池化層，移出了全連接層，能夠預訓練不同大小的圖片。先是將圖片分成  $S$  乘  $S$  方格，判斷結果包含類別、位置與概率。計算邊界框對於每一個類別出現的可能性，機率高的方格會以中心每小格用 5 個不同尺寸的 anchors box 進行預測。

- 預測框：對應格子 cell，左上角的距離為  $(C_x, C_y)$ ， $\sigma$  定義為 sigmoid 啟動函數，處理後，將函數值約束在 0~1 的範圍，讓預預測中心在 cell 中，不會偏離 cell。預測框 anchor 的寬高為  $(P_w, P_h)$ ，預測位置  $(b_w, b_h)$  是相對於 anchor 的寬高，並乘上係數。



圖(10) YOLO 預測框計算

2. 訓練模型的方式：將幾百張物品經過標記的圖片，丟給電腦去學習不同物品的特徵，讓電腦以減少誤差為目標去修改權重，達到更準確的辨識物品。誤差主要分析的是邊 界框的內容，包含五個預測值，中心座標 (x,y)、長寬 (w,h) 和準確度 IoU（檢測評價函數 Intersection over Union）。

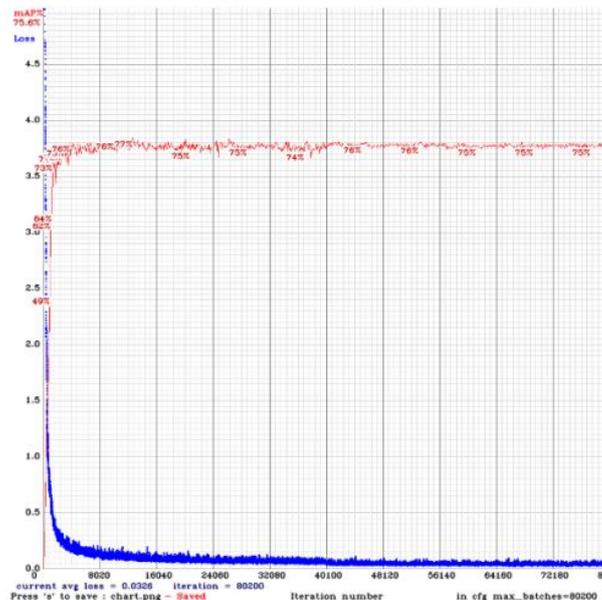
$$\begin{aligned}
 loss_t = & \sum_{l=0}^W \sum_{j=0}^H \sum_{k=0}^A \\
 & 1_{Max\ IOU < Thresh} \lambda_{noobj} * (-b_{ijk}^o)^2 \\
 & + 1_{t < 12800} \lambda_{prior} * \sum_{r \in (x,y,w,h)} (prior_k^r - b_{ijk}^r)^2 \\
 & + 1_k^{truth} (\lambda_{coord} * \sum_{r \in (x,y,w,h)} (truth^r - b_{ijk}^r)^2 \\
 & \quad + \lambda_{obj} * (IOU_{truth}^k - b_{ijk}^o)^2 \\
 & \quad + \lambda_{class} * (\sum_{c=1}^C (truth^c - b_{ijk}^c)^2))
 \end{aligned}$$

$$\begin{aligned}
 IoU = & \frac{\text{預測範圍} \cap \text{標記範圍}}{\text{預測範圍} \cup \text{標記範圍}} \\
 & \text{IoU 計算} \quad 2.2
 \end{aligned}$$

YOLO 辨識誤差數學式

2.1

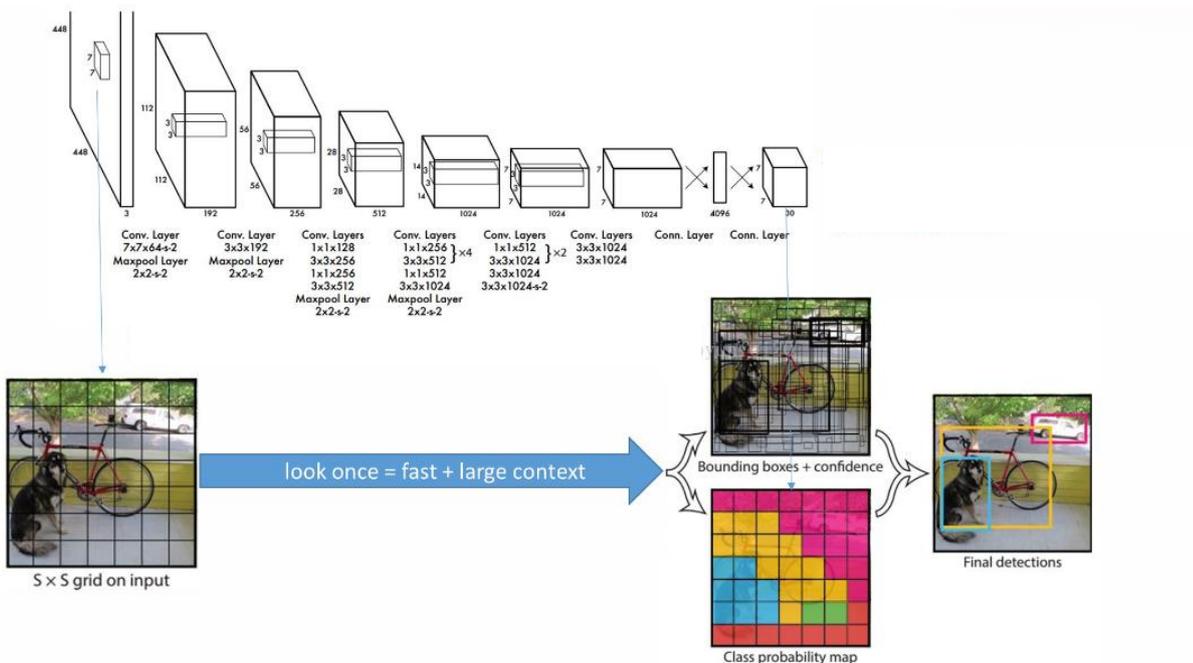
3. 誤差與訓練筆數的關係：下表是在做模型訓練時，誤差與訓練筆數的關係，可知在訓練時隨著訓練筆數增加，辨識誤差會逐漸下降，辨識也會越精準。



圖(11) YOLO 訓練損失與準確度

相比於 R-CNN 系列的方法，YOLO 提供了另外一種思路，將 Object Detection 的問題轉化成一個 Regression 問題。給定輸入影像，直接在影像的多個位置上回歸出目標的 bounding box 以及其分類類別。

YOLO 是一個可以一次性預測多個 Box 位置和類別的卷積神經網路，能夠實現端到端的目標檢測和識別，其最大的優勢就是速度快。事實上，目標檢測的本質就是迴歸，因此一個實現迴歸功能的 CNN 並不需要複雜的設計過程。YOLO 沒有選擇滑動視窗 (silding window) 或提取 proposal 的方式訓練網路，而是直接選用整圖訓練模型。這樣做的好處在於可以更好的區分目標和背景區域，相比之下，採用 proposal 訓練方式的 Fast-R-CNN 常常把背景區域誤檢為特定目標。

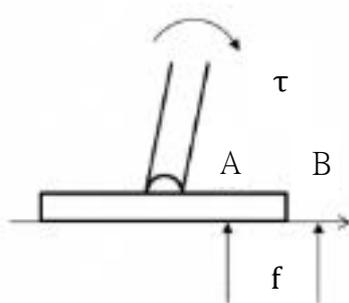


圖(12) YOLO 流程圖

### 三、零力矩點 ZMP 與姿態穩定

零力矩點 ZMP(zero moment point)，字面上意思即合力矩為零的作用點。

人類的步態運動之所以不會摔倒，是因為遵循物理條件下重心對地面的投影點一般來說都落在腳底所構成的支撐多邊形範圍內。將此概念套用到人型機器人上，M. Vukobratovic' 和 J. Stepanenko 兩位學者在 1972 年首度提出 ZMP 觀點的研究[12]。而這個概念的出發點是由於早在人型機器人之前，具有固定基座類型的機械手臂已經研究一段時間了，所以把人型機器人的支撐腳底看成固定端，CoM 軌跡的計算就如手臂的末端操作器可簡化成關節路徑追跡的模式來達成機器人步態運行。但其實這樣的 ZMP 穩定性判據並不是絕對的充要條件，因為人類運動常常是處在動態平衡的狀態，ZMP 點超出支撐多邊形範圍還是可以藉由慣性力來維持平衡；抑或是前述提到的點足式機器人也是可以在不具支撐多邊形概念下實現穩定運動。而各國研究人型機器人的團隊利用 ZMP 作為運動穩定判據也占了一大門派，日本方面多屬此類。那既然 ZMP 穩定判據不是絕對的，那何以獲得廣泛應用？原因還是在於關節路徑追跡，當計算出的關節軌跡可以使足底 ZMP 落在腳底支撐多邊形範圍內，即表示該組運動軌跡不會在腳底發生欠驅動的邊緣翻轉自由度，如圖(13)。



圖(13)當 ZMP 落在 A 處時，無欠驅動翻轉。當落在 B 處時，腳底無法在 B 處給予地面作用力來平衡，故產生翻轉自由度。

ZMP 定義根據 Vukobratovic 和 Borovac 的論述，ZMP 定義為地面上的一個點，對於此點，慣性力與重力造成的合力矩將沒有沿地面水平軸向的分量。令  $m_i$  為機器人各部件質量， $c_i$  為各部件 CoM 於大地座標系之位置， $\omega_i$  為部件繞 CoM 之角速度， $\tau$  為卡氏座標三軸向力矩，則

$$\sum_{i=1}^k [m_i(c_i - ZMP) \times (c_i \ddot{+} g) + I_i \dot{\omega}_i + \omega_i \times I_i \omega_i] = \tau \quad 2.3$$

$$\tau_x = \tau_y = 0 \quad 2.4$$

機器人動態模型與 ZMP 導出除了確切的 D-H 運動模型，要描述機器人和環境間交互作用的動力模型也有其定義。估測 ZMP 的動力模型有將機器人視為實際的多連桿剛體系統如式 2.3，也有視為一個多質點系統或是單質點系統的倒單擺簡化模型。剛體系統和質點系統上的差別在於慣性矩陣的有無，令總質量為  $m$ ，CoM 位置  $c$ ，大地水平面反作用力  $f$ ，繞 ZMP 點之 Z 軸向力矩  $\tau_{ZMP}$ ，對原點的力矩關係式

$$\tau_o = ZMP \times f + \tau_{ZMP} \quad 2.5$$

根據動量  $P$ 、角動量  $H$  對應作用力與力矩的關係

$$\dot{P} = mg + f \quad 2.6$$

$$\dot{H}_o = c \times mg + \tau_o \quad 2.7$$

將式 2.6 帶入式 2.5，再將結果代入式 2.7

$$\begin{aligned} \tau_{ZMP} &= H_o - c \times mg + (\dot{P} - mg) \times ZMP \\ \begin{bmatrix} \tau_{ZMP,x} \\ \tau_{ZMP,y} \\ \tau_{ZMP,z} \end{bmatrix} &= \begin{bmatrix} \dot{H}_{o,x} \\ \dot{H}_{o,y} \\ \dot{H}_{o,z} \end{bmatrix} - \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \times m \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \left( \begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix} - m \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \right) \times \begin{bmatrix} ZMP_x \\ ZMP_y \\ ZMP_z \end{bmatrix} \\ \begin{bmatrix} \tau_{ZMP,x} \\ \tau_{ZMP,y} \\ \tau_{ZMP,z} \end{bmatrix} &= \begin{bmatrix} \dot{H}_{o,x} + ZMP_z \dot{P}_y - ZMP_y (\dot{P}_z + mg) + mc_y g \\ \dot{H}_{o,y} - ZMP_z \dot{P}_x - ZMP_x (\dot{P}_z + mg) + mc_x g \\ H_{o,z} - ZMP_z \dot{P}_y - ZMP_y \dot{P}_x \end{bmatrix} \end{aligned} \quad 2.8$$

根據 ZMP 的定義

$$\begin{bmatrix} \tau_{ZMP,x} \\ \tau_{ZMP,y} \end{bmatrix} = \begin{bmatrix} H_{o,x} \dot{\phantom{y}} + ZMP_z \dot{P}_y - ZMP_y (\dot{P}_z + mg) + mc_y g \\ H_{o,y} \dot{\phantom{x}} - ZMP_z \dot{P}_x - ZMP_x (\dot{P}_z + mg) + mc_x g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad 2.9$$

可導出平面上 ZMP 位置

$$\begin{cases} ZMP_x = \frac{mgc_x + ZMP_z \dot{P}_x - H_{o,y} \dot{\phantom{y}}}{mg + \dot{P}_z} \\ ZMP_y = \frac{mgc_y + ZMP_z \dot{P}_y - H_{o,x} \dot{\phantom{x}}}{mg + \dot{P}_z} \end{cases} \quad 2.10$$

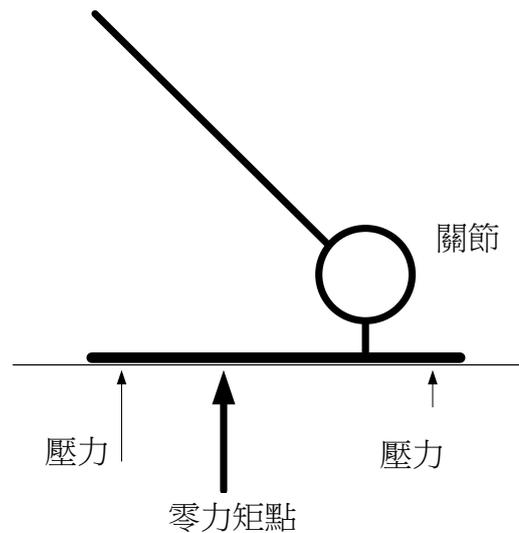
因為 ZMP 是落在地面上的點，所以  $ZMP_z = 0$ ，而  $\dot{P}_z = m\ddot{c}_z$  套入式 2.10 並整理

$$\begin{cases} ZMP_x = \frac{mgc_x - H_{o,y} \dot{\phantom{y}}}{m(\ddot{c}_z + g)} \\ ZMP_y = \frac{mgc_y - H_{o,x} \dot{\phantom{x}}}{m(\ddot{c}_z + g)} \end{cases} \quad 2.11$$

式 2.11 為機器人之動態與 ZMP 的關係，此常應用在一般的分析中

#### 四、FSR(Force sensing resistor)作為人形機器人智慧平衡基礎

腳底壓力感測器是為了計算出零力矩點，機器人的零力矩點定義為機器人運動時，存在一點其反作用力、慣性力的淨力矩和為零的點。由於機器人在平衡狀態時，零力矩點等於壓力中心點，本論文以壓力感測器的感測值來計算零力矩點[19]，如圖(14)。



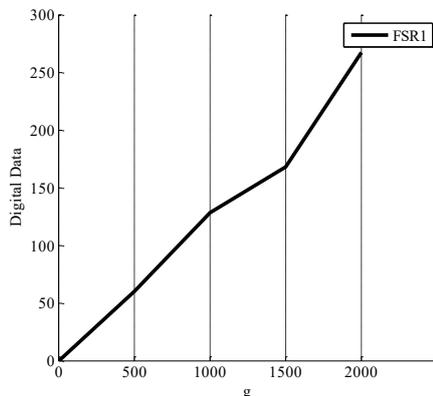
圖(14) ZMP2

為讀取腳底壓力感測器的系統圖，本模組輸入為壓力感測器，輸出為零力矩點，壓力感測器會分別使用微處理器處理電壓感測，線性內插最後計算零力矩點。

平台共使用 8 個壓力感測器，但是各個壓力感測器在受到相同壓力時電阻值會有所不同，再加上壓力感測器的數值會因為微處理器的腳位使用不同導致各個壓力感測器數值的差異，所以本論文使用線性內插公式進行校正，將 8 個壓力感測器的數值轉成公克數用以統一單位，將分別量測單一壓力感測器在受到 500 公克、1000 公克、1500 公克與 2000 公克重量時的數值記錄下來，使用式 2.12

$$w = w_a + (w_b - w_a) \frac{D - D_a}{D_b - D_a} \quad 2.12$$

$w_a$ 與 $w_b$ 為量測重量， $D_a$ 與 $D_b$ 為量測到的數位訊號， $D$ 是現在的感測器數值， $w$ 為量測出的重量，下圖為線性內插方法的壓力感測器校正示意圖。



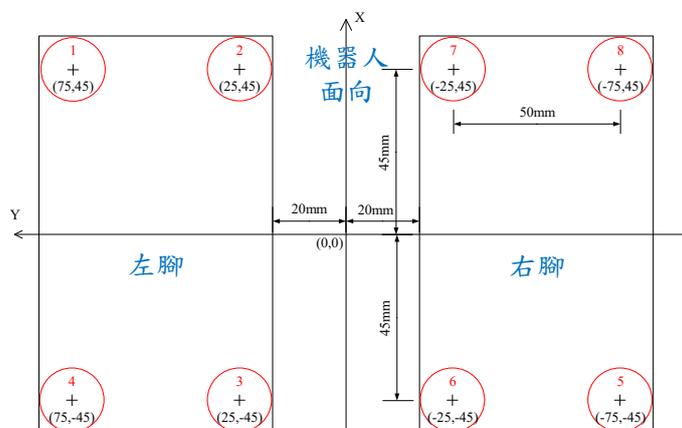
圖(15) 壓力感測器校正示意圖

零力矩點的計算使用權重法，首先知道本論文的實驗平台小型人形機器人的雙腳尺寸，根據雙腳間距與壓力感測器擺放位置得知各個壓力感測器的權重值，如圖(16)所示，利用雙腳中心點做為坐標原點，單位為公釐 mm (millimeter)，2.13 式和 2.14 式為計算公式。

$$zmp^x = \frac{\sum_{i=1}^8 x_i \cdot w_i}{\sum_{i=1}^8 w_i} \quad 2.13$$

$$zmp^y = \frac{\sum_{i=1}^8 y_i \cdot w_i}{\sum_{i=1}^8 w_i} \quad 2.14$$

$w_i$ 為 8 個壓力感測器的所量測的重量， $x_i$ 與 $y_i$ 為各個壓力感測器的座標位置，計算後得到 x 軸與 y 軸的零力矩點位置 $zmp^x$ 與 $zmp^y$ 。

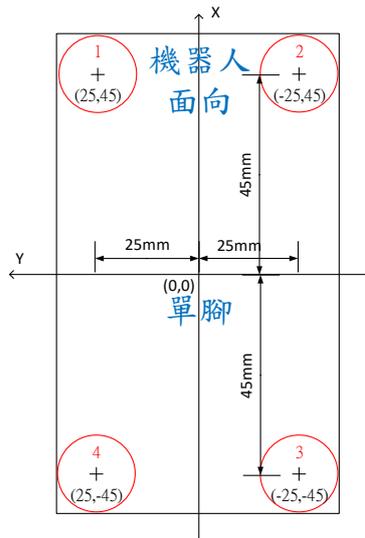


圖(16) 機器人雙足座標系示意圖

同樣的方法計算單腳的零力矩點，如圖 十七所示，坐標原點為腳掌正中心，2.15 式和 2.16 式為左腳計算公式，2.17 式和 2.18 式為右腳計算公式，用權重法計算左腳與右腳的零力矩點。

$$zmp_L^x = \frac{\sum_{i=1}^4 x_i \cdot w_i}{\sum_{i=1}^4 w_i} \quad 2.15$$

$$zmp_L^y = \frac{\sum_{i=1}^4 y_i \cdot w_i}{\sum_{i=1}^4 w_i} \quad 2.16$$



圖(17) 機器人單足座標系示意圖

$$zmp_R^x = \frac{\sum_{i=1}^4 x_i \cdot w_i}{\sum_{i=1}^4 w_i} \quad 2.17$$

$$zmp_R^y = \frac{\sum_{i=1}^4 y_i \cdot w_i}{\sum_{i=1}^4 w_i} \quad 2.18$$

## 五、人形機器人智慧平衡之建構

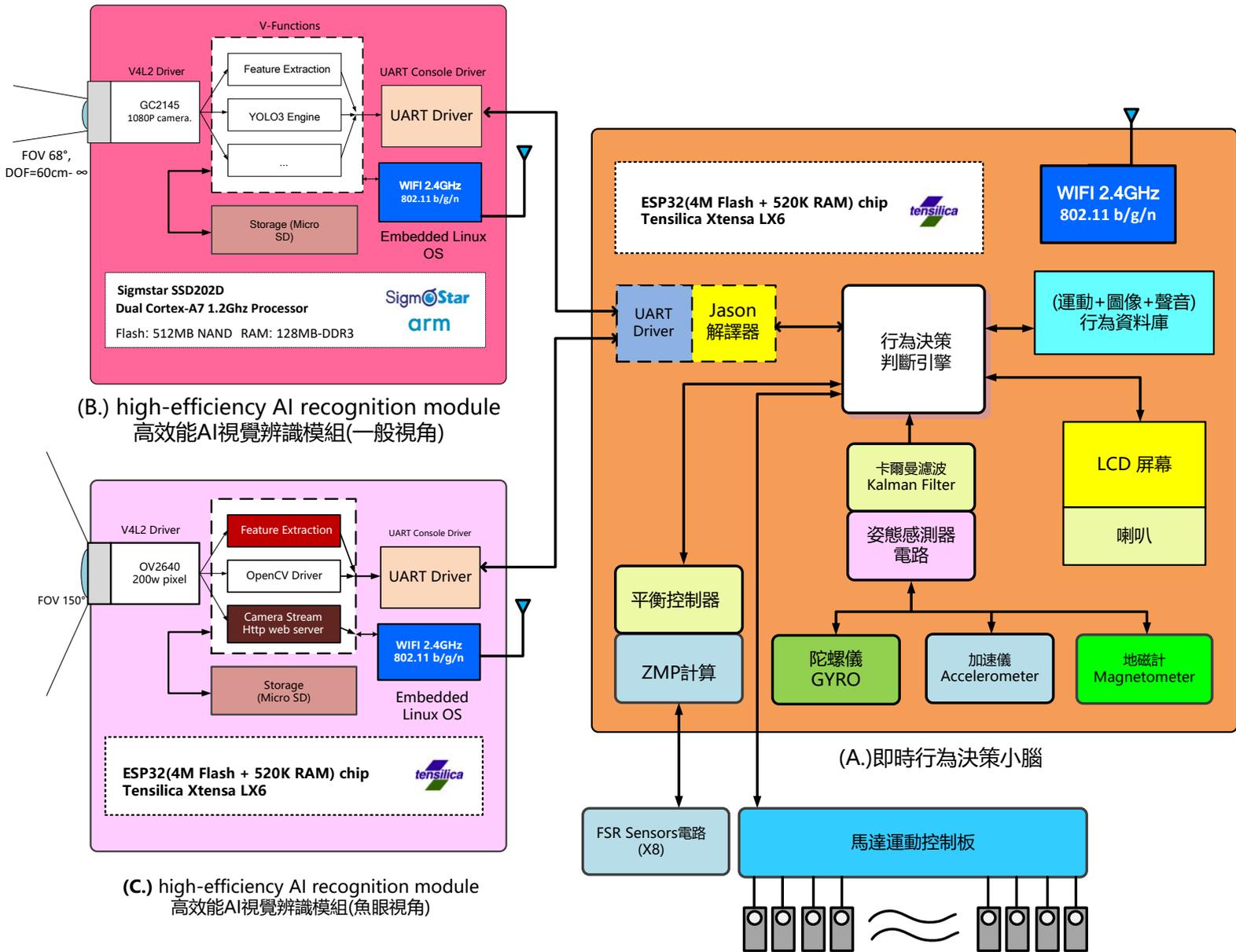
猶如前述機器人分散式網路模型，為了實現系統，我們建構以下三個主要單元軟硬體整合於機器人端。(A.)即時行為決策小腦單元、(B.) High-efficiency AI recognition module (高效能 AI 視覺辨識模組單元[一般視角])、(C.) High-efficiency AI recognition module (高效能 AI 視覺辨識模組單元[魚眼視角])。藉由前言所述之嵌入式網路技術的整合，我們使用機器人分散式網路模型配合 LWIP[20]為基礎的嵌入式網路技術三個模組皆具備嵌入式 TCPIP 並藉由無線網路 2.4GHz 802.11 b/g/n 做網路通透整合。機器人部分系統端三個模組採用 UART Driver 溝通做通訊並運用 JSON 模式做為溝通協議做參數軟體溝通與問詢答案的取得，此系統運用 JSON 協議方便獨立調教以及錯誤判斷，系統可單獨模組化設計。其中相關細節的簡要描述如下：

- (一) 即時行為決策小腦單元：研究者設計機器人系統具備快速決策與彙整其他嵌入式系統的通訊系統軟體之資訊，在此使用 ESP32(4M Flash + 520K RAM) chip 為核心，基本上它是 Tensilica Xtensa LX6 的晶片架構[21]，超低功耗雙核心 32 位微型 MCU(Xtensa LX)系列處理器具有很強的可重構性和可拓展性。Xtensa 處理器具有不同於其它傳統式的嵌入式處理器核心[22]，改變了單芯片系統的設計規則。Xtensa 處理器，其設計方式彈性大，功效高採用硬件可重構加軟件編程的方式，既提升了處理器的運算性能又通過軟件便於實現控制。

其中軟體架構內規劃了幾個軟體模組包含

1. ZMP 計算與平衡控制器，依照前述將零力矩點 ZMP 取得的資訊轉換至實體座標位置提供給平衡控制器補償。
2. 姿態感測系統電路，整合了陀螺儀(GYO)，加速儀(Accelerometer)，用於姿態感測偵測，可以量測系統三軸旋轉角速度角加速度，以及三軸之即時加速度量質以及地面的 G 值方向和量值。即時姿態回授會經由卡爾曼(Kalman)濾波器[23]做修正補償。進一步並提供給行為決策引擎做為決策使用以及即時馬達運動控制單元的即時運算與執行。

3. 行為決策判斷引擎會依照兩個 AI 視覺辨識模組取得的視覺辨識結果理解環境的敵方的姿態位置，並配合系統的身體姿態以及穩定性即時狀態做判斷，並選擇運動圖像聲音行為資料庫內之內容作即時派送到各馬達作姿態解算以及聲音和 LCD 屏幕的撥放。



圖(18) 機器人分散式網路模型

(二) High-efficiency AI recognition module (高效能 AI 視覺辨識模組單元[一般視角]) :

高效能 AI 視覺辨識模組單元分為兩種，一種為一般視角，另一種為魚眼視角。一般視角內涵影像視覺引擎，為了提高系統效能系統使用 Sigmstar SSD202D 的核心晶片，SSD202D 是一顆雙核心 Cortex-A7 1.2 Ghz 的 CPU 配合內建的 512M Nand flash 以及 128 MB DDR3。核心內建 Embedded Linux 為基礎。並藉由 V4L2 Driver[24]驅動程式整合了 GC2145 視覺模組。GC2145 本身是一個具備 1080P 的 Camera，藉由 68 度視角成為機器人視覺，影像接近正常人視角較少扭曲。Embedded Linux 系統內含有 YOLO3 引擎視覺辨識引擎，用來做為機器人的即時追蹤和影像辨識運算。系統上電開機後即載入演算法以及辨識模型和以訓練好之參數。即可即時追蹤敵人以及得知敵方機器人位置和大小並匡列計算出其可能性與百分比。其結果並運用 UART Driver 以 Jason 格式傳至外部提供給即時行為決策小腦做為參考。由於深度學習演算法本身計算需要大量計算，因此演算法大量消耗 CPU 的計算力，核心本身很耗熱，並配備了一個散熱風散提供散熱。

(三) High-efficiency AI recognition module (高效能 AI 視覺辨識模組單元[魚眼視角]) :

高效能 AI 視覺辨識模組單元的魚眼視角模組，目的是為了讓系統不需要運用轉向機構就能夠目視到大範圍面積的角度。核心依然配合 ESP32(4M Flash + 520K RAM) chip 為核心。魚眼視角核心架構配備使用 OV2640 200W pixel 的感測模組以及 150 度角一般魚眼鏡片單元使用，內建使用 opencv driver [25]整合以及網路影像伺服器功能，建構在嵌入式 Linux 內部於底層運行，得以可以使用遠端網路取得機器人本體的影像視覺。藉由 150 度視角成為機器人視覺，影像魚眼鏡頭特性會有些許扭曲，但是視角可達 150 度廣視角。機器人不須轉向機構即可以看到很廣的環境。在使用上可以提供輔助操作者作戰的即時影像情資。系統亦可以運用 UART Driver 以 Jason 格式與外部行為決策小腦做為溝通更新參數。內近的 Http web server 即時提供外部影像更新與視覺延伸計算使用。其相關參數請參閱以下表:

| 即時行為決策小腦單元   |                                                                     |
|--------------|---------------------------------------------------------------------|
| Resources    | Parameter                                                           |
| ESP32        | 240MHz dual core, 600 DMIPS, 520KB SRAM, Wi-Fi, dual mode Bluetooth |
| Flash Memory | 16MB (old: 4MB)                                                     |
| Power Input  | 5V @ 500mA                                                          |
| Port         | TypeC x 1, GROVE(I2C+I/O+UART) x 1                                  |
| IPS Screen   | 2 inch, 320x240 Colorful TFT LCD, ILI9342C, max brightness 853nit   |
| Speaker      | 1W-0928                                                             |
| Button       | Custom button x 3                                                   |
| MEMS Sensor  | BMM150 + MPU6886                                                    |
| Antenna      | 2.4G 3D Antenna                                                     |

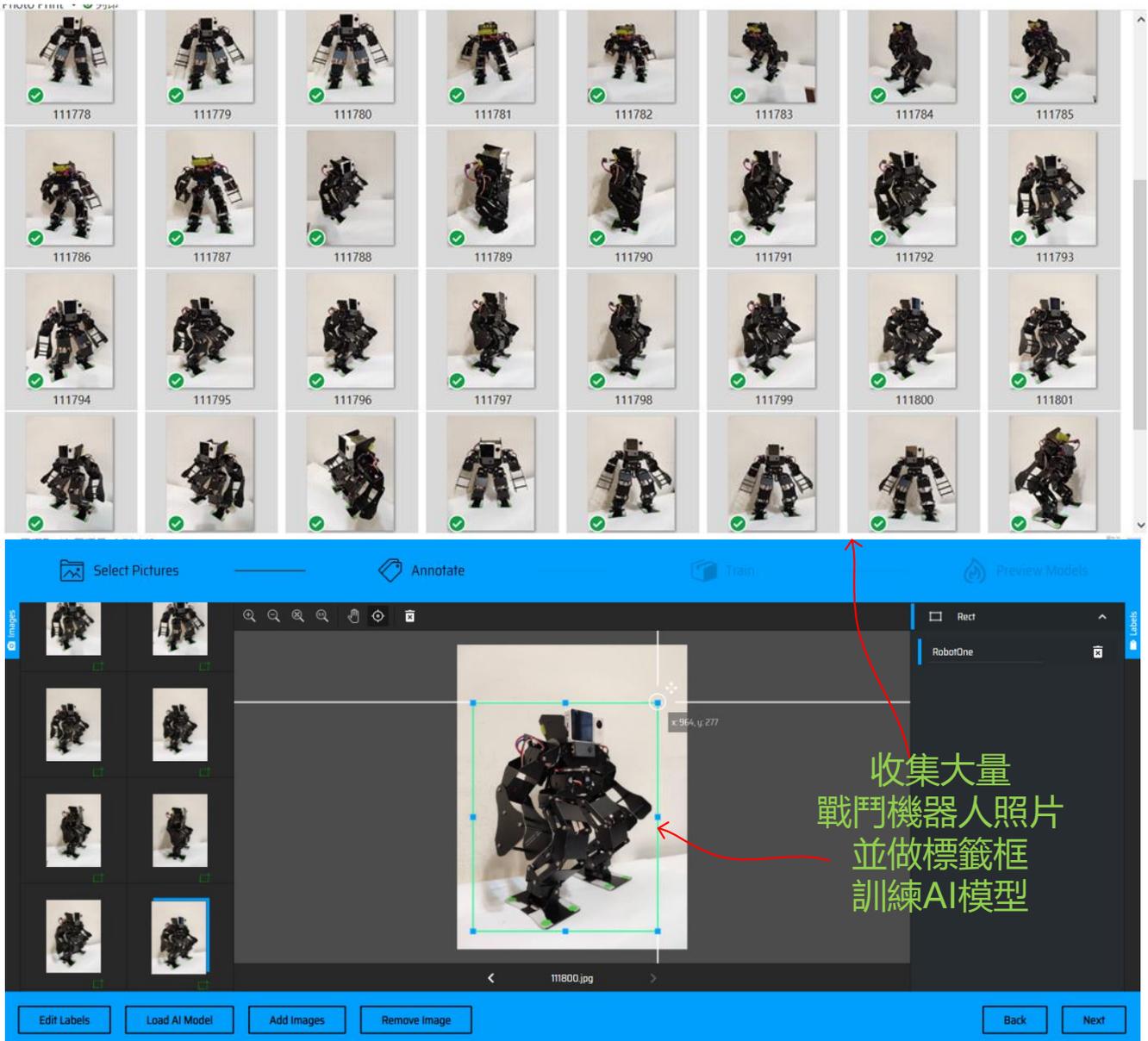
| High-efficiency AI recognition module (高效能 AI 視覺辨識模組單元[一般視角]) |                                                                      |
|---------------------------------------------------------------|----------------------------------------------------------------------|
| Resources                                                     | Parameter                                                            |
| Sigmstar SSD202D                                              | Dual Cortex-A7 1.2Ghz Processor                                      |
| Flash                                                         | 512MB NAND                                                           |
| RAM                                                           | 128MB-DDR3                                                           |
| Camera                                                        | GC2145 1080P Colored Sensor                                          |
| Lens                                                          | FOV 68° , DOF= 60cm- ∞                                               |
| Power Input                                                   | 5V @ 500mA                                                           |
| Peripherals                                                   | TypeC x1, UART x1,<br>TFCard x1, Button x1,<br>Microphone x1, Fan x1 |
| Indicator light                                               | Red, White                                                           |
| WiFi                                                          | 150Mbps 2.4GHz 802.11 b/g/n                                          |

High-efficiency AI recognition module  
(高效能 AI 視覺辨識模組單元[魚眼視角]):

| Resources                          | Parameter                                                                        |
|------------------------------------|----------------------------------------------------------------------------------|
| ESP32                              | ESP32-WROVER (PCB Antenna)<br>Tensilica Xtensa LX6                               |
| Flash                              | 4M                                                                               |
| RAM                                | 4MB                                                                              |
| Image Sensor                       | OV2640                                                                           |
| Maximum resolution                 | 200w pixel                                                                       |
| Output format                      | YUV(422/420)/YCbCr422,RGB565/555,8-bit<br>compressed data,8-/10-bit Raw RGB data |
| Maximum image<br>transmission rate | UXGA/SXGA: 15fps, SVGA: 30fps, CIF: 60fps                                        |
| FOV                                | 150°                                                                             |
| CCD Size                           | 1/4 inch                                                                         |
| WiFi                               | 2.4GHz 802.11 b/g/n                                                              |

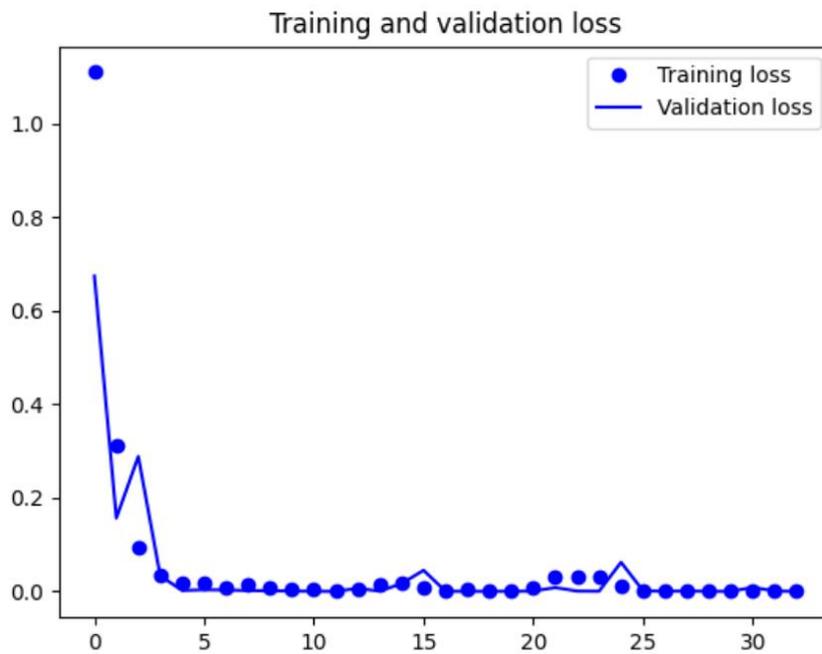
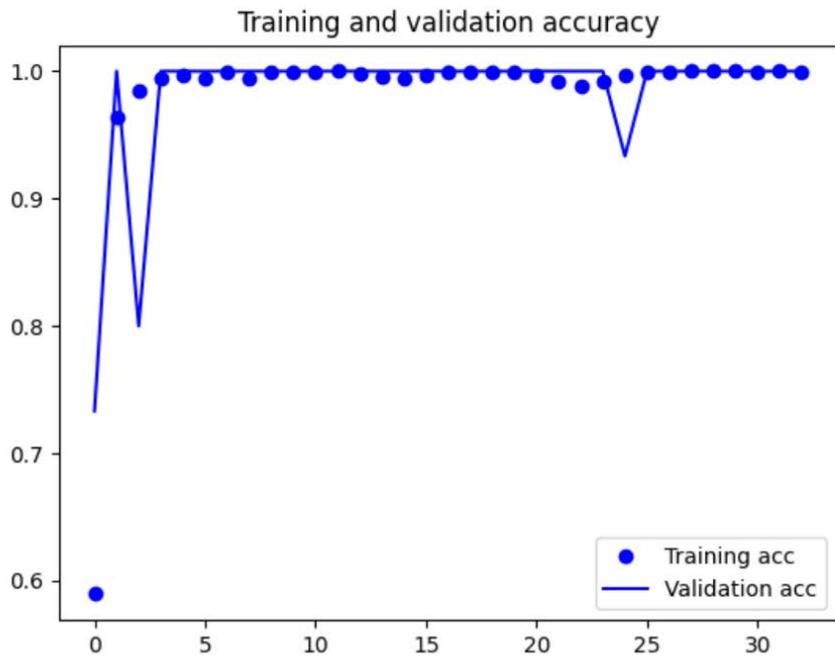
## 六、深度學習 YOLO 影像演算法

通過 AI 模型訓練，建自訂識別模型。使用攝像設備拍攝圖片素材並做標記與標籤框保存至電腦，使用 V-Training 訓練 Yolo 模型與參數，初期以手動一張一張拍攝以及人工標籤。其訓練結果統計其 Training and validation accuracy 以及 Training and validation loss 如下圖(19)所示。



圖(19) V-Training

Model: Classification MobileNetV1 Alpha: 0.7 Depth: 1

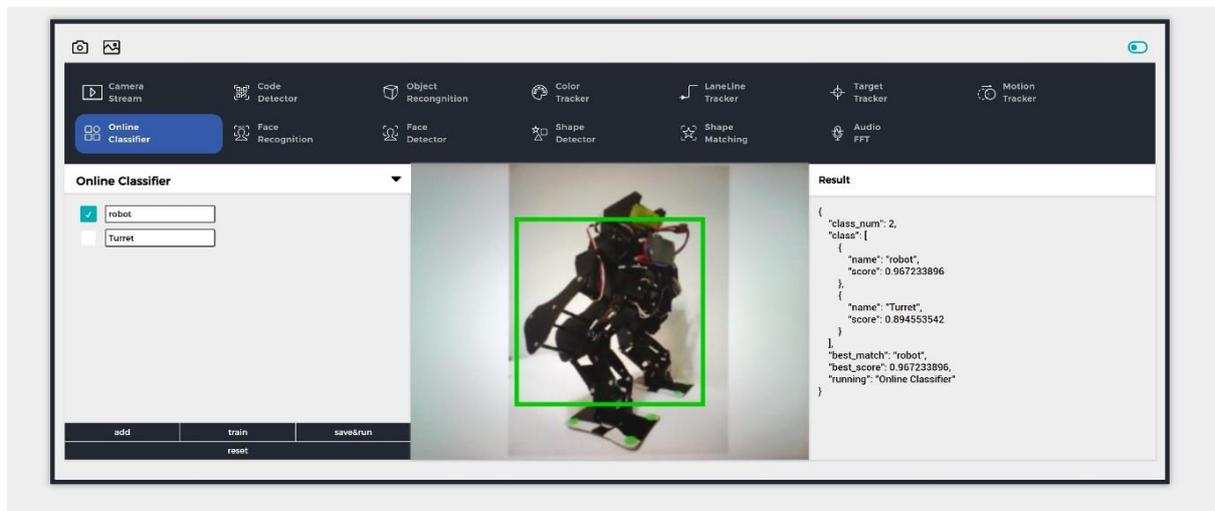


圖(20) Training and validation accuracy

Training and validation loss

## 七、視覺辨識引擎運行狀態

藉由嵌入式網路連線至機器人之影像系統，圖(21)顯示其視覺辨識引擎運行狀態。運行



圖(21) 辨識引擎運行狀態

的即時輸出如前所述是使用 JSON 格式和嵌入式網路和各晶片以及外部溝通調教。以下描述相關的介面功能與應用範例:

### (一) 目標追蹤

回傳 JSON :

```
{  
  "FUNC": "TARGETTRACKERV1.0",  
  "x": 282,  
  "y": 165,  
  "w": 13,  
  "h": 15  
}
```

設置 JSON :

```
{  
  "TARGETTRACKER": " V1.0",  
  "x": 282, //xywh 均不可缺省  
  "y": 165,  
  "w": 13,  
  "h": 15  
}
```

框出目標在畫面中的位置，即可自動追蹤框中的目標。目標最好具有顯著的顏色特徵。



圖(22) 辨識引擎運行狀態

## (二) 色塊識別

回傳 JSON :

```
{  
  "FUNC": "COLORTRACKERV1.1",  
  "TOTAL": 3, //邊界框數量  
  "0": {  
    "x": 45,  
    "y": 18,  
    "w": 126,  
    "h": 72,  
    "area": 342 //該邊界框內變化圖元的數量  
  },  
}
```

```
"1":{  
  "x":0,  
  "y":169,  
  "w":130,  
  "h":24,  
  "area":173  
},  
"2":{  
  "x":39,  
  "y":204,  
  "w":276,  
  "h":34,  
  "area":141  
}  
}
```

設置 JSON :

```
{
"COLORTRACKER":1.1,//功能標記，不可缺省
"thr_w":20,//可缺省邊界框寬閾值，[3,200]
"thr_h":20,//可缺省邊界框長閾值，[3,200]
"stepx":1,//可缺省X 掃描間隔，[0,40]，設置為 0 則關閉邊界框檢測
"stepy":2,//可缺省Y 掃描間隔，[0,40]，設置為 0 則關閉邊界框檢測
"merge":10,//可缺省邊界框合併閾值，[0,40]
"Lmin":0,//可缺省L 閾值下限[0,100]
"Lmax":0,//可缺省L 閾值上限[0,100]
"Amin":0,//可缺省A 閾值下限[0,255]
"Amax":0,//可缺省A 閾值上限[0,255]
"Bmin":0,//可缺省B 閾值下限[0,255]
"Bmax":0,//可缺省B 閾值上限[0,255]
"viewmode":0,//可缺省圖像顯示模式，0 為原始圖像，1 為經過 LAB 過濾後的圖像
"range":20,//可缺省自動確定 LAB 閾值時的容限，閾值為目標顏色的+-range
"roi_x":150,//可缺省，但 ROI 必須四項同時出現，自動讀取 ROI 內的顏色確定 LAB 閾值
"roi_y":110,
"roi_w":20,
"roi_h":20
}
```

area：該顏色的圖元數量

LAB 顏色閾值：是 LAB 顏色空間的顏色值，在該範圍外的顏色將會被過濾。這個值的選取需要專用的閾值編輯器實現，後續我會寫一個小工具。

檢測步進，長寬閾值，合併閾值與運動目標追蹤中所述一致。

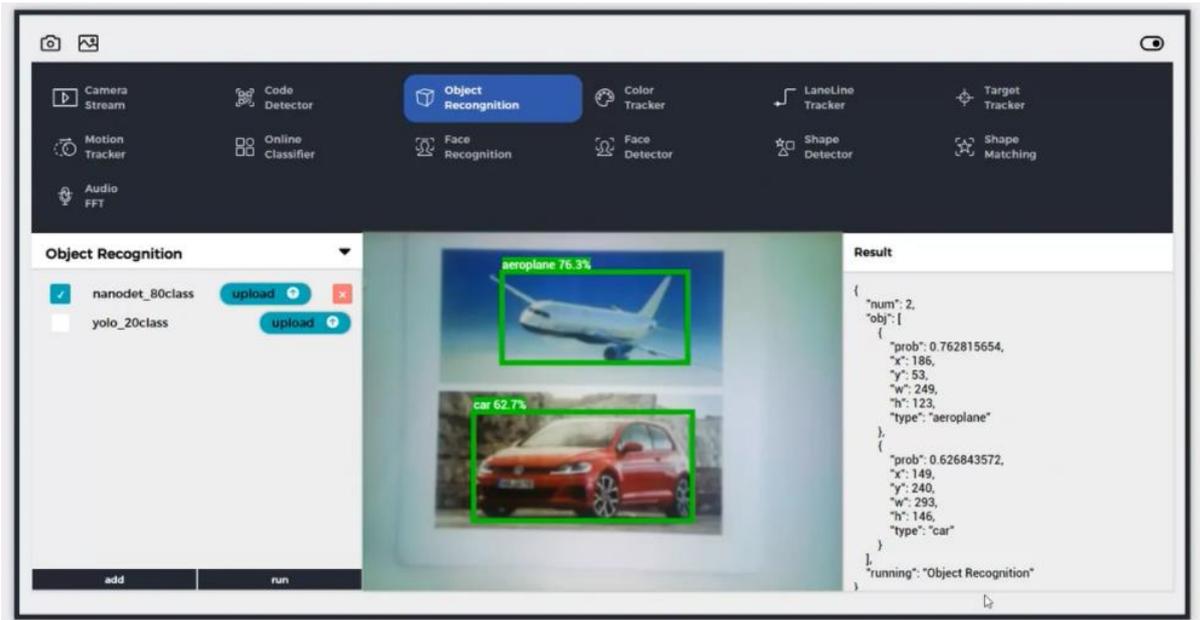
### (三) Object Recognition

```
//選擇參數 “yolo_20class” 切換至該功能
{
  "function": "Object Recognition",
  "args": ["yolo_20class"]
}

//選擇參數 “nanodet_80class” 切換至該功能
{
  "function": "Object Recognition",
  "args": ["nanodet_80class"]
}
```

樣例輸出

```
{
  "num": 1,
  "obj": [
    {
      "prob": 0.938137174,
      "x": 179,
      "y": 186,
      "w": 330,
      "h": 273,
      "type": "person"
    }
  ],
  "running": "Object Recognition"
}
```



圖(23) Object Recognition

## 參、研究結果與討論

本研究主題運用嵌入式網路技術實現邊緣運算技術之人工智能機器人代理人架構，進行人型格鬥用機器人的初期開發研究。格鬥機器人做為場景需求使用設定目標，目前已提供未來一個彈性運算的開發平台的基礎研究。本研究實現以下：

- 一、 本研究運用嵌入式網路技術整合系統，整合分散式嵌入式系統與通訊協議建構機器人分散式網路模型。此架構的確是讓邊緣運算的擴充以及整合上達到傳統機器人計算平台的中央集中式的拓樸架構好很多且容易配置。進一步對於整合過程的配置和彈性與錯誤除錯十分容易且方便。特別是人形機器人的四肢分散於本體馬達控制與感測器，以傳統集中中心的設計方式，機器人的線纜將會過重機器人整體結構上將不易整合，重量。中央 cpu 也必須要運算十分強且耗能才能整合過集中。分散式架構的確實現了改善以上的問題。
- 二、 初步結合深度學習 YOLO3 影像演算法之辨識即時辨識發現對手機器人的位置，以提供系統平台未來計算的基礎。初步實驗對於演算法訓練結果統計其 Training and validation accuracy 以及 Training and validation loss 收斂成果很好。未來在實戰場合需要多加測試驗證增加其可行性。
- 三、 建立 AI 深度學習即時辨識在地演算猶如小腦即時反應。亦能將網路串流傳輸至遠端的計算平台作為分散演算法擴充。
- 四、 進一步在足部平衡上，本研究使用 FSR(Force sensing resistor)作為人形機器人智慧平衡基礎，藉由圖形識別做為預測辨識以及智慧姿態 ZMP 控制技術作為攻防策略判斷。
- 五、 系統運用 JSON 統合提供給核心平台的整合計算，間接可以提供一個外來開發嵌入式系統模組的溝通拓展架構。

六、 整體系統藉由 AI 晶片與嵌入式系統網路作為整合。透過網路即時傳輸環境資訊與指令，使機器人可以知道高層的指令目的資訊。值得一提的是本系統網路設計建構依照仿生哺乳類動物的分層式架構。並初步建立一個高階人形機器人智慧網路平台未來作為開發高階機器人之控制與研究探討，初步結果驗證系統架構可行，未來針對不同性質人工智能研究建構此基礎下做更深入探討。

## 肆、結論與應用

一直以來，仿生機器人格鬥是很多科幻動漫迷的夢想，過去往往在設計此系統整合上相當不容易，要設計此人工智能代理機器人非常困難。奇技術瓶頸在於人形機器人動力結構系統與仿生機構的設計，必須考慮附載大扭矩以及平行性的問題，且系統結構複雜，肢體分散於身體部位，馬達控制制動元件網路系統複雜即時控制以及感測回授運算複雜整合不易。由於控制技術複雜且多樣，一般運算平台運算越強大功耗耗能就越大能量電池就越高成本也越高。系統設計要實現小體積、低工耗、可彈性泛使用是相當大的挑戰。

本研究系統藉由 AI 晶片與嵌入式系統網路作為整合。透過網路即時傳輸環境資訊與指令，使機器人可以知道高層的指令目的資訊。值得一提的是本系統網路設計建構依照仿生哺乳類動物的分層式架構。並初步建立一個高階人形機器人智慧網路平台未來作為開發高階機器人之控制與研究探討，初步結果驗證系統架構可行，未來針對不同性質人工智能研究建構此基礎下做更深入探討。

## 伍、參考文獻

- [1.] ROBO-ONE Official Website, <https://www.robo-one.com>
- [2.] Respawn Entertainment, “Titanfall 2,” [Online]. Available: <https://www.youtube.com/watch?v=YRJH4N-sWeI>
- [3.] Wu, Li-Wei and Hu, Jwu-Sheng, 2006, “Distributed Embedded Control Platform for Robots Using Real-Time Ethernet: Design Concept, System Architecture, and Implementation,” Vol. 15, No. 2, pp. 108-121, June 2008, IEEE Robotics and Automation Magazine (SCI/EI).
- [4.] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, “The intelligent ASIMO: System overview and integration,” *Intell. Robots Syst.*, vol. 3, pp. 2478 – 2483, 2002.
- [5.] T. Ishida, Y. Kuroki, J. Yamaguchi, M. Fujita, and T. T. Doi, “Motion entertainment by a small

- humanoid robot based on OPEN-R,” in Proc. 2001 Conf. Intelligent Robots and Systems, vol. 2, pp. 1079 – 1086.
- [6.] T. Makimoto and T. T. Doi, “Chip technologies for entertainment robots—Present and future,” in Tech. Dig. Electron Devices Meeting, IEDM’ 02, 2002, pp. 9 – 16.
- [7.] B. H. Lee, “Embedded Internet systems: Poised for takeoff,” IEEE Internet Comput., vol. 2, pp. 24 – 29, 1998.
- [8.] R. E. Filman, “Embedded Internet systems come home,” IEEE Internet Comput., vol. 5, no. 1, pp. 52 – 53, 2001.
- [9.] J. Axelson, Embedded Ethernet and Internet Complete: Designing and Programming Small Devices for Networking. Madison, WI: Lakeview Research, 2003.
- [10.] D. L. Sancho-Pradel, S. R. Jones, and R. M. Goodall, “System on programmable chip for real-time control implementations,” in Proc. 2002 IEEE Int. Conf. Field-Programmable Technology, pp. 276 – 283.
- [11.] S. Perez and J. Vila, “Building distributed embedded systems with RTLinux-GPL,” in Proc. 2003 IEEE Conf. Emerging Technologies and Factory Automation, 2003, vol. 1, pp. 161 – 168.
- [12.] J. S. Jwo, S. S. Chang, Y. C. Chen, and D. F. Hsu, “A distributed environment for hypercube computing,” in Proc. 2nd Aizu Int. Symp. Parallel Algorithms/Architecture Synthesis, 1997, pp. 256 – 263.
- [13.] J. Wang, (2002). Soft real-time switched ethernet: best-effort packet scheduling algorithm, implementation, and feasibility analysis, M.S. thesis. Dept. Electr. Comput. Eng., Faculty of the Virginia Polytechnic Institute and State Univ. [Online]. Available: <https://vtechworks.lib.vt.edu/handle/10919/35277>
- [14.] K. Schrodinger, J. Stimma, and M. Mauthe, “A fully integrated CMOS receiver front-end for optic gigabit Ethernet,” IEEE J. Solid State Circ., vol. 37, no. 7, pp. 874 – 880, 2002.

- [15.] B. Blaner, D. Czenkusch, R. Devins, and S. Stever, “An embedded PowerPCTM SOC for test and measurement applications,” in Proc. 13th Annu. IEEE Int. ASIC/SOC Conf., 2000, pp. 204 – 208.
- [16.] J. Bilek and I. P. Ruzicka, “Evolutionary trends of embedded systems,” in Proc. 2003 IEEE Int. Conf. Industrial Technology, vol. 2. pp. 901 – 905.
- [17.] Philippe Sardain and Guy Bessonnet., “Forces Acting on a Biped Robot, Center of Pressure—Zero Moment Point.,” IEEE Trans. Systems, Man, and Cybernetics—Part A. Vol. 34, No. 5, pp. 630 – 637, 2004.
- [18.] 陳玉琨, 湯曉鷗 (主編) (2018)。人工智慧基礎 (高中版)。出版商：華東師範大學出版社。
- [19.] 王彥翔, “小型人形機器人之 多感測器步態平衡系統”, 淡江電機工程學系 碩士論文, 2015。
- [20.] Adam Dunkels, “Design and Implementation of the lwIP TCP/IP Stack,” Swedish Institute of Computer Science, Feb. 2001. [Online]. Available:  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.109.1795&rep=rep1&type=pdf>
- [21.] Cadence Design System, “Xtensa LX6 Customizable DPU” , [Online]. Available:  
[https://mirrobo.ru/wp-content/uploads/2016/11/Cadence\\_Tensillica\\_Xtensa\\_LX6\\_ds.pdf](https://mirrobo.ru/wp-content/uploads/2016/11/Cadence_Tensillica_Xtensa_LX6_ds.pdf)
- [22.] CSDN, “XTENSA 處理器介紹,” [Online]. Available:  
<https://blog.csdn.net/whatday/article/details/87268727>
- [23.] Kalman, R. E., “A New Approach to Linear Filtering and Prediction Problems,” Transactions of the ASME - Journal of Basic Engineering Vol. 82: pp. 35-45 (1960)
- [24.] Wikipedia, “Video4Linux,” [Online]. Available: <https://zh.wikipedia.org/wiki/Video4Linux>
- [25.] OpenCV team, “OpenCV: Home,” [Online]. Available: <https://opencv.org/>

## 【評語】 190003

整個研究題目牽涉到很多重要的技術如邊緣運算、機器人、影像辨識等，是很重要很實用的研究題目，但有進行科學實驗的部分只是用現成影像辨認套件來進行訓練和測試且深度不足。此自發性題目的構想值得嘉許。英文口說能力可多加強。