Hello! I’m Yu-Cheng, and I will introduce this part.

YOLO is a famous object detection tool, and we chose it for our project. YOLO now has version 11, but version 5 works better for different devices. It is also stable and easy to set up, so we used YOLOv5 for our system.

Next is OpenPose. OpenPose was made by CMU in 2017. It connects body parts to make a pose skeleton using PAFs. It can detect more keypoints and still run fast. We used the PyTorch version because it works well on many devices and is easy for beginners.

Now, let’s talk about ANN. ANN is a simple neural network for decision-making. Some people think it’s too basic, but its simplicity helps reduce system load, which is very important for embedded systems.

That’s all for the technical details. Let’s move on to something more interesting!

In our system, we worked hard on AI. For example, YOLOv5 can only use one weight model. If we add new categories, we must retrain all objects. To fix this, we created a feature to stack weight models. We also improved other YOLO versions like v8, v9, and v11. We made an algorithm to compare confidence scores to avoid conflicts (e.g., the same object detected twice).

About OpenPose, YOLO has a pose model, but it’s not as good as OpenPose for real-time, multi-person detection. We used OpenPose to detect actions like raising hands or sitting in wheelchairs. Raising hands shows someone wants to get on a vehicle, and wheelchair detection tells the driver to lower the ramp. We also added an automated ramp feature with servo motors and relays.

Lastly, ANN helps turn poses into gestures. A pose shows the state of an action, but a gesture is a defined movement. ANN combines YOLO and OpenPose to predict gestures.

Our system also includes text and voice alerts. Let’s watch the result in a video.

The first video shows how our trained model detects road signs. We used a video from the internet, but the weights were our own.

The second video shows the complete system, including YOLO, OpenPose, and voice/text alerts. The person in the video is our model.

How does it look? Feel free to ask questions!

哈囉！我是育丞，現在由我來為各位介紹。

YOLO 是舉世聞名的物件偵測技術指標，成為我們開發過程中的不二之選。現今的 YOLO 已經開發至版本 11，但該版本的跨裝置整合能力未能超越 YOLOv5 的全面性，因此我們選擇使用穩定且建構環境簡單的 YOLOv5 作為實作基礎，方便任何使用者快速上手。

接著來說說 OpenPose。OpenPose 是由 2017 年 CMU 發布的一項技術，透過 PAFs 將多人的肢體關係連結起來，生成姿態骨架。不僅如此，該技術在提供最多關鍵點的同時，仍能保持優秀的 FPS 表現。我們採用了基於 PyTorch 的版本，便於多種裝置應用，也更加適合初學者操作與維護。

再來談談 ANN。它是一個基礎神經網路，主要用於決策分析。有些人會覺得結構過於簡單，但正因為它簡單，我們才選用它。我們希望盡可能減少運行負載，確保嵌入式系統能夠順利運行。

說完這些技術部分，現在來談些有趣的內容吧！

在這個系統中，除了前幾位組員提到的感測器功能，我們在 AI 方面也投入了許多努力。例如，YOLOv5 僅支持一個權重模型的運行，但若新增類別，則需重新訓練所有物件。針對這一點，我們開發了可堆疊權重模型的功能。此外，我們對 YOLO 的其他版本（如 v8、v9 和 v11）也進行了相關貢獻。為了解決模型之間的衝突（例如：同一目標被不同模型重複偵測），我們設計了一套基於信心度的比較演算法進行處理。

至於 OpenPose，大家可能會好奇為什麼不用 YOLO 的 Pose 模型。其實，YOLO 雖有提供 Pose 功能，但在即時多人處理上不如 OpenPose 表現優異。因此，我們選擇 OpenPose 來偵測例如舉手或乘坐輪椅等行為。其中，舉手用來判斷是否有搭車需求，而輪椅偵測則提醒司機啟用坡道。我們在系統中也整合了自動化坡道的開啟與關閉功能，該部分運用了伺服馬達與繼電器技術。

最後談談 ANN，它將姿態轉換為姿勢。簡單說，姿態是某個行為的狀態描述，無法直接具體定義為某種動作。而姿勢則是在姿態基礎上定義出的動作。因此，我們使用 ANN 將 YOLO 和 OpenPose 整合後進行預測。

最後，我們還實現了文字及語音提醒功能，並加入選配功能。接下來，我們直接進入成果展示影片。

第一部分展示我們訓練的權重模型在道路標誌偵測上的表現，影片素材來自網路，但模型則是我們自主訓練的。

第二部分展示從基礎整合到加入 YOLO 和 OpenPose，再到語音與文字提示功能的完整系統影片，影片中模擬的場景由我們找來的模特兒參與。

效果是不是還不錯呢？歡迎大家提問喔！