

基於深度學習並整合頭部姿態與視線估測之輕量與全功能雙模式駕駛行為監控系統

組員：馬順康、張耀銘

摘要

本專題旨在設計一套可以依據使用者硬體資源選擇「輕量模式」或「全功能模式」的雙模式駕駛行為監控系統。系統以單顆鏡頭作為影像輸入，結合多項行為辨識功能，將即時偵測到的行為顯示於螢幕，以時刻提醒駕駛潛在的危險並提升駕駛的安全意識，預防交通事故發生。在輕量模式下，系統透過 MediaPipe 偵測面部特徵點，並結合 solvePnP 演算法進行頭部姿態估測，以實現判斷駕駛是否有頭部偏移、疲勞與打哈欠等功能；全功能模式中，系統使用 YOLOv7 進行物件偵測(包含人臉、手機與香菸)，並結合 6DRepNet 估測頭部姿態，與 L2CS-Net 推論視線方向，進一步增加偵測視線偏移、使用手機與抽菸等行為。

輕量化頭部姿態估測演算法設計

1. MediaPipe Face Mesh 偵測 3D 臉部特徵點座標。
2. 臉部特徵點座標輸入至 solvePnP 演算法，透過矩陣運算，求解物空間座標轉換到像空間座標所需的旋轉向量與平移向量。

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

World coordinate system
Camera coordinate system
Image coordinate system

3. Rodrigues 演算法將旋轉向量轉為旋轉矩陣。
4. 由旋轉矩陣求解歐拉角的數學算法，計算頭部在三維座標的偏移角度 (Pitch, Yaw, Roll)。

```

if (R31 ≠ ±1)
    θ1 = -asin(R31)
    θ2 = π - θ1
    ψ1 = atan2( R32/cosθ1, R33/cosθ1 )
    ψ2 = atan2( R32/cosθ2, R33/cosθ2 )
    φ1 = atan2( R21/cosθ1, R11/cosθ1 )
    φ2 = atan2( R21/cosθ2, R11/cosθ2 )
else
    φ = anything; can set to 0
    if (R31 = -1)
        θ = π/2
        ψ = φ + atan2(R12, R13)
    else
        θ = -π/2
        ψ = -φ + atan2(-R12, -R13)
    end if
end if
    
```

5. 角度存在多組解時，加入以下條件判斷式以選擇適當的解，確保估測角度的合理性。

roll = roll_2 if abs(roll_1) > π/2 else roll_1

yaw = yaw_2 if abs(yaw_1) > π/2 else yaw_1

pitch = -pitch_2 if abs(pitch_1) > π/2 else -pitch_1

物件辨識模型訓練

● 資料集標註

專題使用的訓練資料集來自實驗室所拍攝之駕駛影像，並以 LabelImg 作為標註工具，自行定義九個標註類別：Face、Left_eye、Left_eye_Iris、Right_eye、Right_eye_Iris、Nose、Mouth、Cigarette、Phone，讓 YOLO 模型能夠辨識人臉、手機與香菸等物件。

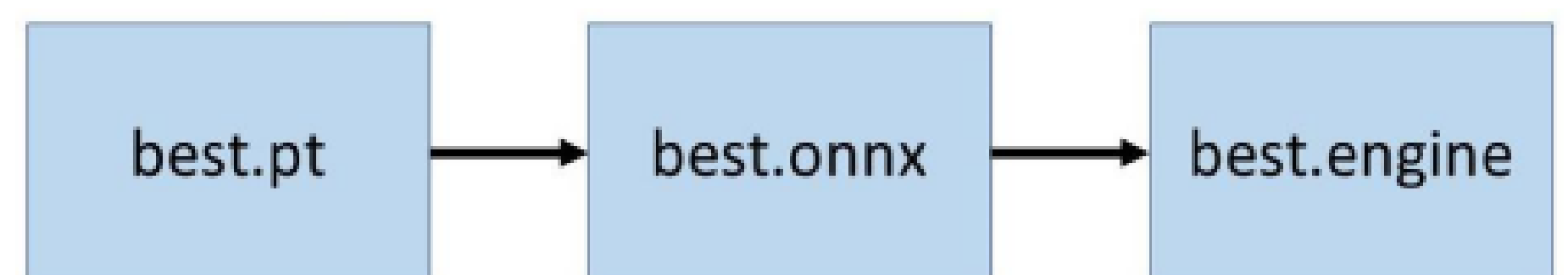
● YOLO 模型訓練

模型訓練使用 GTX 1080 GPU 執行，設定輸入影像尺寸 (img-size) 為 [640, 640]，訓練週期 (epochs) 為 600，批次大小 (batch-size) 為 8。最終選擇模型訓練過程中表現最佳的權重檔，用於系統開發。

	YOLOv7	YOLOv7-tiny
整體精確率	0.981	0.972
整體召回率	0.977	0.966
整體 mAP@0.5	0.98	0.971

● YOLO 模型轉換

為了將模型部屬於資源有限的 NVIDIA Jetson Nano，我們先將模型訓練後的最佳權重檔轉為 ONNX 格式。接著以 TensorRT 將 ONNX 編譯為最適合 GPU 運行的 engine 檔。



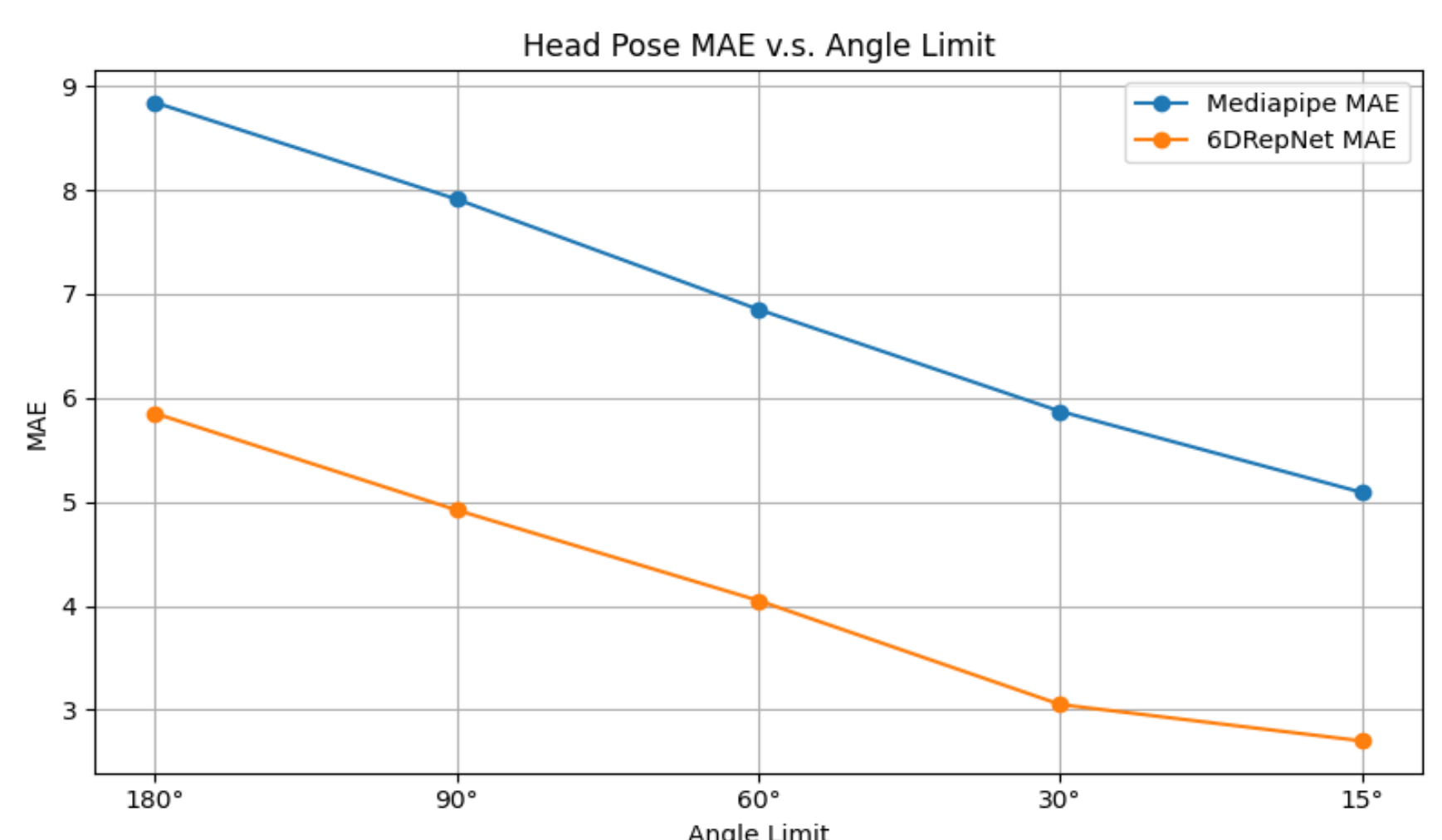
頭部姿態估測方法分析實驗

1. 精確度比較

為比較 MediaPipe 搭配 solvePnP 與 6DRepNet 兩種頭部姿態估測方法要如何選擇於雙模式架構之設計，我們首先以 AFLW2000 公開資料集，通過計算推論角度與資料集提供之 ground truth 之間的平均絕對誤差 (MAE)，驗證兩者估測的精確度。

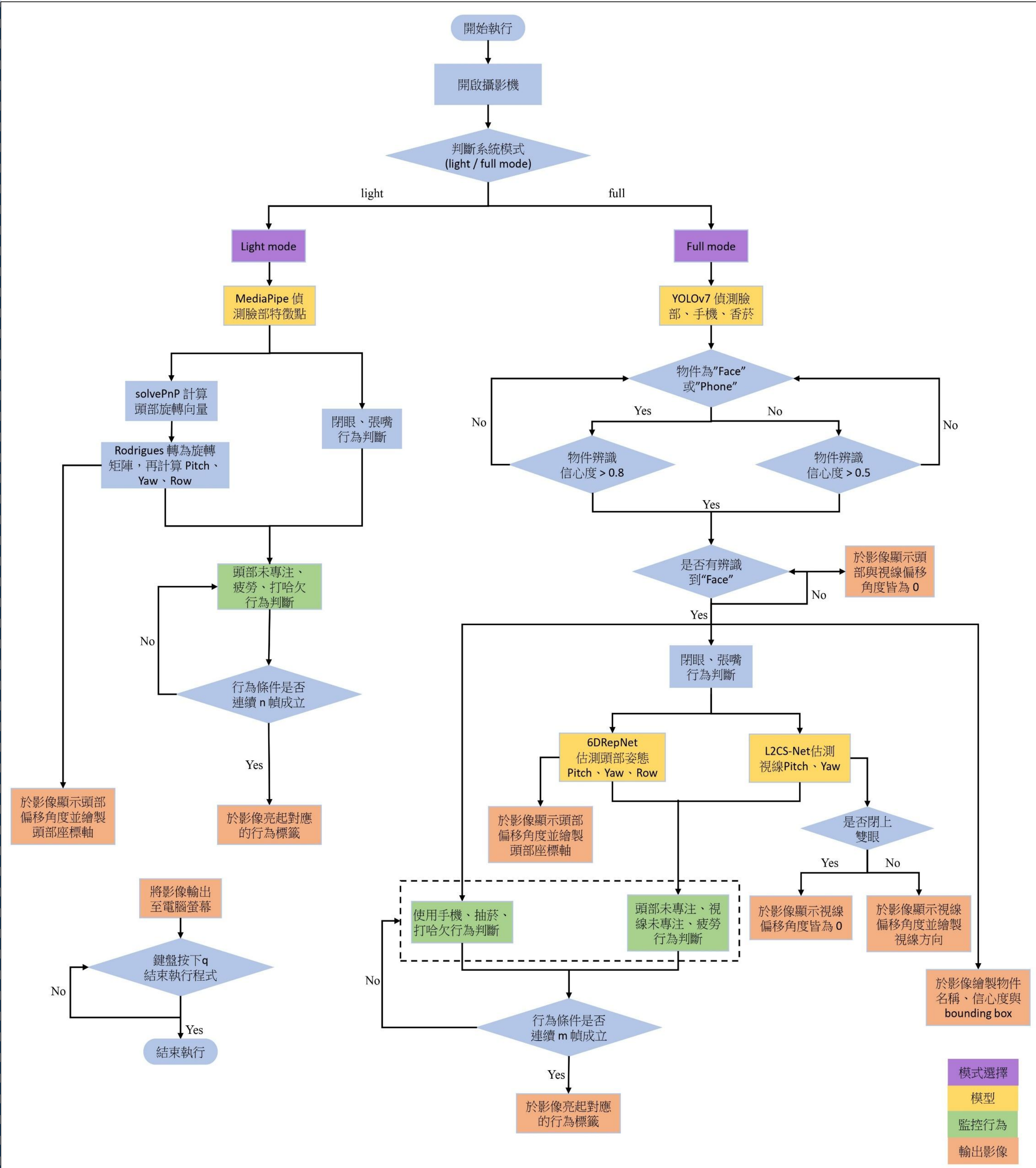
2. 推論速度比較

以開啟攝影機後輸入 100 張影像為樣本，重複三次實驗，計算兩種方法平均估測單張圖片所需的推論速度。

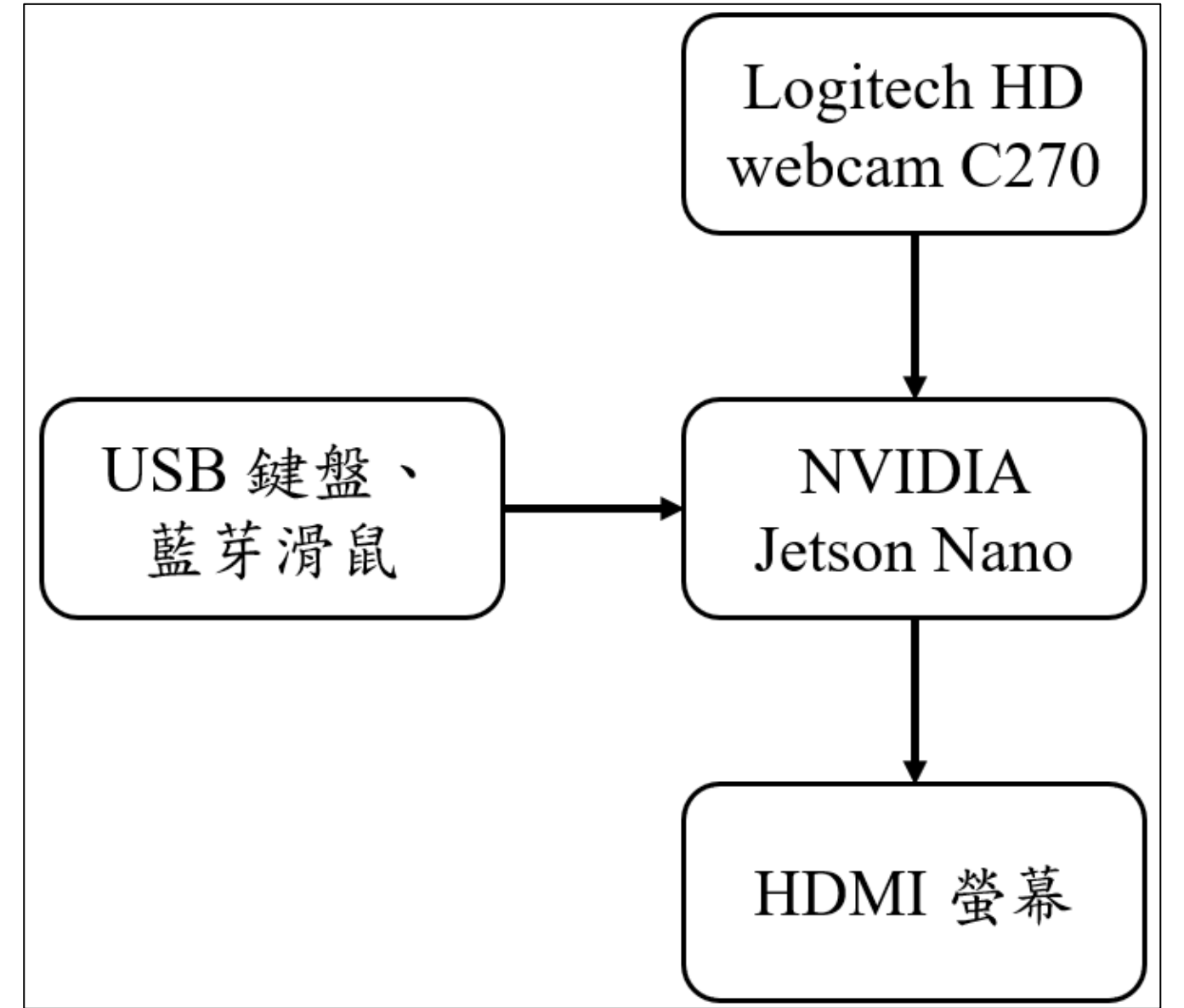


系統架構

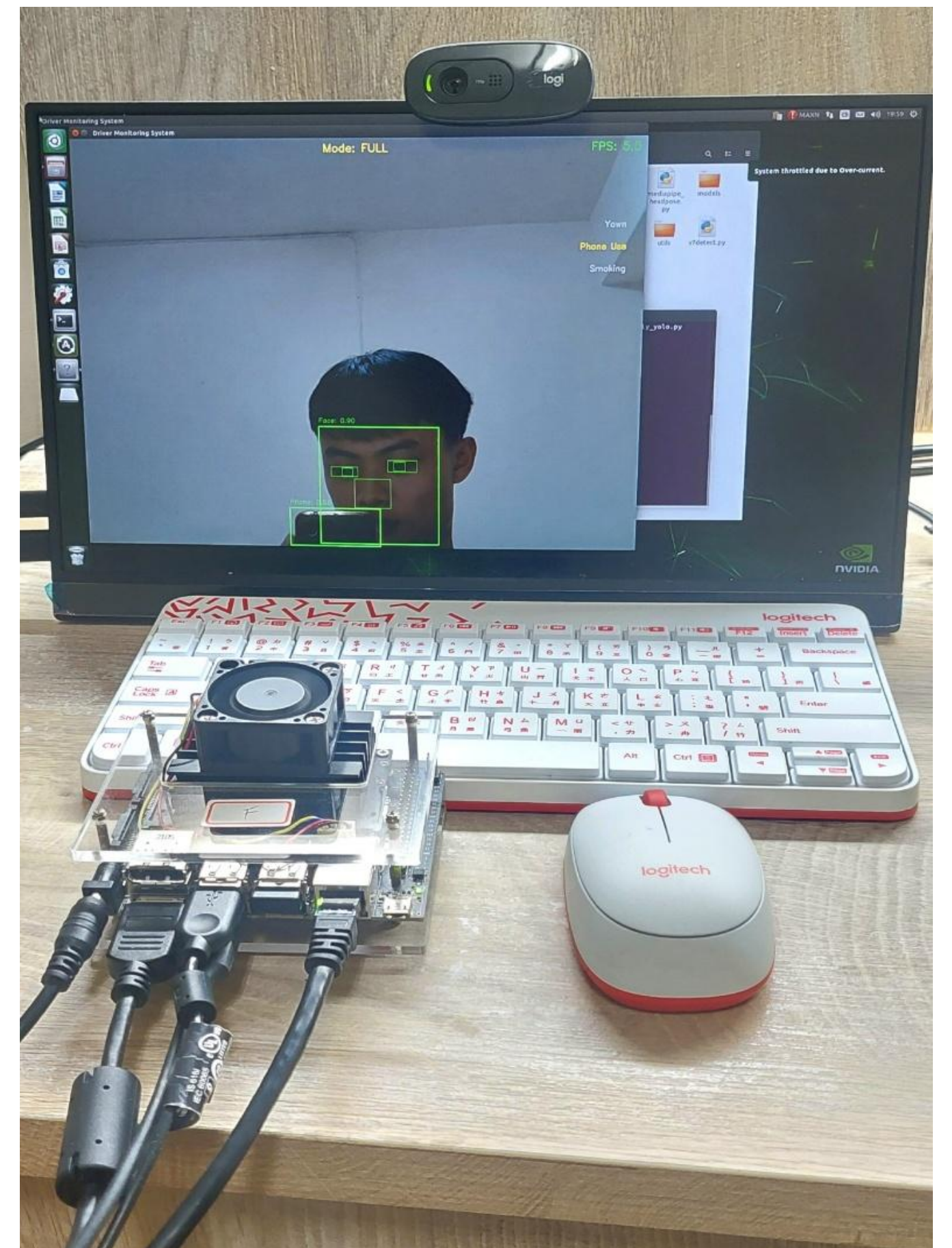
系統使用**延遲實體化(Lazy Initialization)**方式設計，即各模型僅在對應模式啟用後才會載入所需的模型，避免在未需要使用時仍占用額外的硬體資源。



系統流程圖



硬體架構圖

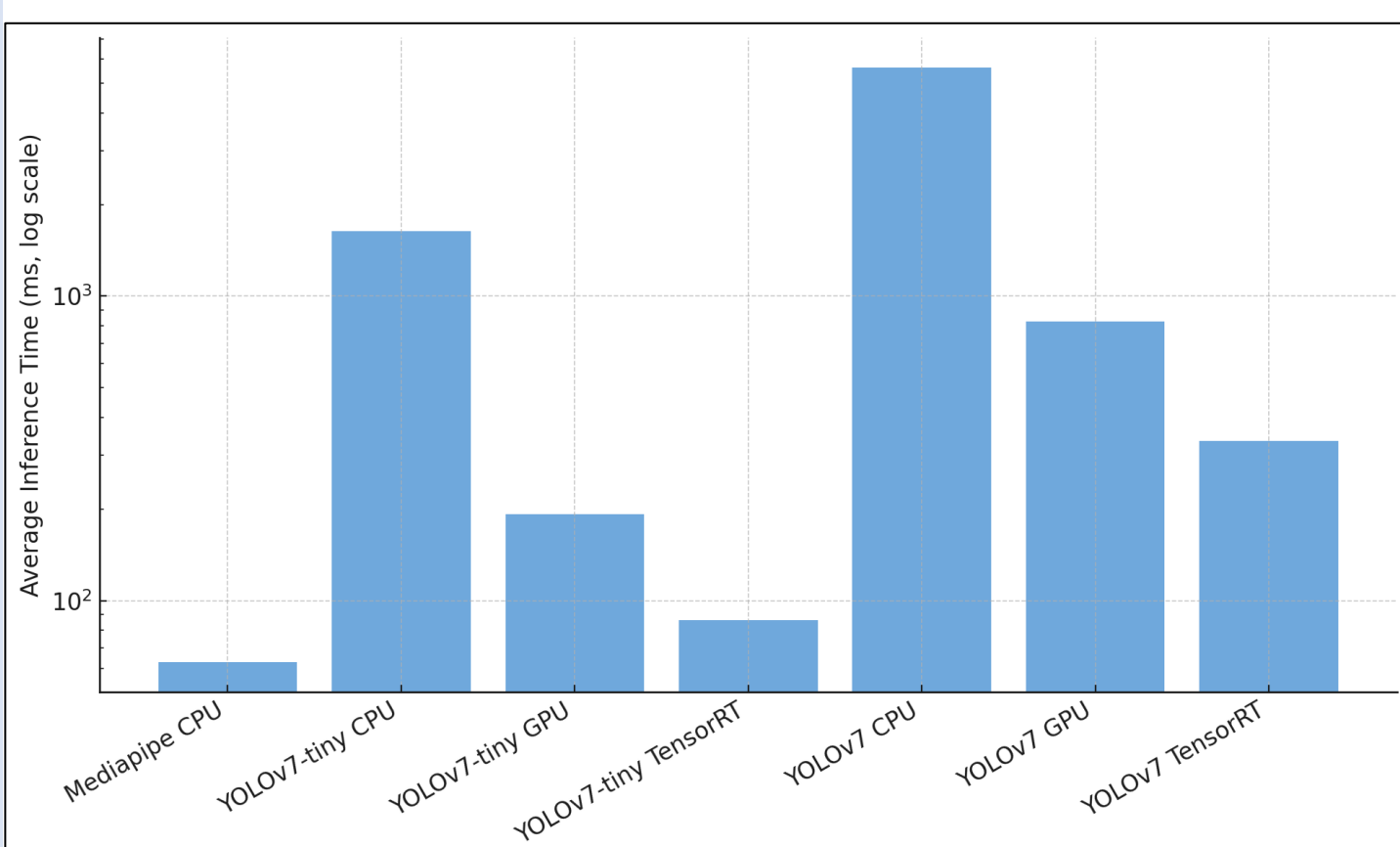


系統部屬於 Jetson Nano

Jetson Nano 推論效能

以開啟攝影機後輸入 100 張影像為樣本，重複三次實驗，計算單張圖片的平均推論時間。

基於 Jetson Nano 運算能力有限，full mode 中僅將 YOLO 模型部屬至 Jetson Nano，並比較 YOLOv7 與 YOLOv7-tiny 在 CPU、GPU 與經過 TensorRT 轉換模型後以 GPU 運算之推論速度。



推論速度比較圖

成果展示

