

**中原大學 x 英特爾 x 華碩「AI PC OpenVINO 應用專題競賽」**  
**2025 CYCU x Intel x ASUS AI PC OpenVINO Application**  
**Project Competition**

**書面報告 Written Project Report**

申請人 【學生】  Applicant	作品名稱 Topic	結合視覺辨識與 OpenVINO 加速的摩托車智慧型 ADB 照明系統 Motorcycle Adaptive Driving Beam Lighting System using visual detection and OpenVINO Acceleration		
	隊名 Team Name	EE Dog		
	組長 Team leader	蔡依琪 YI-QI CAI	電 話 phone	0917081527
	組員 Member	陳芊妤 CHIEN-YU CHEN	電 話 phone	0966440310
	組員 Member	楊尋真 HSUN-CHEN YANG	電 話 phone	0933916613
	組員 Member	彭祥恩 HSIANG-EN PENG	電 話 phone	0916838082
指導老師 Supervisor	姓 名 Name	林康平		
	服務機構及科系(所) Affiliation	中原大學電機工程學系		

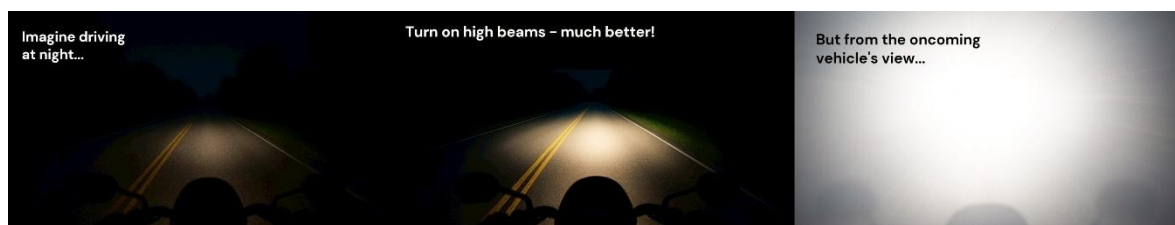
# 目錄

1. 專題背景 ( Project Background ) .....	2
2. 開發目標 ( Objectives ) .....	2
3. 使用技術 ( Technical Methods ) .....	3
3.1 系統架構 .....	3
3.2 技術說明 .....	4
3.2.1 AI 影像辨識模型 .....	4
3.2.2 區域亮度控制 .....	6
3.2.3 UART 傳輸 .....	8
3.2.4 MCU LED 控制 .....	8
4. 成果展示及技術說明 ( Results and Implementation ) ....	8

## 1. 專題背景 (Project Background)

依據經濟部 2019 年發佈報告，**智慧車燈(Adaptive Driving Beam)**為自動駕駛第三階以上必要配備，未來將大幅改變整個車燈產業。

在夜晚中，摩托車大燈無法根據前方車輛或行人的位置自動調整亮度與照射範圍，是道路使用者主要面臨的安全問題之一(如圖一所示)。傳統摩托車大燈多採固定照明模式，若持續開啟遠光燈，可能使對向駕駛或行人造成眩光，影響其視線，進而增加事故風險。相較之下，高階汽車已逐漸引入 ADB 技術，可動態調整光的亮度以兼顧視野及防眩光。然而，在摩托車領域，受限於成本，ADB 技術尚未普及，更缺乏成熟的解決方案。



圖(一) 示意圖

本專題提出一套**摩托車 ADB 智慧照明系統**，預計以 Intel CPU 結合 YOLOv11m 模型，輸入影片或即時影像，接著透過演算法將畫面分成四個區域，即時偵測區域內的汽車、自行車及行人位置，依據區域內物件調整車燈亮度(車燈本專題用 LED 燈模擬)。此方法能有效避免眩光問題，提升夜間行車的安全性與舒適度。

## 2. 開發目標 (Objectives)

本專題意旨在於打造一套能夠即時偵測用路人，並依據目標遠近控制車燈的 ADB 智慧照明系統。本系統的開發目標可分為三大面向：

### 1. 即時偵測

在摩托車高速行駛情境下，系統必須快速且精準地辨識不同位置的車輛、自行車與行人。這部分我們採用 **Intel CPU 搭配 OpenVINO 加速 YOLOv11m 模型**，專注於

物件偵測的技術。

## 2. 低成本

成本是智慧大燈尚未普及到摩托車領域的主要原因之一。我們的專題使用常見的 CPU 加上 OpenVINO 加速推論，再連結微處理機模擬智慧車燈的大燈亮度控制，希望實踐低成本目標。

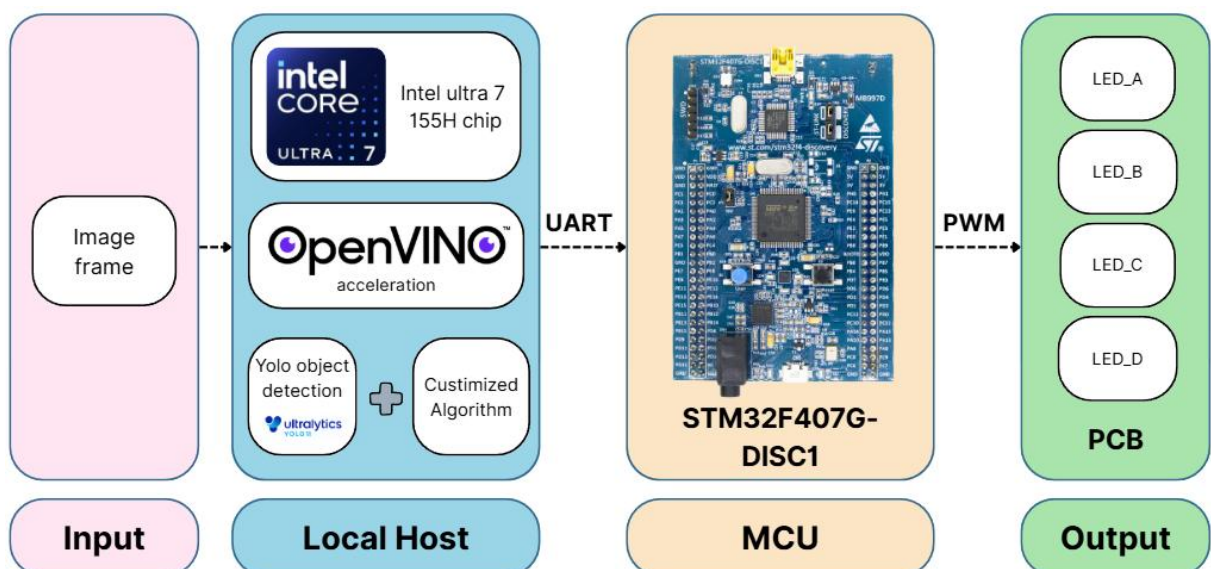
## 3. 低功耗

傳統車燈的功耗是固定的，系統沒有任何調節的機制。而我們的系統則能實現**精準的亮度管理**。透過即時偵測前方路況，系統會根據物體距離、數量與位置，動態調整 LED 燈的亮度與照射範圍。這種**按需照明**的設計，有效避免了傳統車燈持續高功率運作所產生的無效功耗。

# 3. 使用技術（Technical Methods）

## 3.1 系統架構

如圖(二)，摩托車 ADB 智慧照明系統以「**本地端模型運算 + MCU 端亮度控制**」為核心



圖(二) 系統架構圖

系統架構流程如下：

1. 影像輸入：

本次專題輸入為影像、照片等，這些資訊會傳送到由 Intel Core Ultra 7 155H 處理器驅動的系統主機中。影像前處理包括將解析度統一調整至 640x640 像素，符合 YOLOv11m 模型規格同時又能增快運算速度。

2. AI 推論 (YOLOv11m + OpenVINO)：

預處理的影像即進入 **Ultralytics YOLOv11m**，本模型在速度與準確度之間取得了良好的平衡。為了能迅速偵測影像，我們使用 **OpenVINO 工具包**，專門針對 **Intel CPU** 進行推論加速，降低了推論延遲。此階段會即時辨識出影像中所有可能產生眩光的對象，包括**汽車、自行車與行人**。推論結果會輸出每個物件的類別及其在畫面中的邊界框座標，提供後續調整車燈亮度的依據。

3. 區域亮度決策 (Region-Based Brightness Control)

演算法將整個畫面劃分為四個固定區域，並根據每個物件的邊界框座標判斷其所屬區域。將由一組 8 個位元組控制訊號。

4. UART 傳輸 (UART Communication)

生成控制訊號後會透過串列埠傳送給微處理器 (MCU)。

5. 微處理控制 LED (MCU LED Control)

MCU 接收 UART 資料後，解析出各區域亮度值，並即時調整對應的 LED 模組輸出，並調整各區域的對應亮度，兼顧安全性與可視性。

## 3.2 技術說明

### 3.2.1 AI 影像辨識模型

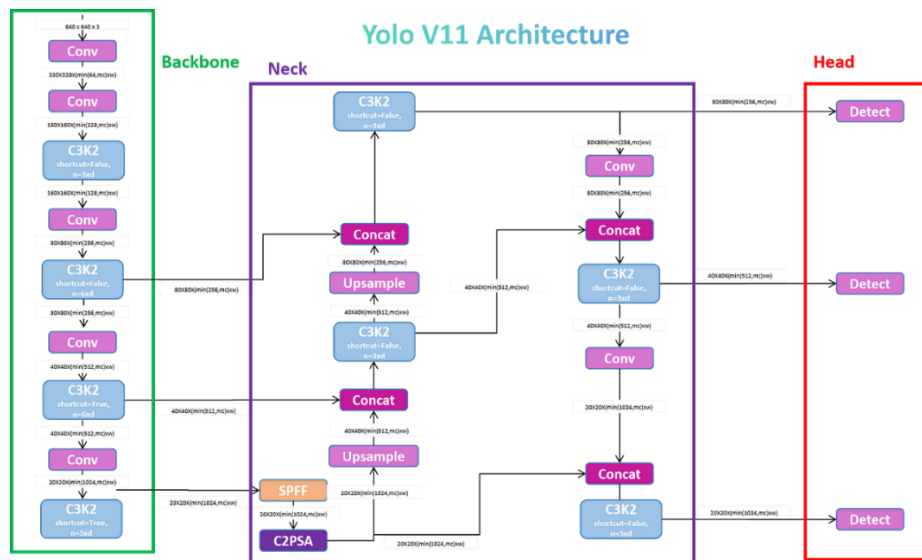
本系統採用 **YOLOv11m** 作為物件偵測模型，並針對摩托車的嵌入式運算環境進行模型壓縮與 INT8 量化，以降低推論計算量並提升 FPS。影像輸入由前置攝影機擷取，經過解析度壓縮後送入

推論模組，直接輸出偵測框與目標類別（汽車、自行車、行人）。YOLOv11 為 Ultralytics 公司於 2024 年推出的最新 YOLO 系列版本，是在 YOLOv8 基礎上的重大升級，兼顧精度與效率，適合即時目標偵測。

如圖(三)，其核心架構引入多項關鍵改進：

- **C3K2 模組**：基於跨階段部分網路（CSP）概念，採用多種卷積核大小與通道分離，提升複雜特徵提取能力。
- **SPFF 模組**（Spatial Pyramid Pooling Fusion）：改進自 SPP 模組，強化多尺度物件的檢測效果。
- **C2PSA 模組**：結合通道與空間資訊，搭配多頭注意力機制，顯著提升特徵表達能力，尤其在複雜場景中可提高檢測精度。

相較於前代版本，YOLOv11 在 COCO 資料集上的準確度提升，同時計算量與參數數量減少，推論速度提升約 25%。其多平台適配能力使其能良好運行於雲端與邊緣裝置，符合本系統的嵌入式部署需求。



圖(三) yolov11 架構圖[1]

在本專題中，我們使用 **Intel OpenVINO™ Toolkit** 為核心開發平台，將 **YOLOv11m** 與 **Intel CPU + OpenVINO** 推論引擎結合，有效縮短推論時間並降低功耗，確保在嵌入式環境中仍能達到

即時運作的性能標準。

## 選擇 YOLOv11m 的原因

表(一) 不同 YOLOv11 模型效能與準確率比較

模型大小	FPS	準確率
n	46.8 - 56.2	中
s	0.4 - 38.9	低
m	12.4 - 18.3	高

在相同測試影片下，我們分別使用 YOLOv11n、YOLOv11s 與 YOLOv11m 三種模型進行推論，並比較其 FPS 與偵測準確度表現(如表一)。

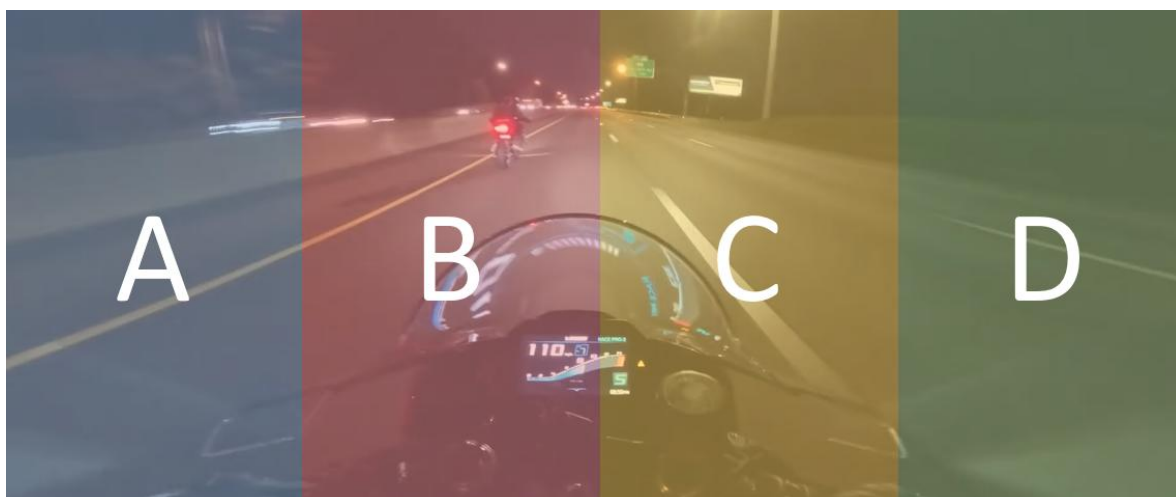
FPS 由系統即時監測所得；準確率部分，雖未進行定量的數值計算，但在多次測試中可明顯觀察到模型間的辨識效果差異：

- **YOLOv11n**：推論速度最快，但在夜間環境下，部分物體（特別是遠距離行人與小型車輛）漏檢情況明顯。
- **YOLOv11s**：速度次之，但偵測效果不穩定，對小目標與邊緣區域物體辨識效果最差。
- **YOLOv11m**：雖然推論速度較慢，但能穩定辨識大部分物體，且在多種光照條件下表現最佳。

基於此觀察，本系統選擇在 Intel CPU + OpenVINO 環境下部署 YOLOv11m，以在可接受的 FPS 範圍內，獲得較佳的夜間辨識準確率與穩定性。

### 3.2.2 區域亮度控制

在完成影像辨識後，系統會依據偵測到的物體位置，將畫面劃分為四個固定區域：A、B、C、D(如圖(四))。每個區域對應摩托車車燈的不同照射方向，目前在實驗階段以 LED 模擬車燈輸出，以方便開發與測試。



圖(四) 畫面分割示意圖

表(二) 亮度等級

亮度等級	字元
0% (完全熄滅)	0
25% (低亮度)	1
50% (中亮度)	2
100% (全亮)	3

亮度共分為四個等級：0%（完全熄滅）、25%（低亮度）、50%（中等亮度）、100%（最亮），並使用字元分別表示為「0、1、2、3」。系統會依照固定順序（A→B→C→D）將四個區域的亮度值組合成一條字串（AxBxCxDx）。例如：若亮度設定為 A=2、B=1、C=3、D=0，則最終生成的控制字串為 A2B1C3D0，亮度等級及其對應字串（如表(二)）。

系統會依據物體所在的區域，動態分配亮度等級。例如，當行人被偵測到出現在 A 區域時，該區域的亮度會立即降低至 25%、50% 或 0%，其餘區域則保持 100%，確保駕駛視線不受影響的同時，也減少對其他用路者的眩光干擾。生成的字串會透過 UART 以固定格式傳送至 STM32F407G-DISC1 微處理機（MCU），由 MCU 調整對應區域的 LED 輸出亮度。此種區域化亮度控制方式，模擬系統能在短時間內完成車燈亮度調整，有效減少對向駕駛的眩光，同



時確保騎士的主要視野清晰。

### 3.2.3 UART 傳輸

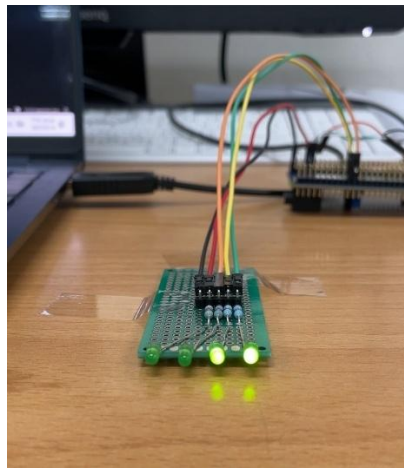
在完成四區域亮度等級計算後，系統會將結果轉換成 UART 字串，並傳送至微控制器（MCU）。UART 傳輸採用 115200 bps 的波特率(Baud Rate)與標準 8-N-1 格式（8 位元組資料、無同位元、1 個停止位元），以確保資料傳送的即時性與穩定性。

### 3.2.4 MCU LED 控制

微處理器（MCU）負責接收來自主控端經由 UART 傳送的亮度控制字串，並即時解析各區域的亮度等級。在接收到控制字串後，MCU 會根據預先定義的 PWM（Pulse Width Modulation，脈衝寬度調變）參數輸出至我們自製的 LED 電路，模擬車燈在不同亮度下的照明效果。

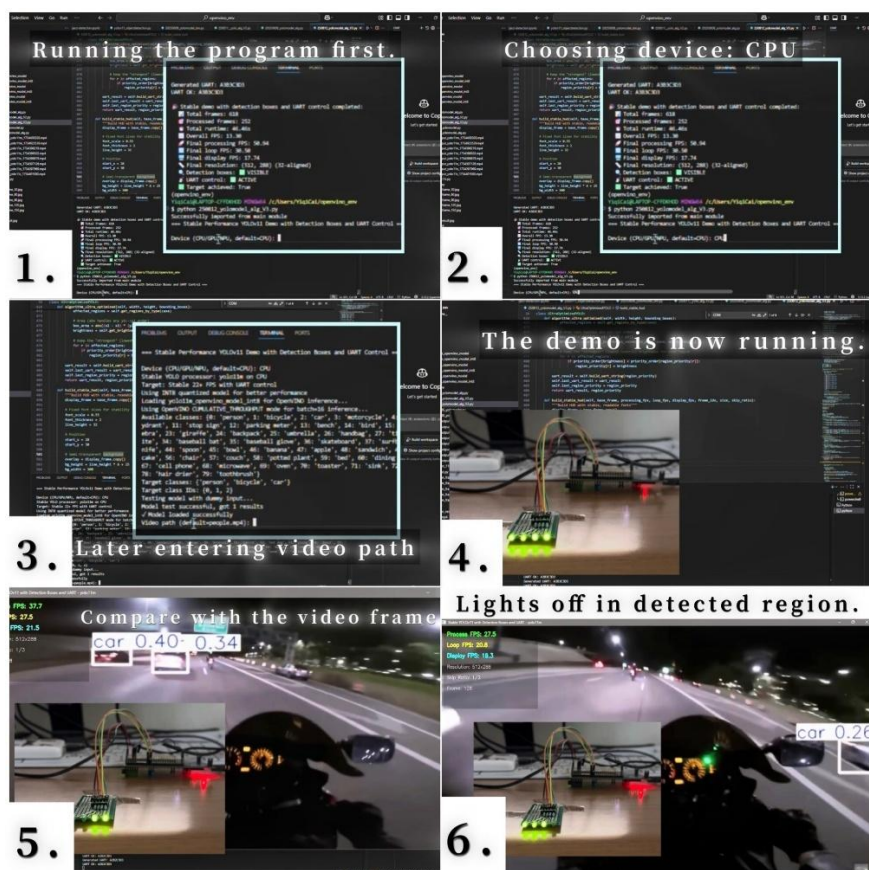
## 4. 成果展示及技術說明（Results and Implementation）

目前，系統已完成 YOLOv11m 模型與 OpenVINO 加速推論的整合，並實作了多區域 LED 燈光亮度控制機制(如下圖 X)。



圖(五)，ADB 模擬實作

如圖(六)，在測試影片[3]中，當行人或車輛進入特定照明區域時，系統能即時降低該區亮度，其餘區域維持 100%，以確保主要視線範圍的清晰度。



圖(六) 測試結果圖

在系統展示影片中，呈現了以下流程：

### 1. 啟動程式

畫面首先顯示在終端機中執行 ADB 智慧照明系統的主程式。這個步驟會載入 YOLOv11m 模型與 OpenVINO 加速環境，並初始化 UART 連線，為後續燈光控制做準備。

### 2. 選擇運行設備

在程式啟動後，使用者選擇運行平台為 **Intel CPU**，這個選擇會直接影響推論速度與功耗表現，本系統針對 CPU 環境做了優化，能在不使用 GPU 的情況下達到即時偵測的需求。

### 3. 輸入影片路徑

程式啟動並設定好設備後，會要求使用者輸入測試影片的路徑。

影片來源為摩托車第一人稱視角夜間行車畫面，用來模擬真實道路場景，測試系統在動態環境下的偵測與燈光控制反應。

#### 4. 系統開始運行

進入運行階段後，YOLOv11m 模型會即時分析影片畫面，辨識出汽車、自行車與行人，並根據位置計算該屬於哪一個燈光控制區域。此時 LED 燈板（由 MCU 控制）會對應模擬摩托車大燈的四個分區，進行亮度調整。

#### 5. 偵測與燈光對應

畫面中可見 YOLO 偵測框標示出目標（例如 "car 0.40" 代表汽車，信心度 40%），並即時關閉該區域的 LED。這代表系統成功將偵測結果轉換為燈光控制指令，透過 UART 傳輸到 MCU 執行。

#### 6. 動態測試效果

在持續播放影片的過程中，系統會根據每一幀的偵測結果，快速地關閉或開啟對應區域的 LED，以模擬真實摩托車大燈的自適應遠光功能。這確保了在車輛接近或對向行駛時，對方駕駛不會被眩光影響，同時保留其他區域的高亮照明以維持騎士的可視範圍。如圖(七)



圖(七) 測試結果圖，分為三情境

表(三) OpenVino 加速對照圖

	FPS
<b>With OpenVINO Acceleration</b>	<b>12.4 ~ 18.3</b>
<b>Without OpenVINO Acceleration</b>	<b>4.5 ~ 6.4</b>

實驗結果顯示，在夜間環境下，系統於即時運行時的平均 FPS 可達約 15。如表(三)所示，滿足高速行駛下的辨識需求。然而，偵測穩定度略低，顯示未來需增加該環境下的訓練資料，以提高辨識準確率。

### 系統結果統整

本系統成功實現了基於 YOLOv11m 與 OpenVINO 加速的摩托車 ADB 智慧照明控制，能夠即時辨識道路上的車輛與行人，並對應到四個獨立照明區域進行亮度調節。

測試結果顯示，LED 能在物體進入對應區域時立即降低燈亮度，模擬了摩托車頭燈能夠即時降低光束亮度，而螢幕畫面可同步顯示偵測框、分區邊界與 FPS 資訊，方便進行觀察與效能分析，達到防眩光與維持駕駛視野的雙重目標。

### 【參考及引用文獻】

圖片來源：

[1][https://www.moea.gov.tw/mns/doit/industrytech/IndustryTech.aspx?menu\\_id=13545&it\\_id=270](https://www.moea.gov.tw/mns/doit/industrytech/IndustryTech.aspx?menu_id=13545&it_id=270)

[2]<https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71>

[3] <https://youtu.be/E1E6X10QyBc?si=0V2Ms8sr2chyXaC0>