

附件四、報告格式

# 即時海洋觀測平台開發暨AI影像數據應用服務主題式研發計畫

期末報告

# 即時海洋觀測平台開發暨 AI 影像數據應用服務主題式研發計畫

## 期末報告

計畫主持人：林家瑜 助理教授

研究機構：國立中央大學資訊工程系

執行期間：113/05/01-114/04/30（請填寫合作研究計畫執行期間）

（以上資料請務必以中文繕打清楚）

中華民國 114 年 01 月 31 日

## 計畫摘要

本研究致力於開發即時海洋觀測平台，並應用 AI 影像數據分析來提升遠距離船隻辨識的準確性與效率。現行船舶識別技術主要依賴人眼辨識，然而在 3 公里以上距離，由於海面反射、霧氣與光線變化等環境因素的影響，傳統方法的準確率明顯下降。本計畫旨在透過深度學習物件辨識技術，建立一套高效能的「即時遠距離多船隻辨識系統」，能夠在 3 公里外準確偵測船隻，並可同時識別 10 艘以上長度超過 15 公尺的船隻。

本研究參考 YOLOv5s 與 YOLOv8 物件辨識模型，針對海洋環境特性進行改良，並利用多種公開及自建數據集如 Singapore Maritime Dataset、ABOships-PLUS、高雄港實測數據進行模型訓練與驗證。透過影像增強、特徵提取優化與多尺度檢測技術，我們顯著提升了模型在複雜海洋場景中的識別能力。

目前的實驗結果顯示，我們的系統在新加坡與高雄測試數據集中的部分影像皆展現超過 90% 的識別準確度，並成功達成：

- 可偵測移動速度超過 15 節的船隻
- 在 5 公里範圍內實現高效識別
- 可即時處理 640×480 解析度影像，幀率≥10FPS
- 具備船隻避碰預警功能

未來將進一步提升船隻辨識與船隻避碰預測的準確度，並擴充模型至夜間與黑白影像辨識，以提升即時監測與應用能力。此技術的發展對於海洋交通安全監控、智慧港口管理與海上救援應用將具備重要價值。

關鍵字：即時海洋觀測、AI 影像辨識、深度學習、YOLOv5、YOLOv8、多尺度檢測、船隻辨識、避碰預警、智慧港口、遠距離監測

## 目錄

計畫摘要 .....	2
一、計畫動機與目的 .....	4
1. 動機 .....	4
2. 目的 .....	4
二、文獻探討 .....	5
三、研究方法 .....	8
1. 資料收集 .....	8
2. 資料前處理 .....	10
3. 船隻辨識模型 .....	10
四、實驗結果 .....	12
五、結論 .....	18
六、參考文獻 .....	19

## 一、計畫動機與目的

### 1. 動機

在海洋環境中，對於船隻的及時且準確的辨識是保障航行安全和有效執行海上作業的關鍵。但目前大部分的船隻都是人工辨識，而人眼無法同時看超過 10 艘船隻進行辨識，因此需要 AI 輔助。現有的 AI 對距離近且較大的船或是俯視角的衛星雷達成像的船有不錯的辨識能力，但若是在第一人稱且距離超過 3 公里的情況下，可能會受到環境條件如海面反射、霧氣、光線變化等的影響，導致偵測的準確度降低，且現有開放資料集大多是較大的船隻，3 公里以上的船隻資料集較少，不利於訓練模型。

### 2. 目的

基於上述的問題，本計畫欲設計一套「即時遠距離多船隻辨識系統」，此系統可以有效辨識 3 公里遠的船隻，且一次可辨識長度 15 公尺的船隻 10 艘以上。

## 二、 文獻探討

現在有一些研究採用物件辨識模型進行船隻辨識，在 Automatic Ship Object Detection Model Based on YOLOv4 with Transformer Mechanism in Remote Sensing Images 論文中提出了名為 Auto-T-YOLO 的架構[1]，用於解決現有物件偵測模型從輸入中提取特徵未能充分考慮全局特徵，且無法根據輸入的特徵自動調整，導致偵測準確度下降的問題。

Auto-T-YOLO 的模型架構分為三個部分，Preattention, Attention 和 Prediction。輸入影像大小為  $608 \times 608 \times 3$ ，Preattention 和 Attention 的設計模仿了人類視覺系統中從大的全局特徵開始，再逐步轉向局部細節的兩階段注意機制。

在 Preattention 架構中，輸入的圖像會根據其中是否包含  $size > 32 \times 32$  的物件分為 A 子資料集及 B 子資料集，並在之後的 Attention 架構中根據其所在的子資料集做不同的影像處理。

Attention 架構主要參考了 YOLOv4 的 Backbone 及 Neck 架構來設計，A 子資料集及 B 子資料集在 Backbone 中的處理流程相同，都是利用一層 CBM 進行卷積運算以及五層 CSPDarknet53 進行殘差及卷積運算來提取特徵同時防止梯度消失，由於 YOLOv4 的 CBL 架構基於卷積運算，而卷積運算受限於 Kernel 大小，因此無法很好地提取全局特徵，所以 A 子資料集在 Neck 中 SPP 架構輸出之後使用了 Transformer 中的 Multi-head Self-Attention 架構取代了 CBL，稱為 SPP-MHSA，其中 Multi-head Self-Attention 使用了四個 head。Neck 的後半段與 YOLOv4 相似，使用了 FPN 做 Upsampling、PANet 做 downsampling 以提升多尺度目標檢測的能力。

Prediction 架構用來對 PANet 輸出做多尺度目標檢測，損失函數使用了 CIoU 以及 DIoU-NMS。模型準確度在 SSDD 資料集、近海及遠海場景上分別達到了 96.3%、91.78% 以及 98.33%。

在另一篇論文 A streamlined approach for intelligent ship object detection using EL-YOLO algorithm 中提出了 EL-YOLO 的架構[2]，用於提升目標檢測在具有波浪、反射等複雜海洋環境中對於小物體的檢測精度，其輕量化的網路架構在硬體運算資源有限的智慧船舶、無人船舶等領域也有良好的適應性。

EL-YOLO 的架構基於 YOLOv8 進行改進，輸入影像大小為  $640 \times 640$ ，包含了 Backbone、Neck、Head 三個部分，Backbone 中與 YOLOv8 相同，使用了一層 CBS 進行卷積運算和四層 CBS 加 C2f 進行殘差及卷積運算來提取特徵同時防止梯度消失。

在 Neck 部分，因為傳統的 FPN 和 PAN 架構在面對因反射或波浪等背景干擾因素導致特徵信息不足的目標時，容易在多層卷積過程中丟失信息，導致檢測精度下降，所以作者提出了 SMFN (short multi-fuse neck) 架構，在

YOLOv8 的基礎上，額外加入了較大尺寸的兩個輸入( $320 \times 320$ 和 $160 \times 160$ )、一個 Upsampling 層( $160 \times 160$ )、一個 Fusion 加 C2f 層( $160 \times 160$ )、一個卷積層( $80 \times 80$ )及一個輸出( $160 \times 160$ )到 FPN 和 PAN 架構中，以捕捉更多的特徵信息，並提高對小型目標的檢測能力。

Head 架構因為 Neck 的 SMFN 架構而多了一個 $160 \times 160$ 的輸出，相較原本只有三個輸出的 YOLOv8 多了一個。在損失函數的部分，因為傳統的 CIOU 在處理特徵信息不足的目標時往往會過度懲罰，因此作者提出了 AWIoU(adequate wise IoU)，通過動態分配不同權重給不同特徵信息量的目標來調整懲罰，避免目標因受背景干擾而造成模型誤判。

在模型訓練完後，為了使其網路架構輕量化以應用於硬體運算資源有限的智慧船舶、無人船舶等領域，作者提出了 GDFP(Greedy-driven filter pruning)，首先計算每一層中所有 filter 的 L1 範數以得到其重要性分數，並將分數最低的 filter 刪除，在刪除後對模型重新評估，並重複以上操作直到得到理想的模型，最後重新訓練該模型，以得到將模型輕量化的效果。

此模型在 SeaShips 資料集上的 mAP0.5 為 0.988，mAP0.5:0.95 為 0.778，在 ABOShips 資料集上的 mAP0.5 為 0.672，mAP0.5:0.95 為 0.348。

另一篇論文 YOLOSeaShip: a lightweight model for real-time ship detection [3]中提出了 YOLOSeaShip 的架構，用於解決海洋環境中目標檢測的準確性與硬體資源需求之間的矛盾，特別是在實時性要求較高且硬體運算能力有限的應用場景如邊境監控、智慧海洋交通中。

YOLOSeaShip 的架構基於 YOLOv7-tiny 進行改進，包含了 Input、Backbone 和 Head 三個主要部分。

Input 架構上採用了 Mosaic 數據增強技術，通過隨機裁剪並拼接多張影像，增加數據的多樣性與小目標樣本數。此外，使用自適應圖片縮放技術來處理不同尺度的目標，並結合 K-means 分群方法優化錨框的尺寸設定，避免人工設計過程中的偏差。

Backbone 的部分，YOLOSeaShip 引入了 PConv (Partial Convolution) 架構替代了原本的卷積操作。該架構僅對部分通道進行計算，從而有效減少了參數量並提升計算效率。此外，每層池化操作後還加入了 PfAAM (Parameter-Free Average Attention Module)。PfAAM 基於空間和通道的平均值計算注意力權重，不增加模型參數的情況下提升了模型對關鍵區域的定位能力。

Head 部分，模型加入了 SPPCSPC\_tiny 架構，通過多尺度( $5 \times 5$ 、 $9 \times 9$ 和 $13 \times 13$ )最大池化層結合本地與全局特徵，增強對不同大小目標的檢測能力。模型的最終輸出包含三個不同大小的特徵圖，用於處理不同尺度的目標檢測。

在損失函數設計上，作者提出了 Focal EIOU (Focal and Efficient IOU) 損失函數，結合了 EIOU 和 Focal Loss 的優點。EIOU 損失通過細化目標框

的寬高比差異來提升回歸精度，而 Focal Loss 則用於平衡正負樣本比例，讓模型更關注高質量的預測框。

為了評估模型性能，作者在 SeaShips(7000)資料集上進行實驗，訓練時採用 Adam 優化器，學習率為 0.001，訓練 300 個 epoch。結果顯示，YOLOSeaShip 的 mAP 達到 0.976，檢測速度達到 119.84 FPS，相較 YOLOv7-tiny 模型，參數量減少至 4.78M，同時大幅提升了檢測速度和資源效率。

在論文 An Improved YOLO v4 Algorithm-based Object Detection Method for Maritime Vessels[4]中提出了基於改進 YOLOv4 的船舶和泊位檢測方法，用於提升目標檢測在複雜海洋環境下的準確性，特別是對港口泊位的檢測效果進行了優化。該方法通過改良錨框的生成方式和數據集構建，改善了 YOLOv4 在海洋場景中的檢測性能，適用於智能船舶的安全輔助系統。

改進的 YOLOv4 架構包含了 Backbone、Neck 和 Head 三個部分。Backbone 部分採用了 CSPDarknet53 作為特徵提取模塊，通過 ResUnit 組件進行深度卷積操作，增強特徵提取能力。Neck 部分引入了 SPP 層和 PANet 層，其中 SPP 層通過多尺度池化操作擴展感受野，而 PANet 則將底層的定位信息與高層的語義信息結合，提高多尺度目標的檢測能力。Head 部分則採用了 YOLOv3 的檢測頭，輸出不同尺度的檢測結果。

針對泊位檢測效果不足的問題，作者採用了 K-means 演算法重新生成錨框。在原始 YOLOv4 模型中，錨框基於 MS COCO 和 VOC 數據集生成，但這些框與船舶與泊位數據集中的物體尺寸不匹配，影響了檢測精度。改進後的 K-means 演算法針對數據集進行分群，生成了 9 組適配的錨框，有效提升了檢測效率和準確性。

此外，作者構建了一個包含船舶和泊位的專用數據集，通過智能實驗船上的攝像頭在港口拍攝的實景影像生成，並結合公開數據集如 MS COCO 和 VOC 補充正樣本數量。最終數據集中包含 6634 張標註影像，其中 70%用於訓練，30%用於測試。標註信息包括物體類別及邊界框的坐標和尺寸。

實驗結果表明，改進的 YOLOv4 模型在 mAP 和 F1-score 方面分別提升了 2.79%和 0.80%，相較於原始 YOLOv4 達到了更高的檢測精度與穩定性。具體指標顯示，改進後的 YOLOv4 模型在 mAP 上達到 83.98%，F1-score 達到 86.18%，而 YOLOv4\_tiny 僅達到 54.45%的 mAP 和 54.82%的 F1-score，說明輕量化模型雖然具有更快的檢測速度，但在檢測準確性上仍有進步空間。



### 三、 研究方法

#### 1. 資料收集

資料集使用 Singapore Maritime Dataset 的 Visible On-Shore 資料集[5, 6]加上 ABOships-PLUS[7]以及手動標註公司提供的高雄資料集。




Visible On-Shore 資料集為網路上的公開資料集，包含 38 部長度小於 33 秒的全彩影片，屬於中、遠港船隻影像資料集，其中含有 7 個船隻類別，此資料集特徵為船隻背景為海面、天空等較乾淨的畫面。

表一、Visible On-Shore 資料集類別

Lable	Data		
Ferry		SailsBoat	
PatrolBoat		MerchantShip	
FishingBoat		Tugboat	
BargeShip			





ABOships-PLUS 資料集為網路上的公開資料集，ABOships 資料集的改進版本，包含 9880 張影像，33227 個標註，屬於近港船隻影像資料集，其中含有 3 個船隻類別，此資料集特徵為船隻背景包含陸地建築物等容易誤判的畫面。

表二、ABOships-PLUS 資料集類別

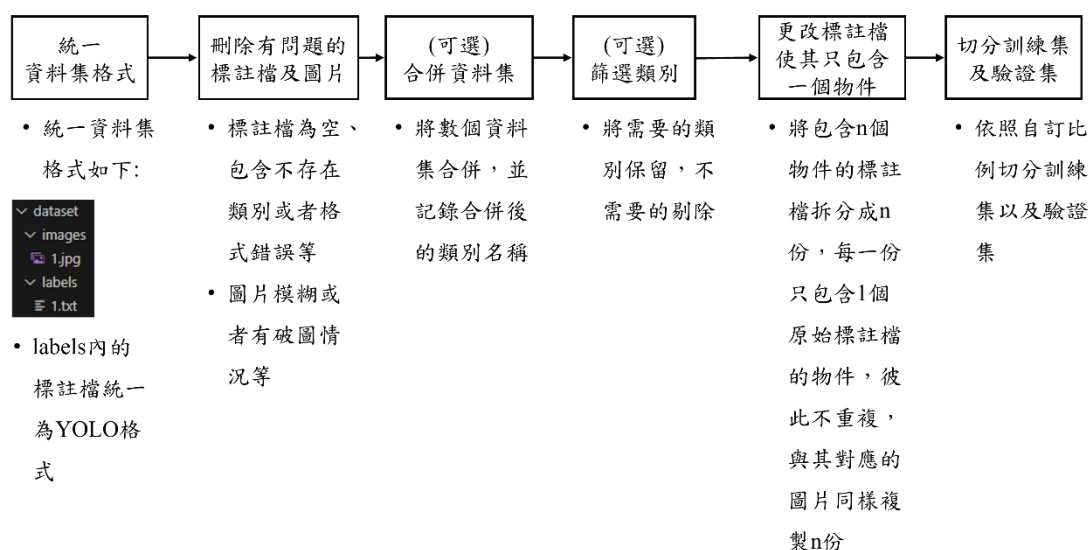
Lable	Data		
powerboat		sailboat	
ship			

高雄資料集為高雄實地拍攝的非公開資料集，包含 504 部長度小於 10 秒的影片，屬於近、中港船隻影像資料集，其中含有 4 個船隻類別，此資料集特徵為船隻背景包含陸地建築物等容易誤判的畫面。

表三、高雄資料集類別

Lable	Data		
PatrolBoat		Yacht	
MerchantShip		Tugboat	

## 2. 資料前處理

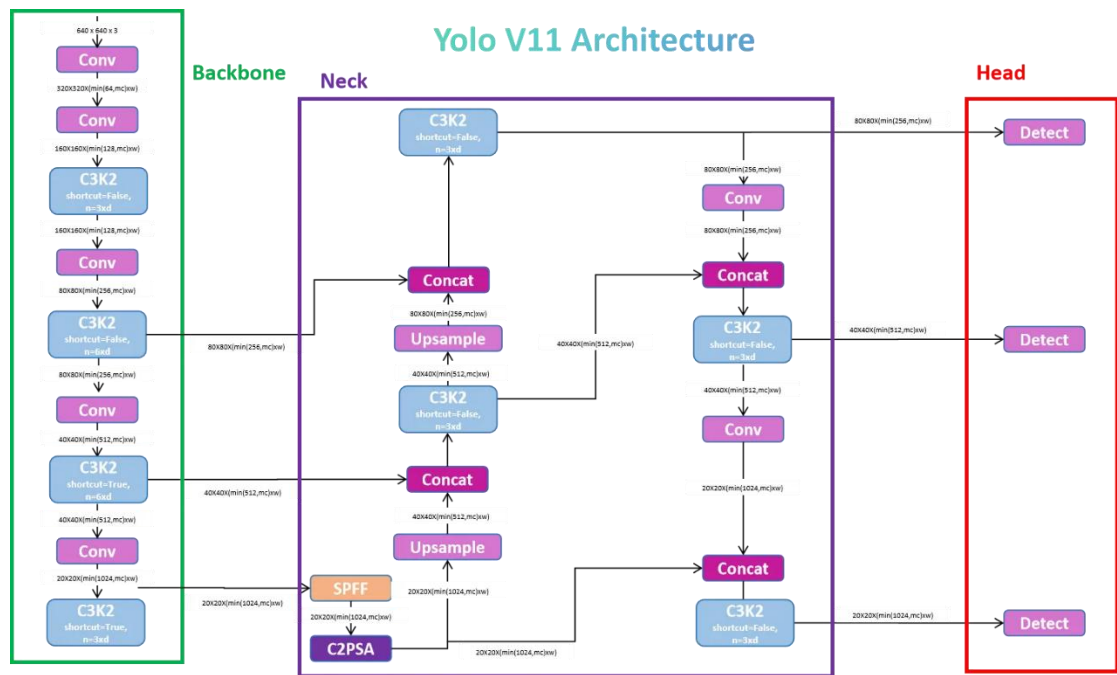


圖一、資料前處理流程圖。

如圖一所示，首先會將資料集處理成只包含 images 和 labels 資料夾的形式，images 和 labels 中的檔案是兩兩對應的，並且 labels 內的標註檔格式為 yolo 格式。再來會刪除有問題的標註檔及圖片，例如標註檔為空、包含不存在類別、格式錯誤、圖片模糊或者有破圖情況等。第三步是合併資料集，這步是可選的，主要會將數個資料集合併，並記錄合併後的類別名稱。第四步是篩選類別，這步同樣為可選，主要會將需要的類別保留，不需要的剔除。接著因為同一圖片中可能包含數量不等的不同類別物件，因此在切分訓練集和驗證集時，容易造成不同類別的訓練驗證比例不同的情況，所以第五步會將包含 n 個物件的標註檔拆分成 n 份，每一份只包含 1 個原始標註檔的物件，彼此不重複，且與其對應的圖片同樣複製 n 份，以確保每個類別都按照正確比例進行切分，最後一步為按照自訂比例切分訓練集及驗證集。

## 3. 船隻辨識模型

我們使用了最新的 YOLOv11[8]模型來建置船隻辨識模型，YOLOv11 基於 PyTorch 架構開發，可在一次前向傳播中同時定位及辨識物件，具有速度快、所需資源少、開發便利及多尺度檢測等優點。圖二為 YOLOv11 架構圖，首先輸入的圖片為 640x640x3 的 RGB 圖像，接著 YOLOv11 可以根據此架構分為三個部分，Backbone, Neck 以及 Head。



圖二、YOLOv11 架構圖[9]

Backbone 如圖二左側所示，主要目的是從輸入影像中提取有用的特徵資訊。在 YOLOv11 中，Backbone 採用了 C3K2 結構，為升級版的 C3，透過更小的卷積核與路徑切分技術，在保持模型表現的前提下大幅降低了計算成本。每個 C3K2 中包含兩條路徑，一條直接進行卷積處理，另一條則經過多個 Bottleneck 區塊進行深度特徵學習，最後進行特徵融合。這種結構不僅有效防止梯度消失問題，也能提升網路的收斂速度與特徵提取效率。

Neck 部分如圖二中間所示，結合了 SPFF(Spatial Pyramid Pooling Fast) 架構和多層 C2PSA 區塊。SPFF 架構是 SPP 的高速版本，透過不同尺度的最大池化操作將特徵圖進行多尺度融合，再透過 Concatenation 將資訊彙整，幫助模型更好地識別大小不一的船隻。此外，Neck 中的 C2PSA 架構引入空間注意力機制(Spatial Attention)，能讓模型聚焦在關鍵影像區域，特別是對小型或部分遮蔽的物件表現更佳。這部分的架構延續了 FPN 與 PAN 的優點，強化了從深層到淺層、再回到深層的資訊傳遞，進一步提升辨識與定位的能力。

Head 如圖二右側所示，負責進行最後的物件偵測任務。YOLOv11 採用多尺度輸出策略，透過多層輸出特徵圖分別負責不同大小目標的偵測。YOLOv11 會利用不同尺寸與比例的錨框(Anchor Boxes)來涵蓋不同形狀的目標，並透過 Non-Maximum Suppression(NMS)過濾重疊的邊界框，保留最具代表性的偵測結果。

## 四、實驗結果

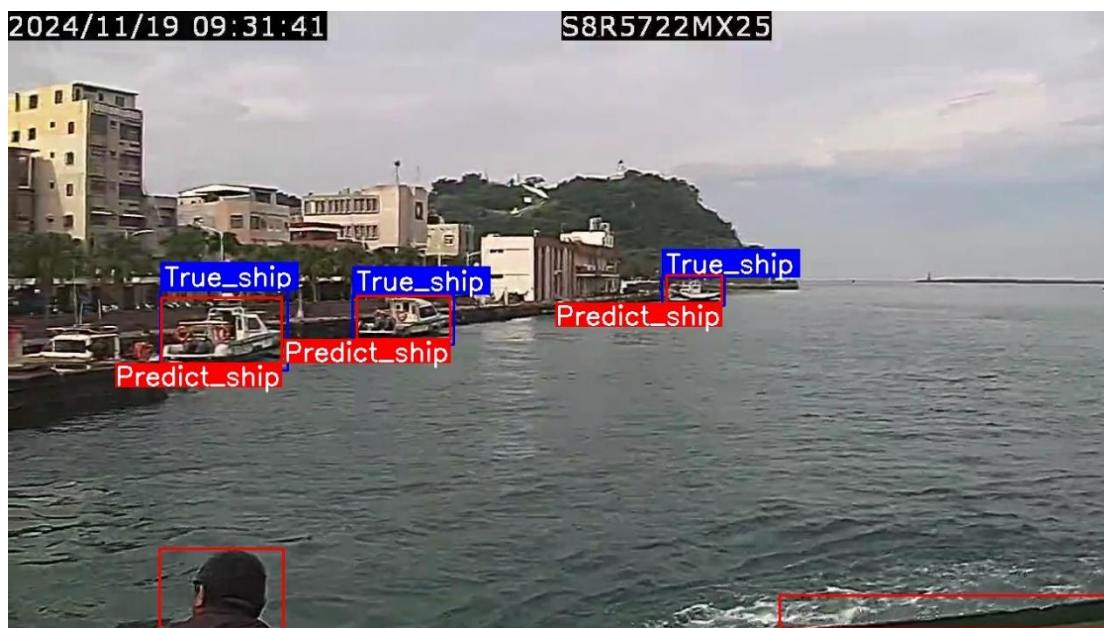
我們使用 YOLOv11l 模型，訓練集使用 ABOships-PLUS，測試集使用高雄資料集，實驗環境為 64-bit Windows 11、Intel i5-13400F 2.5GHz CPU、32GB RAM 及 NVIDIA RTX 4070 GPU。除了測試模型在原先新加坡資料集的效果外，實驗還在高雄資料集上針對船隻速度、距離以及避碰能力做檢測，因實際場域在白天時使用的是彩色影像，黑夜時使用的是黑白影像當作模型輸入，所以實驗也同時測試了模型對彩色和黑白影像的辨識能力，檢測方式為當船隻出現的 3 秒內有辨識到至少 1 次則辨識成功，反之辨識失敗，同一部影片中辨識成功的數量與真實船隻的數量比值即為檢測率。實驗時我們同時比較了 1 秒內有辨識到船隻、2 秒內有辨識到船隻的檢測率以更精確地測量模型的辨識能力，以下為檢測效果。

表四、實驗結果檢測率

測試影片	1 秒內檢測率	2 秒內檢測率	3 秒內檢測率
彩色影片 output_005	1.0	1.0	1.0
彩色影片 output_041	1.0	1.0	1.0
彩色影片 output_080	1.0	1.0	1.0
彩色影片 output_097	1.0	1.0	1.0
黑白影片 output_199	0.67	1.0	1.0
黑白影片 output_202	0.67	1.0	1.0
黑白影片 output_276	1.0	1.0	1.0
黑白影片 output_290	0.75	0.75	1.0

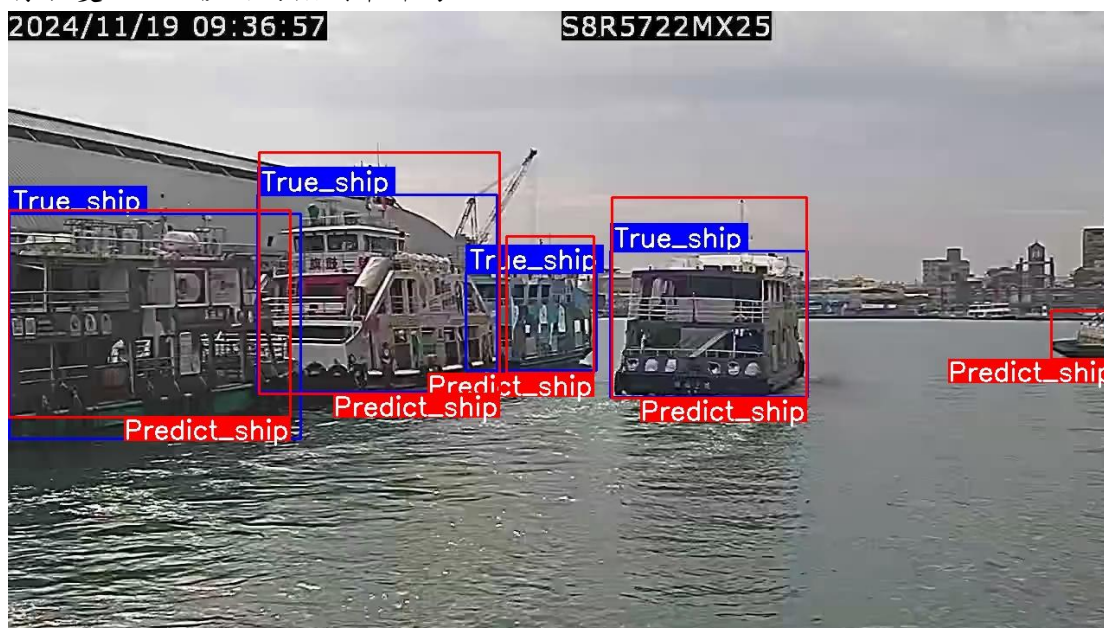
如上表所示，原要求的 3 秒內檢測率全部達標，都為 1.0，2 秒與 1 秒內的檢測率也大部分都是 1.0，其中判錯的情況都是在船隻彼此重疊時或影像較模糊時發生，一般的應用情況幾乎不會有問題。





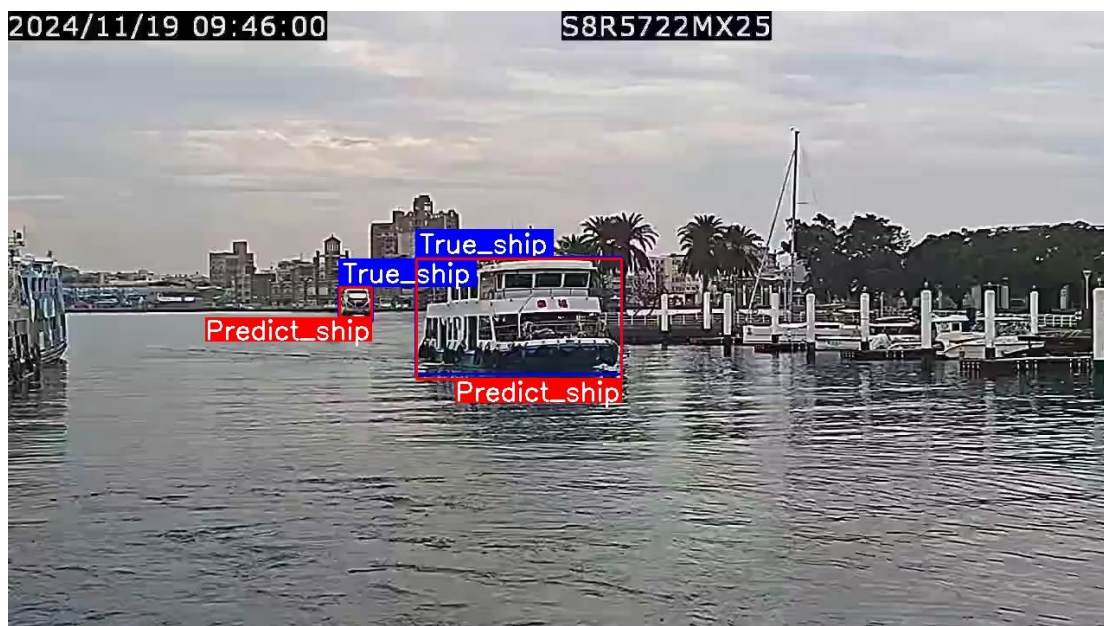
圖三、彩色影片 output\_005 第一秒影像

彩色影片 output\_005 的第一秒影像如上圖所示，圖中模型成功辨識出所有船隻，1~3 秒內的檢測率都為 1.0。



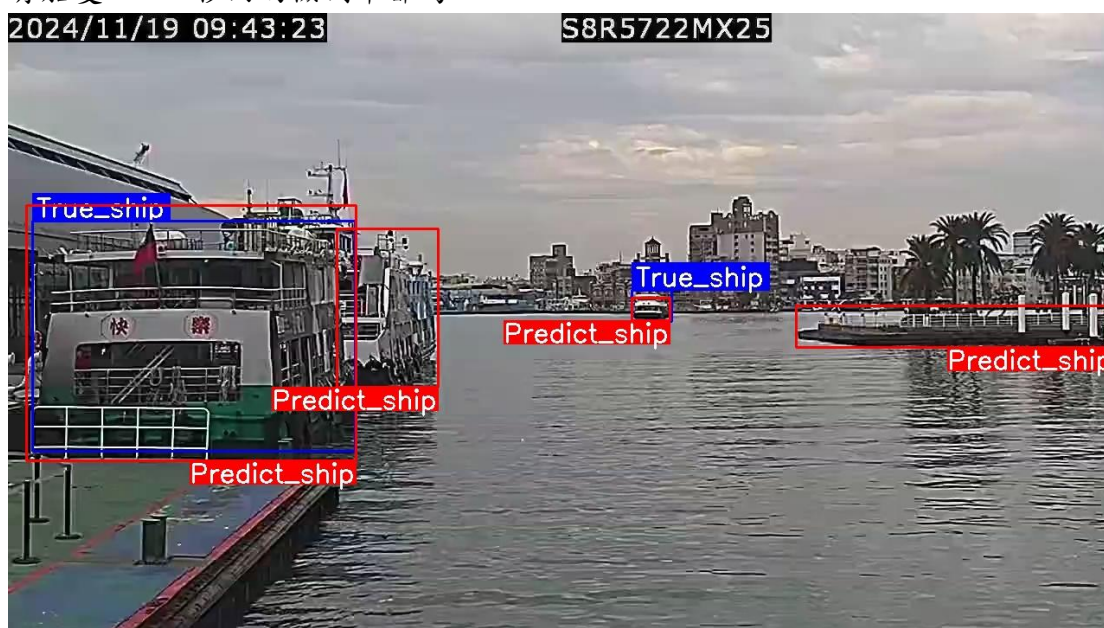
圖四、彩色影片 output\_041 第一秒影像

彩色影片 output\_041 的第一秒影像如上圖所示，圖中模型成功辨識出所有船隻，1~3 秒內的檢測率都為 1.0。



圖五、彩色影片 output\_080 第一秒影像

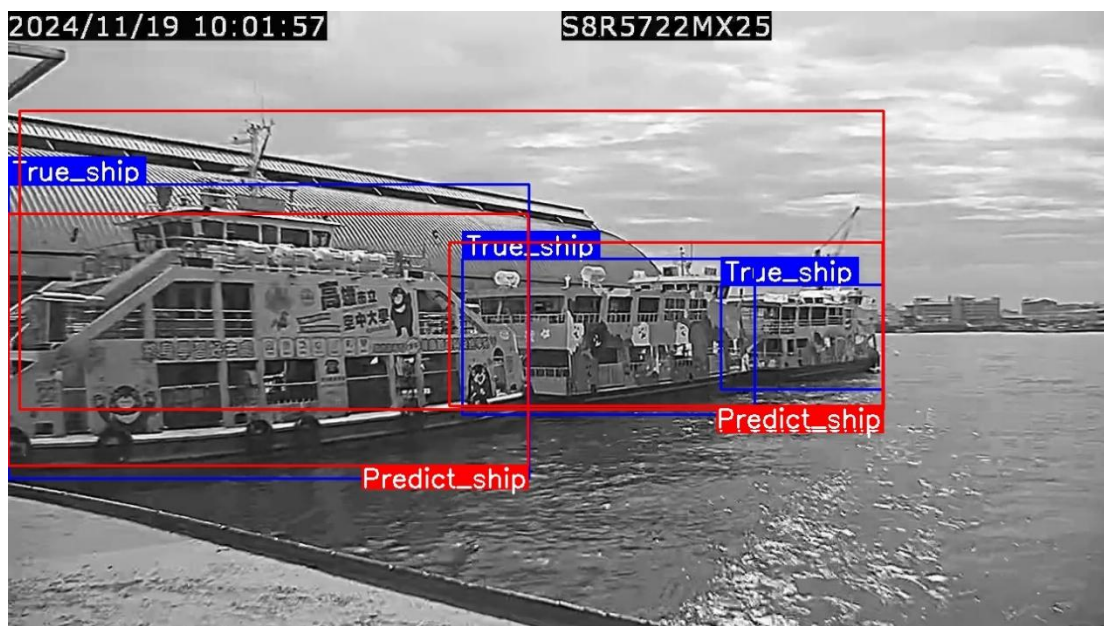
彩色影片 output\_080 的第一秒影像如上圖所示，圖中模型成功辨識出所有船隻，1~3 秒內的檢測率都為 1.0。



圖六、彩色影片 output\_097 第一秒影像

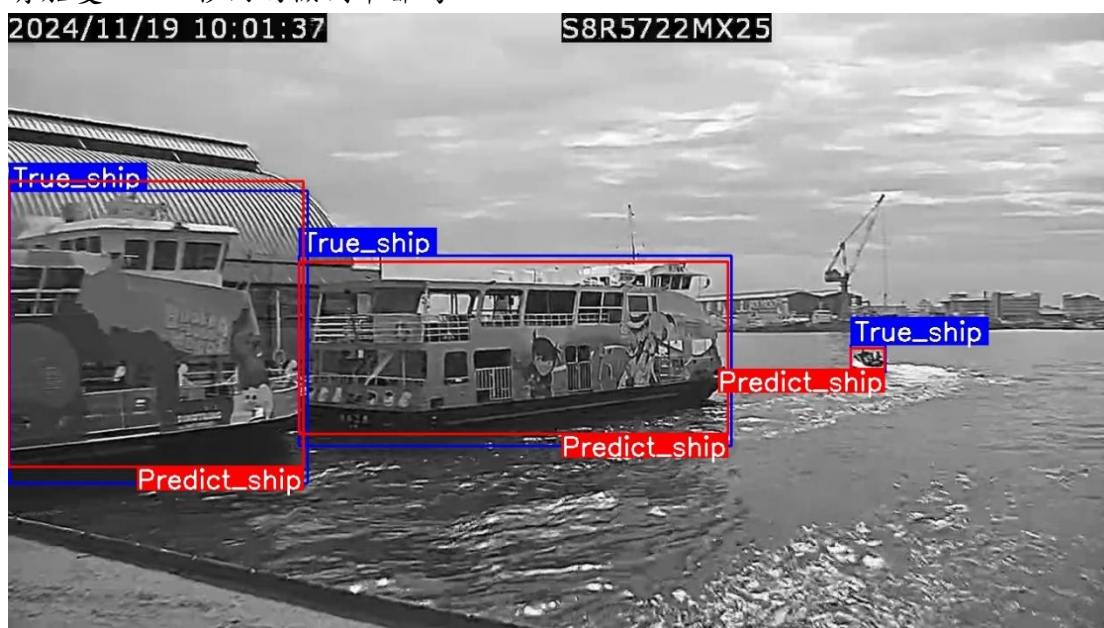
彩色影片 output\_097 的第一秒影像如上圖所示，圖中模型成功辨識出所有船隻，1~3 秒內的檢測率都為 1.0。





圖七、黑白影片 output\_199 第二秒影像

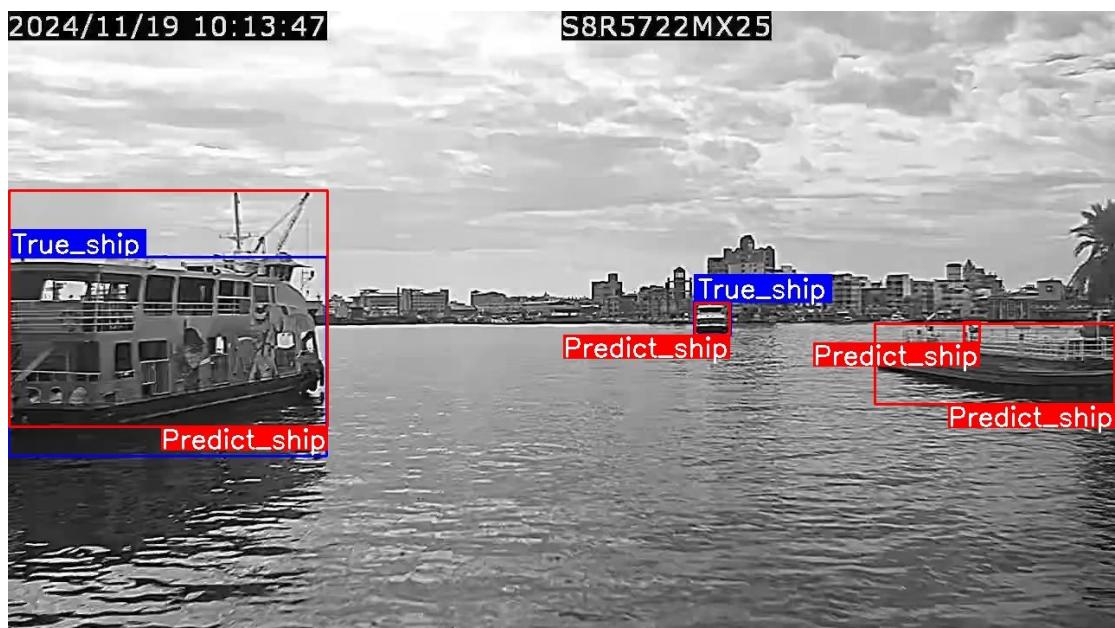
彩色影片 output\_199 的第二秒影像如上圖所示，圖中模型成功辨識出所有船隻，2~3 秒內的檢測率都為 1.0。



圖八、黑白影片 output\_202 第二秒影像

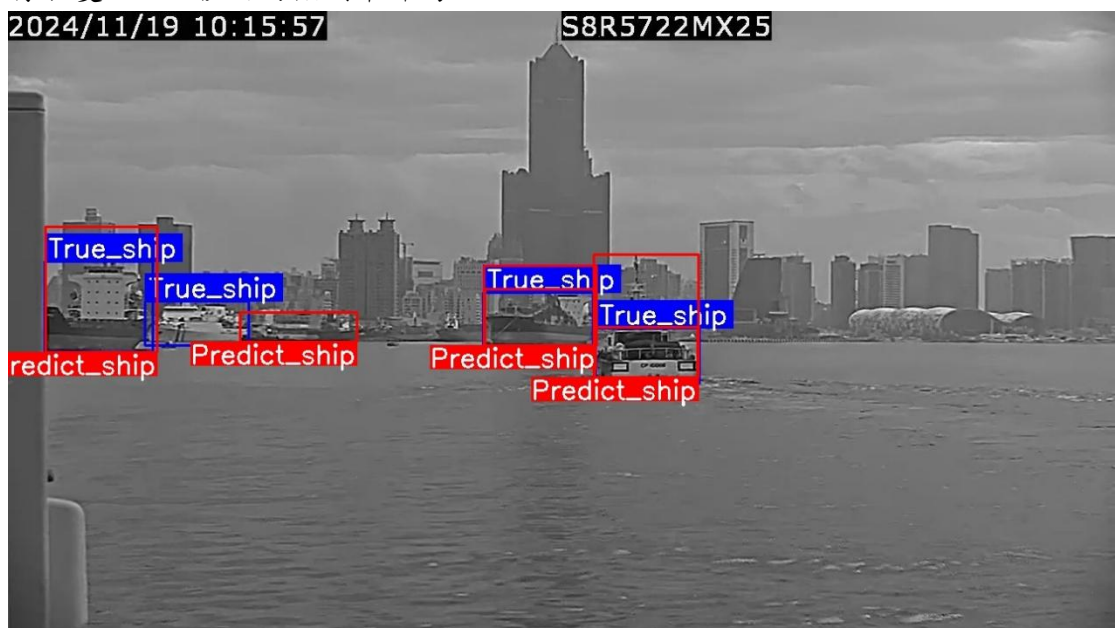
彩色影片 output\_202 的第二秒影像如上圖所示，圖中模型成功辨識出所有船隻，2~3 秒內的檢測率都為 1.0。





圖九、黑白影片 output\_276 第一秒影像

彩色影片 output\_276 的第一秒影像如上圖所示，圖中模型成功辨識出所有船隻，1~3 秒內的檢測率都為 1.0。



圖十、黑白影片 output\_290 第三秒影像

彩色影片 output\_290 的第三秒影像如上圖所示，圖中模型成功辨識出所有船隻，3 秒內的檢測率都為 1.0。

表五、KPI 完成表

項目	達成狀況
1.建立 AI 分析海事觀測演算法邏輯架構	已達成
2.完成演算法建立，並可在本島沿海 3 公里海域內環境進行	已達成
3.演算結果可達以下規格標準：	
通訊傳輸速率>250kbps 以上	已達成
影像解析度> 640x480	已達成
影像幀速率 $\geq$ 10FPS	已達成
船隻移動速度 $\geq$ 15 節(30km/hr)	已達成
船隻辨識數量 $\geq$ 10	已達成
船隻辨識大小> 15m	已達成
船隻辨識距離 $\geq$ 5km	已達成
船隻避碰距離 $\geq$ 3.7km	已達成
船隻辨識準確度 $\geq$ 90%	已達成

KPI 完成表如上所示，全部項目都已完成。

## 五、 結論

本計畫目的為設計出可以有效辨識 3 公里遠的船隻，且一次可辨識長度 15 公尺的船隻 10 艘以上的模型，根據目前的實驗結果，模型在船隻距離、速度、目標數量以及避碰能力上都有達到目標，並且模型在彩色及黑白影片上的表現也能夠符合實際應用場域需求。

## 六、 參考文獻

- [1] Sun, Bowen, et al. "Automatic ship object detection model based on YOLOv4 with transformer mechanism in remote sensing images." *Applied Sciences* vol. 13. No. 4, p. 2488, 2023.
- [2] Yang, Defu, et al. "A streamlined approach for intelligent ship object detection using EL-YOLO algorithm." *Scientific Reports*, vol. 14. No.1, p.15254, 2024.
- [3] Jiang, X., Cai, J., & Wang, B. (2024). YOLOSeaShip: a lightweight model for real-time ship detection. *European Journal of Remote Sensing*, 57(1).  
<https://doi.org/10.1080/22797254.2024.2307613>
- [4] He, Guowen & Wang, Wenlong & Shi, Bowen & Liu, Shijie & Xiang, Hui & Wang, Xiaoyuan. (2022). An Improved YOLO v4 Algorithm-based Object Detection Method for Maritime Vessels. *International Journal of Science and Engineering Applications*. 11. 50-55. 10.7753/IJSEA1104.1001.
- [5] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabaly, and C. Quek, "Video Processing from Electro-optical Sensors for Object Detection and Tracking in Maritime Environment: A Survey," *IEEE Transactions on Intelligent Transportation Systems (IEEE)*, 18 (8), 1993 - 2016, 2017.
- [6] D. K. Prasad, "Singapore Maritime Dataset, Visible On-Shore dataset," Available Online: <https://sites.google.com/site/dilipprasad/home/singapore-maritime-dataset>
- [7] Winsten Jesper, Iancu Bogdan, Soloviev Valentin, & Lilius Johan. (2024). ABOships-PLUS [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.10469672>
- [8] Glenn Jocher and Jing Qiu, "Ultralytics YOLO11," Available Online: <https://github.com/ultralytics/ultralytics>
- [9] Medium, "YOLOv11 Architecture Explained: Next-Level Object Detection with Enhanced Speed and Accuracy," Available Online: <https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71>