OAC-NTOU-109-012(研究報告)

應用深度學習於船舶影像分類研究

(正式報告)

海洋委員會補助研究

中華民國 109 年 10 月

「本研究報告僅供海洋委員會施政參考,並不代表該會政策,該會保留採用與 否之權利。」

應用深度學習於船舶影像分類研究

(正式報告)

學校:國立臺灣海洋大學

指導教授:張麗娜

學生:劉致圻

研究期程:中華民國 109 年 5 月至 109 年 12 月

研究經費:新台幣7萬元

海洋委員會補助研究

中華民國 109 年 10 月

「本研究報告僅供海洋委員會施政參考,並不代表該會政

策,該會保留採用與否之權利。」

圖次IV
表次V
摘要VI
第一章 前言1
第一節 研究緣起、問題背景與現況分析1
第二節 研究目的與重點1
第三節 預期目標2
第二章 YOLO 網路發展歷程3
第一節 YOLOv1
第二節 YOLOv26
第三節 YOLOv312
第三章 研究方法與過程16
第一節 研究資料來源介紹16
第二節 船舶資料數據集建置18
第三節 船體智慧偵測深度學習網路模型研究20
第四節 深度學習網路效能評估21
第四章 實驗結果與討論22
第一節 實驗結果22
第二節 討論25
第五章 結論27
參考文獻

圖次

圖 1 YOLOv1 流程圖[9]	
圖 2 YOLOv1 網路架構[9]	4
圖 3 批次正規化示意圖[1]	
圖 4 PASCAL VOC 與 COCO	數據集的邊界框聚類分析結果[10]7
圖 5 邊界框產生過程[10]	
圖 6 Darknet-19[10]	
圖 7 殘差架構示意圖[5]	
圖 8 拆分重組示意圖[18]	
圖 9 YOLOv2 架構圖	
圖 10 多尺度輸入示意圖[3]	
圖 11 YOLOv3 與其他網路效	能比較[11]13
圖 12 Darknet-53[11]	
圖 13 多尺度偵測示意圖[2]	
圖 14 流程圖	
圖 15 可見光紅外線影像比較	圖 (a)可見光影像 (b)紅外線影像17
圖 16 數據集中的船舶	
圖 17 利用 LabelImg 進行影伯	象標註19
圖 18 船舶分類系統	
圖 19 YOLOv3 的訓練過程圖	
圖 20 白天彩色船隻影像分類	結果
圖 21 晚上紅外線船隻影像分	·類結果25
圖 22 漁船偵測錯誤的影像	

表次

表	1 YOLOv1 至 YOLOv2 之改善方法列表[10]	6
表	2 YOLOv2 與其他方法的效能比較[10]	. 11
表	3 船舶影像資料集	.18
表	4 混淆矩陣(Confusion Matrix)	.21
表	5 系統環境	.22
表	6 YOLOv3 的效能與運算量	.22
表	7 YOLOv3 混淆矩陣	.26

關鍵字:船舶影像分類、深度學習、YOLO 網路、YOLOv3

一、研究緣起

台灣為海島國家,對於船舶的有效監控與管理實為發展智慧海運刻不容緩的 要務。近年來隨著深度學習網路的蓬勃發展,許多應用深度學習於目標物辨識也 獲得很好的成果。

二、目的

考量到船舶偵測的即時性與準確性,本研究將應用深度學習方法提升船舶影 像分類的效能。

三、研究方法與過程

對於深度學習而言,需要有足夠數量的資料集才能使分類網路有效的運作。 考量船舶監控需日夜進行,然而目前並沒有合適的船舶數據集能提供本研究使用。 因此本研究首先建立台灣船舶數據集,此數據集中除了常見的可見光影像外,也 加入紅外線影像,以提升於夜晚極低亮度下船舶分類的效能,並以此數據集進行 深度學習網路的訓練與船舶分類。

四、重要發現

本研究建置的台灣船舶數據集是由紅外線影像和可見光影像所構成的異質 數據集,意為同一種類的船舶影像會由兩種資料類型所構成。而根據實驗結果, 以 YOLOv3 為基礎的船舶分類網路不受異質數據集的影響,在分類效能上有很 好的表現,並在分類速度上擁有 YOLO 網路即時性的優點。

五、結論與展望

根據實驗結果,船舶分類網路可以成功的分類出本研究所建立6種不同種類的船舶,並且平均精度(mean average precision, map)達到 89.7%,分類速度可達 71.3 FPS(Frames Per Second)。藉由建置的船舶數據集及應用 YOLOv3 於船舶分 類,將有助於船舶管理與監控之實現。

第一章 前言

第一節 研究緣起、問題背景與現況分析

台灣為海島國家,海運和漁業是國家經濟的命脈。近年來國際海運需求量逐 年增加,使得港口的管理的業務量也越來越繁重,因此港口的管理勢必也要走向 智慧化、自動化的系統來精簡人力。而為了對船舶有效監控與管理,研發智慧化、 自動化的系統實為發展海運和漁業時刻不容緩之要務。

若要建立自動化、智慧化的系統必須要使用傳統的影像處理及電腦視覺之方 法來進行,但其缺點在於需要先透過人工經驗法則進行目標物相關參數的設定, 再進行目標物的特徵萃取,而若有多目標或目標不易辨識時,特徵萃取就需要投 入更多人力來進行,往往也會造成偵測效果的強健性不佳。近年來由於圖形處理 器(Graphics Processing Unit, GPU)的效能大幅提升以及動態隨機存取記憶體 (Dynamic Random Access Memory, DRAM)的單位容量成本大幅下降,需要大量記 憶體空間及強大的平行運算效能的深度學習網路得以蓬勃發展。目前深度學習網 路已被應用於語音識別、影像分類等領域,而其帶來的應用層面也非常寬廣,例 如在醫學上已有許多人成功的利用深度學習在電腦斷層影像上偵測出腦中風及 各種外傷性病灶,在自駕車領域特斯拉(TESLA)也是應用深度學習來建置其自動 駕駛系統,由此可見目前深度學習網路受廣泛應用的程度。而目前也有許多深度 學習網路架構是特別以物件偵測為目的所開發,而物件偵測也為本研究船舶影像 分類有很強的關聯,因此本研究應用這個方法,提出結合人工智慧的「船舶影像 自動分類」的研究。

第二節 研究目的與重點

此研究主要為應用現有基於深度學習概念的物件偵測網路架構進行船舶影 像分類,其重點為數據集的蒐集。數據集的蒐集對於深度學習領域來說是至關重 要的,若能建立一個豐富的訓練集對於深度學習網路的訓練過程是十分有幫助的, 不過現階段並沒有一個數據集能夠符合與本研究所需,原因在於以深度學習為基 礎的船舶偵測系統大多是以全偏極孔徑雷達影像為主,其以雷達波的反射特性獲 得影像,優勢為能在不受日夜、天氣的影響的情況下,都能獲得足以偵測船舶解 析度的影像。但是對於港口的船舶辨識而言,其解析度太低,只能從影像判讀出 有船舶,若要辨識出船的種類就會變得極其困難,因此本研究採用港邊、海上的 船舶影像,以獲得足夠清晰的船舶影像,並以此來建立本研究所需要的船舶數據 集。

第三節 預期目標

本研究會應用 YOLOv3,提出結合人工智慧的「船舶影像自動分類」的研究, 預期可達到以下四個目標。

- 蒐集台灣船舶影像資料,建置台灣船舶數據集。將以配置紅外線設備的攝影 機蒐集基隆附近港口的可見光和紅外線影像,建置船舶數據集將涵蓋軍艦、 帆船、郵輪、貨櫃輪、漁船和遊艇等六類船舶的日夜影像。
- 應用深度學習 YOLOv3 網路模型於船舶影像分類,完成船舶偵測深度學習模型研究。
- 電腦實驗模擬並完成船舶分類效能評估。完成船舶影像分類的電腦模擬,並 以平均分類精確度、分類速度和計算複雜度評估船舶分類效能。
- 4. 預期研究成果將有助於智慧化船舶監控與港口管理。

第二章 YOLO 網路發展歷程

YOLO(You Only Look Once)網路近年來廣泛應用於影像目標物辨識,本研究也將探討 YOLO 網路於船舶分類的效能。YOLO 模型的特性只需要對圖片做一次 CNN 便能夠判斷裡面的物體類別及位置,大大提升辨識速度。相較於 R-CNN,是先提出幾個可能性區域,再針對每個區域使用 CNN 做分類,最後再以回歸修正 Bounding Box 的位置,這樣的方法速度較慢且不好訓練,而 YOLO 的好處在於是單一網路設計,判斷結果會包含 Bounding Box 位置以及每個 Bounding Box 所屬類別和機率,整個網路是屬於 end-to-end 的,此種方法是屬於較易訓練且速度較快的。目前 YOLO 有三個版本,第一個版本在提升速度下犧牲了一些精度,因此第二個版本引入了其他深度學習方法進行改進,第三個版本則進一步將網路加深,提升準確度。

第一節 YOLOv1

YOLOv1[9]將先前目標偵測所使用的各個部分統一為一個單獨的神經網路, 並將整張圖片作為特徵預測出所有的 Bounding Box,並同時預測出 Bounding Box 框顯區域的物種類別,其流程圖如圖 1 所示。



圖 1 YOLOv1 流程圖[9]

首先 YOLOv1 會先將圖片分為S×S的網格,而若某個對象的中心在某個方格內,則那個方格就需就須負責對目標偵測,並且做定位與分類的兩個動作。定位的部分每一個網格會負責 B 個 Bounding Box 及他們的信任度,每個 Bounding Box 也包含 5 個預測值:x,y,w,h 和 confidence。x,y 代表 Bounding Box 的中心位置,w,h 為 Bounding Box 的長寬, confidence 則為信任度,其代表著網路偵測模

型對於所框出的目標有多少信心其定義方式如下:

$$confidence = P r(Object) \times IOU_{pred}^{truth}$$
(3)

*Pr(Object)*為 Bounding Box 內有目標的機率,*IOU*^{truth}則如圖 8 所示,為 Bounding Box 與 Ground Truth 的交疊率,而若網格內沒有任何目標偵測物體,則 *confidence* = 0°分類的部分,每個網格則會預測 C 個條件機率,其表示式如下:

$$pr(class_i|Object) * Pr(object) * IOU_{pred}^{truth} = Pr(class_i) * IOU_{pred}^{truth}$$
 (4)

此公式即是在計算出confidence後,再乘上目標種類的條件機率,即可得 到每個 Bounding Box 在每個類別上的得分。

YOLOv1 的網路架構如圖 2 所示,是以 GoogLeNet 作為基底,有著 24 個卷 積層及 2 個全連接層,位於架構前端的卷積層負責提取影像中的特徵,全連接層 則負責預測輸出的機率與座標。





由此架構圖可以得知,會先將輸入影像尺寸調整為448×448後,會進行三個 三個尺寸 7×7、3×3、1×1 卷積核進行卷積,其中 7×7 與 3×3 的卷積核用於萃取 特徵,1×1 的卷積核則是負責降維,減少運算量。經過 24 層卷積層後,最後加 入 2 層全連接層,第一個全連接層是將卷積層萃取的特徵圖轉為 1×1×4096 的一 維向量,第二層全連接層則是將第一層輸出的一維向量轉為 7×7×30 的向量,而 這裡的 7×7 是根據作者在 PASCAL VOC 數據集評估 YOLO 效能時是將影像分為 7×7 的網格,30 則是根據 B×(5+C)而來,B 為每個網格會產生 Bounding Box 的 數量,5 為 Bounding Box 中心位置 x,y,長寬 w,h 及信任度confidence,C 的部 分則為總共有幾個類別,而這裡由於 PASCAL VOC 數據集有 20 個類別,因此 C=20,這也是最後輸出 7×7×30 的原因。

YOLOv1 損失函數的設計方針是希望讓 Bounding Box 中心座標 x,y,長寬 w,h, confidence信任度及分類這四個方面可以達到一個很好的平衡,因此採用

平方誤差總和(Sum-Squared Error, SSE)來計算損失,其優點在於可以很簡單的被優化,損失函式如下:

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
(5)
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\ + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

損失函式的結構大概分為五個部分,上述的損失函式中,第一列為 Bounding Box 中心座標的預測,第二列為 Bounding Box 長寬的預測,第三列為confidence 信任度的預測,第四列為未包含偵測目標的 Bounding Box 的confidence信任度 的預測,第五列為類別預測,其中 l_i^{obj} 代表當有目標出現在網格 i 時, l_{ij}^{obj} 代表第 i 個網格中第 j 個 Bounding Box 包含目標物的狀況, l_{ij}^{noobj} 為第 i 個網格中第 j 個 Bounding Box 不包含目標物的狀況。不過若簡單的採用平方誤差總和會有缺點 存在,例如網格內沒有目標,而通常在一張影像當中,多數的網格內都是這種情 況,這會導致這些網格內 Bounding Box 的confidence = 0,相較於少數有目標 的網格,這些不包含目標的網格對梯度更新的貢獻就會遠大於有目標的網格,而 這樣的情況就會造成整個網路不穩定甚至發散,因此為了解決這個問題,加入了 兩個權重參數 λ_{coord} 及 λ_{noobj} 。作者在 PASCAL VOC 的實驗當中,由於更重視第 一列及第二列的中心座標與長寬預測,因此將 λ_{coord} 設為 5,提高了在整體損失 函數的權重,另外希望不包含目標的網格能夠減少對整體損失函數的貢獻,因此 將 λ_{noobj} 設為 0.5,降低了在整體損失函數的權重。

第二節 YOLOv2

YOLOv1 雖然速度較之前如 R-CNN、Fast-RCNN 的方法來的更快,但是準 確度卻沒有更好,且在目標的定位的準確度上也還有改善的空間,另外,每個網 格只偵測2個邊界框和1個類別,造成重疊或較小的目標物會較難被辨識,因此 YOLOv2[10]參考其他深度學習方法的概念,提出10項的改進,藉此近一步提高 YOLOv2 的準確度,以下表2即是 YOLOv1 至 YOLOv2 的改善比較表,並以 map(Mean Average Precision)作為效能指標來評估改善幅度。

表 1 YOLOv1 至 YOLOv2 之改善方法列表[10]

	YOLO								YOLOv2
batch norm?		~	~	~	\checkmark	\checkmark	\checkmark	\checkmark	√
hi-res classifier?			~	~	\checkmark	\checkmark	~	\checkmark	~
convolutional?				~	\checkmark	\checkmark	\checkmark	~	\checkmark
anchor boxes?				\checkmark	\checkmark				
new network?					\checkmark	\checkmark	\checkmark	\checkmark	~
dimension priors?						\checkmark	\checkmark	~	~
location prediction?						\checkmark	\checkmark	\checkmark	\checkmark
passthrough?							\checkmark	\checkmark	~
multi-scale?								\checkmark	~
hi-res detector?									\checkmark
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	78.6

首先,批次正規化(Batch Normalization)是為了神經網路進行倒傳遞的傳播過程中梯度消失與梯度爆炸的問題。一般來說,輸入層可以使用正規化將 Feature範圍加以限制,但此種方法的弊病是後面幾層的網路仍會因梯度的加深喪失輸入之間的資料差異敏感度,因此如圖 3 所示,會在權重與激勵函數(Activation Function)之間加入批次正規層,以解決這個問題,利用這樣的方式可以讓 map 進步 2.4%。





圖 3 批次正規化示意圖[1]

接下來 hi-res classifier 為高解析度分類器,其與 YOLOv1 的差別為, YOLOv1

會在訓練階段時採用 224×224 的解析度訓練分類網路,並以 448×448 的解析度 訓練檢測網路,這樣會造成神經網路必須同時在適應新的解析度的情況下,訓練 檢測網路。YOLOv2 的做法為,一樣先以 224×224 的解析度進行預訓練,但是會 再以 448×448 為解析度的輸入進行微調,使神經網路適應 448×448 的解析度,並 再以 448×448 的解析度訓練檢測網路。多了這個過渡的過程,可以使 map 再上 升 3.7%。

Convolutional With Anchor Boxes 即是參考 Faster R-CNN 的作法採用先驗 框的作法,在每一個網格預先設定一組不同大小與長寬比的 Anchor Box 來覆蓋 整個圖像的不同位置,並以這些 Anchor Box 作為候選區來檢測神經網路是否存 在目標,並微調邊框的位置。與 YOLOv1 相比, YOLOv1 並沒有採用先驗框的 概念,而是每個網格只預測兩個 Bounding Box,根據 YOLOv1 輸入圖片最終會 被劃分為 7×7 的網格為例,最終僅會有 98 個候選邊框;而 YOLOv2 採用 416×416 作為輸入的情況下,最終的特徵圖大小為 13×13,而若每個網格採用 9 個 Anchor Box,則會有上千個 Anchor Box,與 YOLOv1 相比相差許多。以結果來說雖然 map 下降了 0.3%,但是召回率(recall)由 81%上升到 88%。



圖 4 PASCAL VOC 與 COCO 數據集的邊界框聚類分析結果[10]

接下來改善的部分為 Anchor Box 尺寸的設定,雖然 Faster R-CNN 也有採用 鮮驗框的概念,但是其尺寸為手動設定,若尺寸、數量設置的好,深度學習網路 就能更容易學習,進而做出更好的預測,反之,若設置的不好,也會造成深度學 習網路訓練上的障礙,因此 YOLOv2 採用 K-means 聚類方法對訓練集中的邊界 框做聚類分析,以此自動設置 Anchor Box 的尺寸,而聚類方法最重要的就是如 何計算兩個邊界框的距離,對於常用的歐式距離會造成大邊界框產生更大的誤差, 因此 YOLOv2 改採用 IOU 作為評分標準,其計算方式如下 centroid為聚類時被選為中心的邊框,box為其他邊框,d則為兩者間的距離, IOU 越大,距離越近。採用此種方法可以使 map 有圖 4 為 PASCAL VOC 與 COCO 數據集的邊界框聚類分析結果,從圖中可以得知,雖然邊框數越多,平均 IOU 值 越大,但是過多的邊框也會造成訓練複雜度提高,因此 YOLOv2 選擇 5 個邊框。

而在使用 Anchor Box 的概念在 YOLO 上時,主要會遇到模型不穩定的問題, 尤其是在訓練早期的時候,由於 Anchor Box 可以在圖象中的任意一點,因此模 型隨機初始化的時候會需要很長的一段時間才能穩定預測目標的位置,因此 YOLOv2 沿用 YOLOv1 的作法,預測邊界框中心點相對於網格左上角位置的相 對偏移值,並以以下的公式計算邊界框的位置和大小。

$$b_{x} = \sigma(t_{x}) + c_{x}$$

$$b_{y} = \sigma(t_{y}) + c_{y}$$

$$b_{w} = p_{w}e^{t_{w}}$$

$$b_{h} = p_{h}e^{t_{h}}$$

$$(7)$$

 $Pr(object) * IOU(b, object) = \sigma(t_o)$

此公式的 $C_x n C_y$ 代表網格的左上角坐標, $p_w n p_h$ 為該網格預測之邊界框的 寬與高,並會同時讓每一個邊界框輸出5個值,分別為 $t_x \cdot t_y \cdot t_w \cdot t_h n t_o$, $t_x n$ t_y 為座標的偏移量, $t_w n t_h$ 為為寬高的偏移量, t_0 則是信任度。由於邊界框中心 點約束在網格當中,因此使用 sigmoid 函數處理 $t_x n t_y$ 偏移值,如此就能使預測 的偏移值落在 0~1 的範圍內,而在這裡會將每個網格的尺度定為 1。 $t_w n t_h$ 則是 取 exponential 函數,最終輸出 $\sigma(t_0)$ 產生信任度,其輸出的過程如圖 5 所示。



圖 5 邊界框產生過程[10]

YOLOv2 採用了一個新的特徵萃取網路架構,稱為 Darknet-19,其包含了 19 個卷積層及 5 個池化層,如圖 6 所示,其主要採用 3×3 的卷稽核進行卷積,並以 1×1 的卷稽核進行降維,以此特徵萃取架構雖然 map 並沒有顯著的提升,但主要 的目的為降低系統運算量。

Туре	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7 imes 7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

圖 6 Darknet-19[10]

訓練或檢測的過程當中,目標會有大有小,而在影像輸入經過多層的萃取特 徵後,較小的對象可能已經變的不明顯甚至是被忽略掉了,因此為了能檢測出較 小的目標,必須在最後輸出的特徵圖保留更為細節的特徵。YOLOv2 為此導入 Passthrough 層,其方法先參考 Resnet 網路的殘差架構(residual),將淺層的特徵圖 直接連接到深層的特徵圖,如圖7所示,並將淺層的特徵圖拆分重組,如圖8所 示,並與原生的深層特徵圖連接,經過這個方法後,深度學習網路即可在較深層 的網路中檢測出較小的目標。



圖 7 殘差架構示意圖[5]

1	2	3	4]
5	6	7	8	1
9	10	11	12	1
13	14	15	16	1

1	3	2	4	5	7	6	8
9	11	10	12	13	15	14	16

圖 8 拆分重組示意圖[18]

layer	fi	lters		si	ίzε	5				ir	ηρι	Jt				outp	but	ŧ.
0	conv	32	3	х	3	1	1	416	х	416	x	3	->	416	х	416	х	32
1	max		2	х	2	1	2	416	х	416	х	32	->	208	х	208	х	32
2	conv	64	3	х	3	1	1	208	х	208	х	32	->	208	х	208	х	64
3	max		2	х	2	1	2	208	х	208	х	64	->	104	х	104	х	64
4	conv	128	3	х	3	1	1	104	х	104	х	64	->	104	х	104	х	128
5	conv	64	1	х	1	1	1	104	х	104	х	128	->	104	х	104	х	64
6	conv	128	3	х	3	1	1	104	х	104	х	64	->	104	х	104	х	128
7	max		2	х	2	1	2	104	х	104	х	128	->	52	х	52	х	128
8	conv	256	3	х	3	1	1	52	х	52	х	128	->	52	х	52	х	256
9	conv	128	1	х	1	1	1	52	х	52	х	256	->	52	х	52	х	128
10	conv	256	3	х	3	1	1	52	х	52	х	128	->	52	х	52	х	256
11	max		2	х	2	1	2	52	х	52	х	256	->	26	х	26	х	256
12	conv	512	3	х	3	1	1	26	х	26	х	256	->	26	х	26	х	512
13	conv	256	1	х	1	1	1	26	х	26	х	512	->	26	х	26	х	256
14	conv	512	3	х	3	1	1	26	х	26	х	256	->	26	х	26	х	512
15	conv	256	1	х	1	1	1	26	х	26	х	512	->	26	х	26	х	256
16	conv	512	3	х	3	1	1	26	х	26	х	256	->	26	х	26	х	512
17	мах		2	х	2	1	2	26	х	26	х	512	->	13	х	13	х	512
18	conv	1024	3	х	3	1	1	13	х	13	х	512	->	13	х	13	x 1	024
19	conv	512	1	х	1	1	1	13	х	13	X	1024	->	13	х	13	х	512
20	conv	1024	3	х	3	1	1	13	х	13	х	512	->	13	х	13	x 1	024
21	conv	512	1	х	1	1	1	13	х	13	X	1024	->	13	х	13	х	512
22	conv	1024	3	х	3	1	1	13	х	13	х	512	->	13	х	13	x 1	024
23	conv	1024	3	х	3	1	1	13	х	13	X	1024	->	13	х	13	x 1	024
24	conv	1024	3	х	3	1	1	13	х	13	X	1024	->	13	х	13	x 1	1024
25	route	16																
26	conv	64	1	х	1	1	1	26	х	26	х	512	->	26	х	26	х	64
27	геогд					1	2	26	х	26	х	64	->	13	х	13	х	256
28	route	27 24																
29	conv	1024	3	х	3	1	1	13	х	13	X	1280	->	13	х	13	X1	024
30	conv	425	1	х	1	1	1	13	х	13	X	1024	->	13	х	13	х	425
31	detecti	ion																

圖 9 YOLOv2 架構圖

實際的實現方式由圖 9 的 YOLOv2 的完整架構圖所示,圖中第 25 層 route 16 即是代表來自第 16 層的 output 26×26×512,這個就是為 Passthrough 層的來 源,並在第 26 層經由 1×1 的卷積核降低通道數後,輸出 26×26×64,然後再於第 27 層進行重組拆分,變為 13×13×256。第 28 層為疊加第 27 層與第 24 層的輸出, 因此得到 13×13×1280,最後再經過 3×3 與 1×1 的卷積核進行卷積後,輸出 13×13×125 的特徵圖。

由於 YOLOv2 去掉了全連接層,因此 YOLOv2 的輸入影像就不限於 416×416, 而為了增加深度學習網路的穩定性,YOLOv2 加入了多尺度輸入來訓練網路,其 方法是每隔一定數量的 iteration 後,就會更改輸入影像的大小,而輸入影像的大 小為 32 的倍數,示意圖如圖 10 所示,輸出的特徵圖尺寸也就隨著輸入影像尺寸 的不同而更動。藉由採用多尺度輸入的方法,深度學習網路可以更好的適應不同 大小的輸入影像。根據作者所做的實驗,如表 3 所示,YOLOv2 在低解析度的圖 片檢測當中,其檢測速度較先前的其他發法都來的快,map 也保持一定的水準, 而當在高解析度的圖片檢測當中,YOLOv2 的 map 都比先前的其他方法都還好, 且在速度上也基本滿足實時辨識的要求。



圖 10 多尺度輸入示意圖[3]

Detection Frameworks	Train	mAP	FPS
Fast R-CNN	2007+2012	70.0	0.5
Faster R-CNN VGG-16	2007+2012	73.2	7
Faster R-CNN ResNet	2007+2012	76.4	5
YOLO	2007+2012	63.4	45
SSD300	2007+2012	74.3	46
SSD500	2007+2012	76.8	19
YOLOv2 288×288	2007+2012	69.0	91
YOLOv2 352×352	2007+2012	73.7	81
YOLOv2 416×416	2007+2012	76.8	67
YOLOv2 480×480	2007+2012	77.8	59
YOLOv2 544×544	2007+2012	78.6	40

YOLOv2 與 YOLOv1 一樣是使用平方誤差總和的概念建構損失函式,建構的概念一樣須假設每個網格至多含有一個 Ground Truth,並計算和 Ground Truth 匹配 Anchor Box 的坐標誤差及置信度物誤差,其計算公式如下:

$$loss_{t} = \sum_{i=0}^{W} \sum_{j=0}^{H} \sum_{k=0}^{A} \qquad 1_{Max \ IOU < Thresh} \ \lambda_{noobj} * \left(-b_{ijk}^{o}\right)^{2} \\ + 1_{t < 12800} \ \lambda_{prior} * \sum_{r \ \epsilon \ (x,y,w,h)} \left(prior_{k}^{r} - b_{ijk}^{r}\right)^{2} \\ + 1_{k}^{truth} \left(\ \lambda_{coord} \ * \sum_{r \ \epsilon \ (x,y,w,h)} \left(truth^{r} - b_{ijk}^{r}\right)^{2} \\ + \lambda_{obj} * \left(IOU_{truth}^{k} - b_{ijk}^{o}\right)^{2} \\ + \lambda_{class} * \left(\sum_{c=1}^{C} \left(truth^{c} - b_{ijk}^{c}\right)^{2}\right)\right)$$
(8)

首先,W,H分別代表輸出特徵圖的寬與高,A為先驗框數量,λ為損失函示 各個部位的權重係數。第一個部分為計算背景的置信度誤差,其方法為計算各個 預測框和所有 Ground Truth 的 IOU,並取最大值,倘若該值小於預先設定的 Threshold,則這個預測框就被視為背景,並計算非目標的置信度誤差。第二個部 分為計算先驗框與預測框的誤差,並只在訓練前 12800 個 iteration 計算,第三個 部分為計算 Ground Truth 與預測框之間各部分的誤差,如坐標、置信度和分類。 藉由上述的改善方法與新的損失函式,YOLOv2 的 map 較 YOLOv1 有顯著的提 升,並且在速度上也更快。

第三節 YOLOv3

YOLOv3[11]對比 YOLOv2 進一步加深網路和參考其他已推出的優化方法加以應用,目標進一步提升分類速度及準確度,並更加優化小目標的偵測。以 COCO 資料集為例,與其他網路相比, YOLOv3 在差不多的準確率當中可以達到更快的 響應速度,如圖 11 所示。



圖 11 YOLOv3 與其他網路效能比較[11]

YOLOv3 之所以可以進一步提高校的原因主要有三個部分,首先是採用一個 新的特徵萃取網路進行特徵萃取,該網路為基於 YOLOv2 的 Darknet-19 進行設 計,並進一步加深網路,達到有 53 層卷積層,因此也被命名為 Darknet-53,其架 構如圖 12 所示。除了以 3×3 的卷積核進行特徵萃取及 1×1 的卷積核進行降維外, 與 YOLOv2 相比雖然保留了 route 層的動作但刪去了拆分重組的動作,改以使用 shortcut 層。與 route 層不同的是, shortcut 的功能為當前經卷積得到的特徵圖與 指定較淺的特徵圖進行逐元素(element-wise)相加, route 層則是維度上的相加, 並不會影響原來裡面的元素。shortcut 層的使用目的在於避免網路過深造成特徵 消失而發生的梯度爆炸及梯度消失的問題。

	Туре	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	3×3/2	128×128
	Convolutional	32	1 × 1	
1×	Convolutional	64	3 × 3	
	Residual			128×128
- 25	Convolutional	128	3×3/2	64×64
1	Convolutional	64	1 × 1	
2x	Convolutional	128	3 × 3	
	Residual			64×64
8	Convolutional	256	3×3/2	32×32
	Convolutional	128	1 × 1	
8x	Convolutional	256	3 × 3	
	Residual			32×32
	Convolutional	512	3×3/2	16×16
3	Convolutional	256	1 × 1	
8×	Convolutional	512	3 × 3	
	Residual			16×16
3	Convolutional	1024	3×3/2	8 × 8
	Convolutional	512	1 x 1	
4x	Convolutional	1024	3×3	
	Residual			8 × 8
204	Avgpool		Global	
	Connected		1000	
	Softmax			

圖 12 Darknet-53[11]

接下來為多尺度特徵圖的預測,YOLOv3 參考了 Feature Pyramid Network(FPN)萃取大中小三個不同尺度的特徵,如圖 13 所示,不同於YOLOv2 直接在每個網格的特徵圖上使用 5 種不同形狀、大小的 Anchor Box,YOLOv3 在 三個不同大小的特徵圖上分別使用 3 個不同形狀、大小的 Anchor Box,總共 9 個 Anchor Box,並再使用 K-means 聚類方法計算 Anchor Box 的尺寸。YOLOv3 的預設 Anchor Box 尺寸是利用 PASCAL VOC 數據集而來,其尺寸為 (10×13),(16×30),(33×23),(30×61),(62×45),(59×119),(116×90),(156×98),(373×326)。 數據集當中,若輸入影像尺寸為 416×416,則會產生先產生 13×13 的特徵圖,並 在經由最鄰近內差發法的上採樣得到 26×26 特徵圖,並在與網路前半部的特徵圖 結合,完成後再繼續進行一樣的步驟得到 52×52 的特徵圖。在最小的特徵圖 13×13 上,其感受視野最大,因此適合檢測較大的目標因此用最大的 3 個 Anchor Box 尺寸(116×90),(156×98),(373×326),中間大小的特徵圖 26×26 感受視野也是中等 的,因此 Anchor Box 即選擇 3 個中等尺寸的大小(30×61),(62×45),(59×119),最大 的 特徵圖,其能感知最多的細節,因此採用最小尺寸的 Anchor Box(10×13),(16×30),(33×23)。



圖 13 多尺度偵測示意圖[2]

在分類上, YOLOv3 取消採用 softmax 分類器,原因在於類別標籤可能是重疊的,例如狗與動物,而 softmax 分類器的使用前提為,每個 Box 只能對應到一個分類,這會造成在使用上的限制,並往往與事實不相符,因此 YOLOv3 改採用 logistic 分類器,使一個 box 可以對應到多個分類,讓 YOLOv3 可以更好的應用 在更複雜的應用場景。

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j}^{obj} [(t_x - \hat{t}_x)^2 + (t_y - \hat{t}_y)^2 + (t_w - \hat{t}_w)^2 + (t_h - \hat{t}_h)^2] \\ + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j}^{obj} [-\log(\sigma(t_o)) + \sum_{k=1}^{C} BCE(\hat{y}_k, \sigma(s_k))] \\ + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j}^{noobj} [-\log(1 - \sigma(t_o))]$$

$$(9)$$

以上式為 YOLOv3 的損失函式,大概可分為 Bounding Box 的偏移量預測和 信心度預測,與 YOLOv2 不同的是,由於 YOLOv3 將類別分類器由 Softmax 改 為 Logistic,因此在損失函式的類別預測也從平方誤差總和改為 Binary Cross-Entropy(BCE),做二元分類預測,二元的意思為有這個類別或沒有這個類別,而 損失函式裡的總和即是代表針對所有可能出現的類別各做一次預測。

第三章 研究方法與過程

本研究將會先利用具有紅外線夜視功能的移動式攝影機在港口邊大量拍攝 船舶影像以建立船舶影像數據集,之後再利用此數據集進行船舶智慧偵測深度學 習模型訓練,並評估訓練模型是否合宜研究,其流程圖如圖 14 所示,並會在以 下做步驟詳細說明。



第一節 研究資料來源介紹

應用人工智慧技術於船舶偵測的最大瓶頸為各種類型船舶影像資料的收集。 比較常見的影像數據集如 PASCAL VOC2007[14]、CIFAR-10。PASCAL VOC2007 數據集有 20 個類別,其中船舶類有 791 艘船和一個標籤"boat",但其資料太少並 不數使用。CIFAR-10 數據集有 6000 張 32×32 張標籤為"ship"的圖像,但是每張 圖像都太小了,且只包含一個目標,因此不適合船隻部分被遮擋等實際應用。目 前可應用於檢測海上船舶的影像主要有四種主要類型:光學遙測影像[6]-[8][16][22][23],雷達影像[4][13][15][20][21][24],紅外線影像[17][19]和可見光影 像[12]。其中光學遙測影像因容易遭受海浪和雲層等天候的影響,很難在較長的 操作時間內實現即時監控。而雷達影像雖可不受天候狀況影響,因其是利用雷達 波的反射獲取影像的細節,但缺點為成像解析度不夠高,造成在雷達影像中僅能 辨識出有船,但不易辨識出船的種類,因此本研究會使用可見光影像搭配紅外線 影像建置船舶資料集以供訓練和測試使用。

由於現階段尚無相關單位可提供此研究所需之船舶影像資料集,但本研究將 使用深度學習進行船舶辨識,基於此技術須透過大量的船舶影像資料來訓練深度 學習網路模型,由於船舶偵測的應用且需在不分日夜、全天候下進行,此數據集 會在基隆港和海洋大學附近漁港:碧砂漁港、八斗子漁港和正濱漁港以移動式攝 影機 Sony AX700 蒐集全天候船舶影像資料,此台攝影機除了擁有 18 倍光學變 焦外,另還具有紅外線夜視功能,可以很好的應用在船舶的影像蒐集。若有不足 之處再以網路上的船舶影像進行補足,以利後續訓練的船舶偵測深度學習模型。

本研究蒐集的數據集主要包含可見光影像與紅外光影像,可見光影像主要來 獲取日間的照片或影片,能提供畫面清晰、色彩和紋理豐富的船舶影像。紅外光 影像主要由此次移動式攝影機內建的紅外線夜視模式來拍攝。雖然夜間影像也能 使用可見光影像搭配一般光源作為照明進行辨識,但是其需要很強的的投射照明, 並可能因此影響生態,因此採用紅外線影像是較為妥善的選擇。一般而言,紅外 線影像的獲取分為主動式和被動式兩種方式,主動式是利用紅外線燈進行補光, 規格分為 850nm 及 940nm 兩種波長的補光燈,850nm 波長的紅外線補光燈的穿 透性較 940nm 長,並在工作過程中會發出微弱的紅光,為一般具有紅外線夜視 功能的監視器所採用,940nm 的紅外線燈則是在工作過程中完全不會發出任何亮 光,適合夜間的秘密監控,不過本實驗的目的主要是以穿透性及移動方便性為考 量,因此採用 SONY HVL-LEIR1 具有可見光及 850nm 波長紅外線的補光燈進行 照明。

另外,以被動方式也能獲取是因為在絕對溫度 0K 以上都會有紅外線輻射, 因此就算在全黑狀態在沒有補光燈的情況下,紅外線影像多少還是能獲得一些畫 面細節。而一般攝影機的感光元件原本是可以感受到紅外光的,不過這樣拍攝出 來的影像就會和一般人類肉眼所見的影像相差甚遠,為了解決這個問題,一般的 攝影機的感光元件都會再加裝屏蔽紅外線的濾波器(IR cut filter),使攝影機的感 光元件盡量避免感測到紅外光,若是具有紅外線夜視功能的攝影機這個濾波器就 會變成控制式的,可見光模式就會啟用紅外線濾波器,反之則將其關閉。如圖 15 所示,在同一時間幾乎無照明亮度的樓梯間當中,可見光影像幾乎是全黑,而紅 外線影像中還能辨別出空間中的樓梯、扶手...等等。



圖 15 可見光紅外線影像比較圖 (a)可見光影像 (b)紅外線影像

第二節 船舶資料數據集建置

要利用深度學習來實現影像的物件偵測時,需有大量的數據集,本研究的數 據集主要有六大分類:軍艦、帆船、郵輪、貨櫃輪、漁船和遊艇,總計 5543 張 影像,各分類的數量如表4所示。由於海洋大學附近的漁船數量較多也較易拍攝, 因此數據集的漁船分類當中含有最多的紅外線影像447張,遊艇有30張,其餘 未加入紅外線影像的船舶分類是因為在拍攝上有其困難,如在疫情期間基隆港內 並無停泊郵輪,晚上並無出航的帆船且停泊在港內的帆船都是收帆的狀態...等。 而在獲取影像的過程中,也會盡量涵蓋有可能的成像變化,例如,不同的比例尺、 拍攝方向、背景環境等。

在建立自己的數據集時,初期都會使用人工的方式來手動標註。對此我們將 可以透過 LabelImg 的圖形介面標註邊界框(Bounding-Box)和標籤(Label), LabelImg 會直接將訊息標註訊息轉化為 txt 文件,如圖 17 所示,此格式將被之 後所用的訓練網路 YOLOv3 接受。標註完後的所有船隻圖像均帶有船型標籤和 高精度邊框,如此一來將可以把標註樣本加入網路做訓練。不過數據數量對於人 工智慧深度學習是非常重要的,因此本研究也會透過資料擴增(Data Augmentation) 的方式對原始影像進行裁切、隨機翻轉以及縮放等動作來增加樣本的多樣性。

	軍艦	帆船	郵輪	貨櫃輪	漁船	遊艇	
數量	1012	999	512	1013	951	1026	



表3船舶影像資料集



18



圖 16 數據集中的船舶



圖 17 利用 LabelImg 進行影像標註

第三節 船體智慧偵測深度學習網路模型研究

本研究會透過深度學習網路自行學習偵測有效船隻、船體特徵,並有效和快速的達成船體檢測的任務。在進行 YOLO 網路的研究之後,本研究選擇深度學習模型 YOLOv3 來進行物件偵測,其實現方法是將一張影像分成 M×M 個網格,若某目標物(Object)的中心落在此網格中,該網格就負責此目標物的預測。每個 網格要預測 N 個 Bounding Box,每個 Bounding Box 除了迴歸本身位置之外,還多了預測可信度(confidence),可信度表示預測的 Bounding Box 包含目標物的可 信度與 Bounding Box 預測的準確度,以下式表示,若 Object 落在網格中, pr(object) = 1,反之為 0, IOU_{pred}^{truth} (Intersection-Over-Union)表示預測出的 Bounding Box 與實際 Ground Truth 之間交集與聯集的比值。

$$Pr(object) \times IOU_{pred}^{truth}$$
 (10)

每個 Bounding Box 要預測(x, y, w, h)和可信度共 5 個值, (x, y)表示某一個物件在這個網格的中心座標,這個物件相對應的寬高分別為 w, h。每個網格還要預測類別資訊,記為 C 類。因 M × M 個網格中,每個網格要預測 N 個 Bounding Box 以及 C 個類別,輸出為 M × M × (5 × N + C)的一個張量(Tensor)。以本研究 YOLOv3 為基礎的船舶分類網路為例,YOLOv3 會在每個網格預測 3 個不同尺寸 大小邊界框,因此 N = 3。此外,本研究有六個目標分類的船舶,包含漁船、郵輪、貨櫃輪、帆船、遊艇、軍艦,因此 C = 6。

透過大量的船舶數據集讓模型進行學習、訓練後,再放入未參與訓練過程 的船舶影像,結果即能輸出能框出影像中船舶所在位置及辨識船舶類型,如圖18 所示。







圖 18 船舶分類系統

第四節 深度學習網路效能評估

本研究將在具有 NVIDIA RTX2080 顯示卡的高效能電腦進行實驗。輸入具有船體的可見光及紅外線影像,透過 Tensorflow 建置 YOLOv3 神經網路模型,訓練好後再將即時影像輸入訓練好的網路進行船體分類。

對於訓練模型的效能將使用精準度(Precision)、召回率(Recall)和平均精度 (Average Precision, AP)三個指標進行評估。精準度和召回率的定義如下:

$$Precision = \frac{TP}{TP + FP}$$
(11)

$$Recall = \frac{TP}{TP + FN}$$
(12)

其中,TP(True Positive),FP(False Positive),FN(False Negative)及TN(True Negative)的概念如表 5 的混淆矩陣(Confusion Matrix)所示,TP 代表真實樣本是目標且預測結果也是目標;FN 為真實樣本是目標,但預測結果不是目標;FP 為 真實樣本不是目標,但預測結果是目標;TN 為真實樣本不是目標,且預測結果 也不是目標。

	實際 YES	實際 NO
預測 YES	TP	FP
預測 NO	FN	TN

表4 混淆矩陣(Confusion Matrix)

在一般情況下,準確度和召回率是互相影響的,精準度高時召回率低,精準 度低時召回率高。對物件偵測系統,可用精準度與召回率這兩個值來判別對哪一 類別偵測較高,但要表示整體判別效能並不合適。Precision「所有被檢測為目標」 中「正確分類為目標」的比例;Recall 表「所有目標」中「正確分類為目標」的 比例。因精確度和召回率往往是無法同時兼顧,因為假設把所有樣本都判為此類 樣本,則精確度就會很低,但召回率此時為1;相反地如果將極少數的樣本判為 此類,能得到很高的精確度,但召回率就很低。平均精度(AP)這個指標則兼顧了 兩者,可以將它看成 Precision-Recall 曲線(PR-curve)下方的曲面面積,此指標可 清楚比較個別框架的偵測效能,用於評估此模型應用在不同場景之適合度。

另外於效能評估中使用 IOU 作為測量在特定資料集中檢測相應物體準確度 的一個標準, IOU 表預測框與真實資料的交集和聯集的比值。IOU 標準可用於 測量真實和預測之間的相關度,相關度越高, IOU 值越高。在目標檢測研究 中,會將認為預測的目標的 bounding box 和 Ground truth 進行 IoU 計算,如果 IOU 大於一個閾值(通常是 0.5),則認為此目標為 TP,如果小於此閾值則為

第四章 實驗結果與討論

第一節 實驗結果

本節的實驗環境如表 5 所示,並會採用平均精度(Mean Average Precision, map)、每秒幀數(Frames Per Second, FPS)及十億浮點數運算量(Billion Float Operations, BFLOPs)進行效能、速度、運算量的比較。實驗中學習率皆設定為 0.001, Batch size 設定為 48, Max Batches 設定為 20000。

OS	Ubuntu 18.04		
CPU	Intel i7-8700K		
RAM	DDR4 2666 16GB		
Graphics	NVIDIA RTX 2080		
Software	darknet		

表5 系統環境

本實驗的影像輸入尺寸為 480×480,測試 FPS 的值是在輸入 1080P 影片得 出,並在圖 19 附上 YOLOv3 的訓練過程圖。圖 19 的訓練過程圖之 x 軸為 batch 的次數, y 軸為對應到 Training Loss 的值,紅色的線為 map,藍色的線為 Training Loss,由圖中可以得知, Training Loss 屬於收斂狀態。

表 6 YOLOv3 的效能與運算量

網路效能值	YOLOv3		
BFLOPs	86.973		
FPS	71.3		
map	89.7%		



圖 19 YOLOv3 的訓練過程圖

根據實驗結果,應用 YOLOv3 的船舶影像分類網路可以正確的分類白天的 可見光及夜間的紅外線船舶影像。圖 20 與圖 21 的白天彩色影像及夜間紅外線 影像即是應用 YOLOv3 網路進行分類結果展示。



圖 20 白天彩色船隻影像分類結果



圖 21 晚上紅外線船隻影像分類結果

第二節 討論

 紅外線影像的蒐集為本研究較困難之處,如帆船,並不會在晚上出航,而停泊 在港區的帆船,帆布大都是收起來的,而帆是帆船重要的特徵,因此會對船舶分 類系統有不好的影像;而如郵輪,由於疫情的因素,並沒有郵輪停泊在基隆港內, 因此無法獲得紅外線影像。

2. 從表 7 的混淆矩陣可以觀察到漁船比其他種類的船精準度都差很多,原因在 於其他種類的船舶背景都較為單純,而漁船時大多是在漁港內拍攝,而漁港內 的場景較為複雜,取景上也較為困難,導致較難拍攝到單純或完整的漁船影 像,如圖 22 的照片所示,像這樣的影像就無法辨識出有漁船,因此使用背景較 為簡單的海上作業中漁船是比較好的選擇,但若要拍攝這樣的漁船,勢必就要 投入大量的人力與時間,因此本研究最終選擇在漁港旁拍攝。

YOLOv3							
\ge	貨櫃輪	郵輪	帆船	軍艦	遊艇	漁船	
貨櫃輪	103	1	0	2	0	0	
郵輪	1	95	0	2	2	0	
帆船	0	0	91	0	0	0	
軍艦	0	0	0	89	0	0	
遊艇	1	1	0	0	103	1	
漁船	0	0	0	0	0	79	
未辨識	8	8	9	11	6	32	
準確率	91.2%	90.5%	91%	85.6%	92.8%	70.5%	
總計	113	105	100	104	111	112	

表7YOLOv3 混淆矩陣



圖 22 漁船偵測錯誤的影像

第五章 結論

由於台灣四面環海且地理位置佳,因此海運與漁業一直以來都是我們相當重要的產業,隨著海運與漁業的蓬勃發展,船舶的數量也越來越多,而港口的管理 需仰賴人力對船舶進行出入港檢查,隨著這樣的發展趨勢下這些勤務對於巡防人 員造成龐大的負荷,因此建立一個及時、精確的智慧化船舶影像分類系統是非常 重要的。

在本研究當中,採用近年來蓬勃發展的深度學習方法應用於船舶影像分類, 考量到船舶影像分類系統有即時性需求,因此採用兼具速度與準確度的 YOLOv3 網路進行船舶影像分類的研究。在數據集影像的選擇當中,採用可見光影像及紅 外線影像。可見光影像有細節豐富,色彩紋理佳,但有夜間效能不足的缺點,因 此加入紅外線影像提升夜間分類效能。以利本研究船舶影像分類系統不分日夜下 的進行。

本研究透過應用 YOLOv3 能在船舶分類的準確度及分類速度取得很好的成 果。根據實驗結果,分類速度達到 71.3 FPS,平均精度達到 89.7%。整體而言透 過加入紅外線影像及應用 YOLOv3 可有效優化船舶影像分類系統於夜晚時的表 現並獲得很好的分類速度,有助於智慧航運的實現。

27

參考文獻

[1] Leyan Bin Veon, "YOLO v2 物件偵測~論文整理" May 2019.

< https://medium.com/程式工作紡/yolo-v2-物件偵測-論文整理-a8e11d8b4409>

[2] YOLOv3-引入:FPN+多尺度偵測(目標檢測)(one-stage)(深度學習)(CVPR 2018) <<u>https://blog.csdn.net/Gentleman_Qin/article/details/84350496</u>>

[3] 目標檢測 | YOLOv2 原理與實現(附 YOLOv3)

<https://blog.csdn.net/hejin some/article/details/80581789>

[4] 張麗娜、陳威霖, "應用階層式影像切割技術於 SAR 影像船舶偵測,"船舶 科技, 第五十期, pp. 1-13, 2018.

[5] 殘差網路的理解 <<u>https://www.itread01.com/content/1545338181.html</u>>

[6] C. Zhu, H. Zhou, R. Wang, J. Guo, and J. A Novel, "Hierarchical method of ship detection from spaceborne optical image based on shape and texture features," IEEE Transactions on Geoscience & Remote Sensing, Vol. 48, pp. 3446-3456, 2010.

[7] G. Liu et al., "A new method on inshore ship detection in high-resolution satellite images using shape and context information," IEEE Geosci. Remote Sens. Lett., Vol. 11, No. 3, pp. 617-621, Mar. 2014.

[8] J. Xu, X. Sun, D. Zhang, and K. Fu, "Automatic detection of inshore ships in highresolution remote sensing images using robust invariant generalized Hough transform," IEEE Geosci. Remote Sens. Lett., Vol. 11, No. 12, pp. 2070-2074, Dec. 2014.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: unified, real-time object detection," In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp. 779-788, 2016.

[10] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517-6525, Nov. 2017.

[11] J. Redmon and A. Farhadi "YOLOv3: an incremental improvement," Computer Science, arXiv 1804.02767, 2018.

[12] M. Ren and Z. Tang, "One effective method for ship recognition in ship locks," Proc. SPIE, Vol. 3720, pp. 467-472, Apr. 1999.

[13] M. Tello, C. Lopez-Martinez, and J. J. Mallorqui, "A novel algorithm for ship detection in SAR imagery based on the wavelet transform," IEEE Geosci. Remote Sens. Lett., Vol. 2, No. 2, pp. 201-205, Apr. 2005.

[14] M. Everingham, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," Int. J. Comput. Vis., Vol. 88, No. 2, pp. 303–338, 2010.

[15] M. U. Selvi1 and S. S. Kumar, "A novel approach for ship recognition using shape and texture," International Journal of Advanced Information Technology, Vol. 1, No. 5, pp. 23-29, 2011.

[16] R. Zhang, J. Yao, K. Zhang, C. Feng, and J. Zhang, "S-CNN ship detection from high-resolution remote sensing images," in Proc. Int. Congr. Arch. Photogramm., Remote Sens. Spatial Inf. Sci., pp. 423–430, 2016.

[17] S. R. Rotman, "Region-of-interest-based algorithm for automatic target detection in infrared images," Opt. Eng., Vol. 44, No. 7, pp. 166-169, Jul. 2005.

[18] Santosh GSK, "Training Object Detection (YOLOv2) from scratch using Cyclic Learning Rates," <<u>https://towardsdatascience.com/training-object-detection-yolov2-from-scratch-using-cyclic-learning-rates-b3364f7e4755</u> > Mar. 2018.

[19] W. Tao, H. Jin, and J. Liu, "Unified mean shift segmentation and graph region merging algorithm for infrared ship target segmentation," Opt. Eng., Vol. 46, No. 12, pp. 127002-1-127002-7, 2007.

[20] X. Xing, K. Ji, L. Kang, and M. Zhan, "Review of ship surveillance technologies based on high-resolution wide-swath synthetic aperture radar imaging," J. Radars, Vol. 4, No. 1, pp. 107-121, 2015.

[21] X. Wang and C. Chen, "An automatic ship detection method based on local graylevel gathering characteristics in SAR Imagery," Electronic Letters on Computer Vision and Image Analysis, Vol. 12, No. 1, pp. 33-41, 2013.

[22] X. Yang, H. Sun, K. Fu, J. Yang, X. Sun, M. Yan, and Z. Guo, "Automatic ship detection of remote sensing images from google earth in complex scenes based on multi-scale rotation dense feature pyramid," Remote Sens., Vol. 10, No. 1: 132, 2018.

[23] Y. D. Yu, X. B. Yang, S. J. Xiao, and J. L. Lin, "Automated ship detection from optical remote sensing image," IEEE Geoscience & Remote Sensing Letters. Vol. 9, pp. 749-753, 2012.

[24] Y. L. Chang, A. Anagaw, Lena Chang, Y. Wang, C. Hsiao, and W. Lee, "Ship detection based on YOLOv2 for SAR imagery," Remote Sensing, MDPI, Vol. 11, No. 7: 786, 2019.