國立臺灣大學工學院土木工程學系

碩士論文

Department of Civil Engineering

College of Engineering

National Taiwan University

Master's Thesis

卷積網路應用於高畫質無人機影像中小物件偵測的優化辦法

An Advanced Technique for Enhancing the Performance of Convolutional Neural Networks in Small Object Detection with High Resolution UAV Image

林育銓

YU-CHUAN LIN

指導教授: 韓仁毓 博士

Advisor: JEN-YU HAN Ph.D.

中華民國 113 年 8 月

August, 2024

doi:10.6342/NTU202404011

誌謝

在本碩士論文的研究與撰寫過程中，我深感一路上所獲得的幫助和支持的重要性。在此，我衷心感謝所有給予我指導和幫助的人。

首先，我要特別感謝我的指導老師，老師不僅在學術上給予我指導，在跟隨老師的這幾年裡，老師的一言一行在各種面向上都給予我不少的啟發。而老師的專業知識和敏銳的學術洞察力，使我在研究中能夠不斷進步和成長，並打好學術研究的堅實基礎。

感謝實驗室的所有同學們，他們的友好合作和互相支持，為我提供了一個良好的環境。在日常的學術交流和討論中，我獲得了許多寶貴的意見和建議。他們同時也是我人生中值得敬愛的朋友，祝福各位朋友的未來，期許大家都能在自己的人生尋覓出一片天地。

此外，我還要感謝研究室的助理，雖然常常笑我延畢，但碩士生涯中能有這樣的朋友倒也增添了不少樂趣，同樣祝福各位的未來。

最後，感謝我父母對我的愛和支持。他們一直是我最堅強的後盾。感謝母親的關心，感謝父親來台北時總會找我一起吃飯，即便我都不回家也還是常常關心我，真是十分感謝。

再次感謝所有支持和幫助過我的人，是你們讓這段旅程變得意義非凡。

doi:10.6342/NTU202404011

中文摘要

無人機作為遙測的新興工具，若能結合深度學習進行自動化物件偵測，將能使人們對於無人機的應用範疇有更深的拓展與運用。然而無人機影像與普遍影像不同，其具備高解析度與目標物相對尺寸較小的特性，在應用卷積神經網路技術時，會面臨到高解析度導致的 GPU 記憶體容量不足的限制。過往研究在解決這項限制時多以降低影像解析度的方式來減少 GPU 記憶體的需求，但這樣的做法會使目標物的特徵變模糊且稀缺，導致小物件偵測的困境。本研究提出一依照 Focus-and-Detect 概念的優化辦法，先對影像進行子區域提出，後再針對子區域進行偵測。在沒有降低解析度的情況下減少模型輸入影像的尺寸，減少 GPU 記憶體需求的同時維持無人機具備的高解析度，進而在相對尺度較小的目標物也能有足夠清晰的特徵，避免了小物件偵測的困境。在子區域提出的部分，本研究發展一子區域提出演算法 Modified Selective Search。演算法所提出的候選區域能具備理想子區域的特性，大小適當且包含多個目標物。本研究透過子區域提出方法所訓練的 yolov8 模型，在 VisDrone 無人機影像資料集中的高解析度影像上，達到比透過降低影像解析度方式所訓練的 yolov8 模型更好的表現結果。在測試資料的評估上增加了 12% 的 mAP 表現，並且在特徵相似的物件中有更好的區分效果。然子區域提出的方法會需要更多的訓練時間與儲存空間來儲存子區域影像，總結來說是一以時間換取表現以及資源的方法。

關鍵字：無人機影像、深度學習、卷積神經網路、小物件偵測、區域提出

doi:10.6342/NTU202404011

Abstract

Drones, as an emerging tool in remote sensing, already possess extensive application potential in the field of civil engineering. The integration of deep learning for automated object detection could further expand the applications of drones and enhance their utility. However, drone imagery differs from conventional computer vision images, characterized by high resolution and relatively small object sizes. When applying convolution neural networks, this leads to constraints due to GPU memory limitations caused by high resolutions. Previous studies often addressed this limitation by reducing image resolution, which blurs and sparsifies the features of small objects, thus complicating small object detection. This study proposes an optimization approach based on the Focus-and-Detect concept, which involves segmenting the image into sub-regions before processing object detection. This method reduces the need for GPU memory without sacrificing image resolution. Furthermore, the high-resolution capability of drones ensures that even relatively small objects possess sufficient feature content, making the sub-region proposal method

doi:10.6342/NTU202404011

particularly suited to drone imagery. In this study, we modify the Selective Search algorithm to propose an optimized sub-region proposal method, Modified Selective Search. The proposed candidate regions by this algorithm have the ideal characteristics of sub-regions, being appropriately sized and containing multiple objects. The yolov8 model trained using this sub-region proposal method achieved superior performance on the high-resolution VisDrone drone imagery dataset compared to models trained with reduced-resolution images, improving the mAP by 12% in test data evaluations and exhibiting enhanced differentiation among vehicles, trucks, and vans, which have similar features. While the sub-region proposal method requires additional training time and storage space, it essentially trades time for improved performance and lower GPU memory requirement.

Keywords: UAV imagery, deep learning, CNN, small object detection, regional proposal

doi:10.6342/NTU202404011

目次

doi:10.6342/NTU202404011

doi:10.6342/NTU202404011

圖次

doi:10.6342/NTU202404011

doi:10.6342/NTU202404011

doi:10.6342/NTU202404011

表次

doi:10.6342/NTU202404011

第一章　緒論

1.1　研究背景

近年來無人機的硬體設備有著蓬勃的發展，搭載光學相機的無人機已是非常普及的工具。其迅速、便捷、高可及性與廣域的搜集能力使無人機無論是在土地調查 (Akar, 2018; Chen et al., 2020)、設施巡檢 (Lattanzi and Miller, 2017)、精準農業 (Delavarpour et al., 2021)、救災 (Silvagni et al., 2017) 或是災損評估 (周巧盈 et al., 2020) 上都能有所發揮，更重要的是無人機能提供人所不能及的鳥瞰視角，而這過去僅能由航空拍攝或衛星拍攝才能獲取。無人機具備高度的彈性與便利性，使其適合各式各樣的下游任務 (downstream task)，多樣的任務都能靠無人機得到來自天空的俯視資料。而且相較於航空與衛星的飛行高度，無人機的飛行高度低上許多，配合高畫質的光學相機，能夠獲取空間解析度更高的地面影像，在應用上能延伸的更廣 (Iizuka et al., 2018)。筆者認為無人機是資料獲取的未來方向，尤其在土木工程領域上，相關的應用在未來只會越多且越受重視。但無人機終歸只是一種蒐集資料用的工具，如何從資料中獲取資訊是另一大課題。

近幾年中，深度學習在電腦視覺任務上取得了重大的突破，Krizhevsky et al. (2012) 透過卷積神經網路 (Convolution Neural Network) 的設計，成功在影像辨識任務上獲得了突破性的結果，自此之後，卷積神經網路應用在影像上的研究層出不窮 (Gu et al., 2018; Li et al., 2021a; Yamashita et al., 2018)，並被證實能夠獲得更甚

doi:10.6342/NTU202404011

以往的結果。卷積神經網路透過卷積計算將影像的空間資訊做特徵提取，並透過機器學習中的神經網路架構進行自動化學習，使模型從相對應的輸入與輸出間找出問題的解。常見的電腦視覺任務都能套用此模式進行，諸如影像辨識、物件辨識亦或是影像分割。卷積神經網路的優勢在於其對空間特徵的提取與透過機器學習實現的複雜模型架構，更好的去適應了影像任務的複雜性與多變性。但相對應的缺點則是卷積神經網路需要較多的運算資源，當今的卷積神經網路模型絕大部分都需要依靠圖形處理器 (GPU) 加速運算才能在有效的時間內完成。卷積神經網路因其近乎於人的辨識效果與可自動化的發展性，已然是進行影像分析的最好辦法之一。

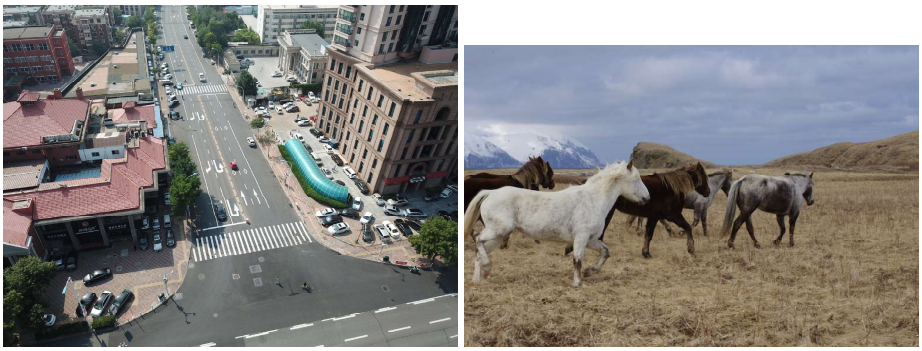
無人機影像若能結合卷積神經網路進行電腦視覺任務，尤其是物件偵測，將能提供非常多樣化的應用 (Osco et al., 2021; Tijtgat et al., 2017; Xu et al., 2017)。但無人機影像與電腦視覺任務常處理的普遍影像有所不同，此處所述之普遍影像是指在電腦視覺領域中常見的影像，通常為地面視角的影像且影像中的目標物較大，同時影像解析度一般在 1000 × 1000 pixels 以下。無人機影像為發揮無人機的優勢，在拍攝時大多是從高空中往地面進行拍攝，且距離地面有一定高度。為了在遠距離拍攝的情況下依然能提供清晰的地面資訊，無人機搭載的光學相機通常都會拍攝高解析度的影像，HD、2K、4K 甚至 8K 的影像都是現如今常見的無人機影像規格。但即便影像有高解析度，若要論相對尺度，無人機影像中的目標物相對於影像本身來說是依然相對較小的。統整來說，無人機影像與與普遍影像的最大不同在於：

1. 無人機影像有較高的解析度。
2. 無人機影像中目標物的相對尺寸較小。

圖1.1中左圖1.1a為無人機影像，出自 VisDrone2019(Du et al., 2019)，影像尺寸為

doi:10.6342/NTU202404011

2000×1500 pixels，其中的目標物－人與車，個別僅佔影像中不足 1% 的面積，相反的右圖1.1b為普遍影像，出自 COCO Dataset(Lin et al., 2014)，影像尺寸為 773×512 pixels，其中的目標物－馬，個別都有較大的佔比面積，兩相比較可見無人機影像與普遍影像之間的差異。上述提到的這兩項不同使卷積網路若要應用於無人機影像上時會有所限制與挑戰。



(a) 無人機影像範例圖片　　(b) 普遍影像範例圖片

圖 1.1、無人機影像與普遍影像的比較。無人機影像為鳥瞰視角，目標物較小。普遍影像為地面視角，目標物較大。

無人機影像應用卷積神經網路的限制在於 GPU 記憶體容量不足的硬體上限，由於無人機影像的高解析度特性，若直接使用影像到卷積神經網路進行訓練，所需要的 GPU 記憶體是十分可觀的。Gao et al. (2020) 在研究中指出影像的大小會影響到單一層卷積神經網路中多個計算步驟的參數大小。而常用的卷積神經網路模型更是由多層的卷積神經網路構成 (Girshick, 2015; Simonyan and Zisserman, 2014; Wang et al., 2023)，因此當輸入影像的解析度越大，其訓練所需要的 GPU 記憶體容量會是多倍的成長。本研究在 Google Colab 上使用記憶體容量 15GB 的 T4 GPU 運行參數量為 3.2M 的 yolov8n 模型 (Ultralytics, 2023)，影像尺寸設定為 640 pixels、Batch Size 設定為 16，則每次 epoch 所需的 GPU 記憶體容量大約為 5-6GB。但若將影像尺寸設定為兩倍的 1280 pixels，訓練時便會發生記憶體容量不足的錯誤 (Out Of Memory, OOM)，無法運行。

誠然硬體的上限會隨著時代的進步而有所進展，過往的上限在未來的時代

doi:10.6342/NTU202404011

可能只是低標，但觀察近幾年市面上較為普及的 GPU 的規格 (表1.1)，可以發現 GPU 的記憶體容量大小成長並不快速，在 RTX 30 系列與 RTX 40 系列的最高規格 3090 與 4090 上，儘管差了一個世代，但兩者的記憶體容量皆是 24G 的規格。而若要在單一顯卡上追尋更高的記憶體容量，NVIDIA RTX 5000 Ada Generation 提供 32G 的記憶體容量、NVIDIA RTX 6000 Ada Generation 提供 48G 的記憶體容量，但這兩者的價格都不親民，並不是普及的配備。GPU 記憶體容量成長並不快速的原因在於 GPU 記憶體容量並不直接影響運算，雖然對 AI 訓練而言，更大的 GPU 記憶體容量能讓使用者在訓練時使用更大的 Batch Size，增加超參數 (Hyperparameter) 的調整彈性，可能有助於獲得更好的結果 (Kandel and Castelli, 2020)，但 Batch Size 也不是越大就一定越好 (Lin, 2022)。總體而言過量的 GPU 記憶體對運算的幫助並不顯著，GPU 記憶體容量大小僅作為是否能運行的門檻。且在一般的電腦視覺任務中高解析度的影像並不常見，因為在普遍影像上目標物的相對大小較大，並不需要靠高解析度才能獲得清晰的特徵。是以 GPU 記憶體容量若不從無人機影像的角度來考量，已經是能應付大部分情況的狀態了。

表 1.1、近年發售的 GPU 型號與其配備的記憶體容量大小以及其在 2023 年的發售價格。

NVIDIA GPU	**記憶體大小**	**價格 (NT)**	**出廠年份**
NVIDIA RTX 3090	24G	46,900	2020 年
NVIDIA RTX 4090	24G	59,900	2022 年
NVIDIA RTX 5000 Ada	32G	180,000	2023 年
NVIDIA RTX 6000 Ada	48G	299,000	2023 年

doi:10.6342/NTU202404011

1.2 研究動機

當面臨 GPU 記憶體容量不足的問題時，最直觀同時也是最常見的做法便是將影像降解析到 GPU 能夠負荷（當然，花錢買更多張顯卡也是解方之一）。Li et al. (2021b) 將原始尺寸為 1920×1080 pixels 的影像縮小成 416×416 pixels、608×608 pixels 和 832×832 pixels 的大小，以訓練在記憶體大小為 11GB 的 NVIDIA GTX 1080ti GPU 上；Liu et al. (2020) 將影像尺寸統一縮小至 608×608 pixels 的大小，以訓練在記憶體大小為 12GB 的 Nvidia GTX Titan XP GPU 上；Zhai et al. (2023) 將原始為 1920×1080 pixels 的影像縮小至 640×640 pixels 的大小，以訓練在記憶體大小為 8GB 的 NVIDIA GeForce RTX 3070 GPU 上。降解析雖然能讓模型進行訓練，但由於無人機影像中目標物的相對尺度較小，一旦將影像進行降解析，這些相對尺度較小的目標會變得模糊且缺失特徵，成為名副其實的小物件（圖1.2），而小物件偵測就是無人機影像進行卷積網路計算時所面臨的主要挑戰。小物件本身特徵就缺稀，又容易在特徵萃取時損失小物件的特徵，同時在評估 (evaluation) 階段的時候小物件對於位置錯誤更敏感，交聯比 (Intersection over Union, IoU) 的分數計算會更嚴苛（圖1.3）。這些都導致了小物件偵測的困難，Cheng et al. (2023) 使用其中一種 State-of-The-Art 的物件偵測模型模型 DyHead(Dai et al., 2021) 對 COCO Dataset 裡的小物件 (0 - 32 × 32 pixels)、中物件 (32 × 32 - 96 × 96 pixels) 及大物件 (96 × 96 - 10^5 × 10^5 pixels) 進行偵測。小物件的全物件平均準確率之均值 (mean Average Precision, mAP) 為 28.3%、中物件與大物件的 mAP 則分別是 50.3% 與 57.5%，足見小物件偵測與一般物件偵測之間依然有一個巨大的表現落差。可想見無人機影像若都面臨小物件偵測，則在表現上依然有可提升的空間。

doi:10.6342/NTU202404011

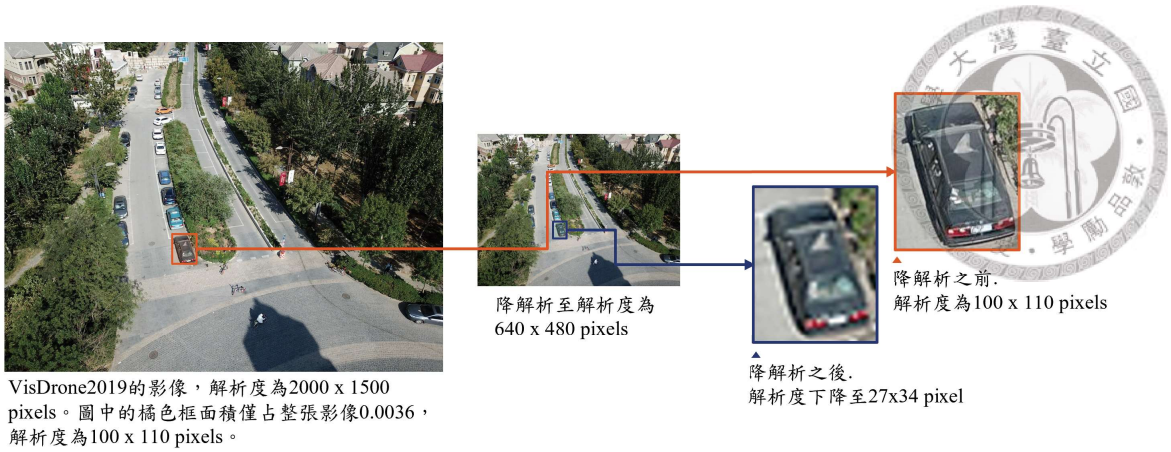


圖 1.2、無人機影像中的目標物在進行降解析之前與之後成像上的差異。降解析後目標物的成像模糊且像素數量較少。

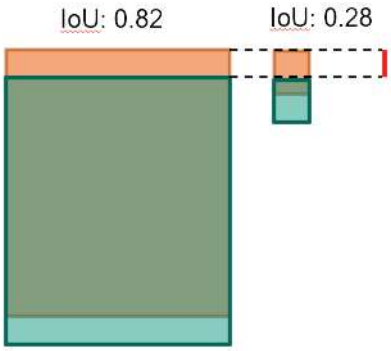


圖 1.3、在物件框有相同的位移下（紅色線），小物件的交聯比 (Intersection over Union, IoU) 分數遠低於大物件的 IoU 分數。圖中橘色矩形為預測物件框。藍綠色矩形為地真資料的物件框。預測物件框與地真資料的物件框大小相同。

綜觀上述，由於無人機影像具備高解析度，使用者需要降低 GPU 的記憶體需求才能使用在卷積神經網路模型上，因此普遍的做法是對影像進行降解析，但經過降解析後，無人機影像中的目標物變成模糊且缺少特徵的小物件，造就了偵測的困難。這是無人機影像所面臨的小物件偵測困境（圖1.4）。針對小物件偵測不論是在無人機影像上或是在普遍影像上都有所研究 (Cheng et al., 2023; Micheal et al., 2021)，普遍的思路是針對卷積神經網路的架構進行調整，使模型對於小物件有更好的偵測能力。但細究在無人機影像中小物件偵測困境發生的原因，其源自於無人機影像在使用的過程中需要進行經過降解析這一步驟，在降解析之前，由於無人機影像是高解析度的，所以即便是相對較小的目標物，目標物的絕對尺寸其實也是足夠的。圖1.5為 VisDrone(Du et al., 2019) 資料集中的一張高解析度無

doi:10.6342/NTU202404011

人機影像，圖中標註的物件框僅佔整張影像的 0.0037，與 COCO Dataset(Lin et al., 2014) 中物件框的平均相對尺寸 0.190±0.203 相比小上許多，但若看絕對尺寸的話，標註的物件框像素尺寸為 100×110 pixels，與 COCO Dataset 中物件框的平均絕對尺寸 99.5±107.5 相比是差不多的。這啟發了本研究的思路，若是能在進行卷積神經網路計算的過程中避免影像降解析，則依照目標物的解析度來看的話目標物並非小物件。因此若有辦法在不降解析度的情況下進行卷積神經網路的計算，那無人機影像所進行的偵測就不再是小物件偵測，小物件偵測的困境便也不復存在。而基於中與大物件偵測的表現與小物件偵測間的落差，一旦繞過小物件偵測的困境，可以預期模型表現會有顯著的成長。

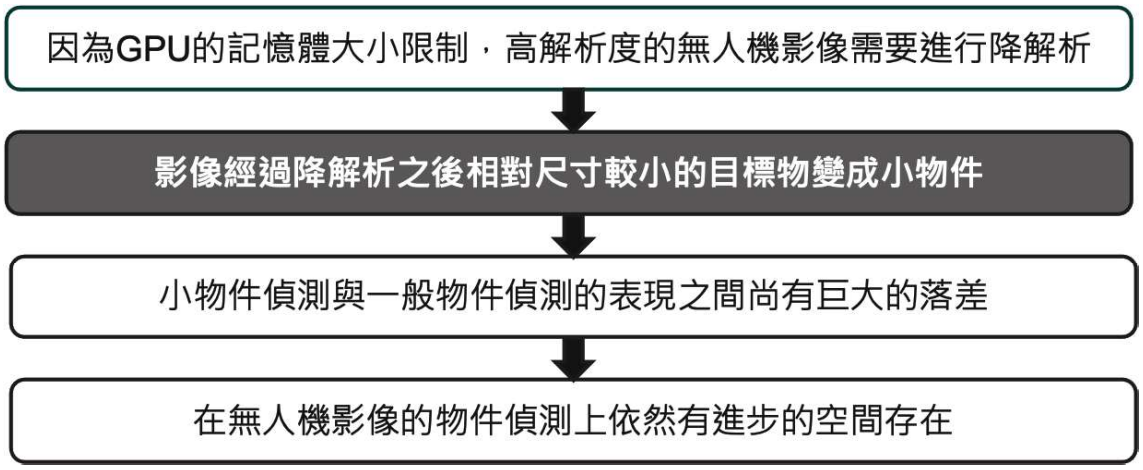


圖 1.4、無人機影像的小物件偵測困境由來脈絡，小物件的困境是因過程中進行降解析而導致。

doi:10.6342/NTU202404011

dataset	absolute size	relative size	aspect ratio
TinyPerson	18.0±17.4	0.012±0.010	0.676±0.416
COCO	99.5±107.5	0.190±0.203	1.214±1.339
Wider face	32.8±52.7	0.036±0.052	0.801±0.168
CityPersons	79.8±67.5	0.055±0.046	0.410±0.008

圖 1.5、無人機影像中物件的相對尺寸與絕對尺寸與多個資料集的平均相對尺寸與平均絕對尺寸的比較。可以看見無人機影像中物件的相對尺寸雖然小於 COCO Dataset，但在絕對尺度上與 COCO Datset 差不多。圖中關於多個資料集平均尺寸的表格出自 Yu et al. (2020)。

1.3 研究目的

為使影像維持高解析度，所以不能依靠降解析來降低所需要的 GPU 記憶體容量，但又要在不提高 GPU 的記憶體容量上限的情況下以卷積神經網路進行訓練，因此勢必要使輸入影像的尺寸變小，一可行的做法是裁切 (crop) 影像。先在高解析度的影像中提出可能有物件存在且是尺寸適當的子區域，將子區域的影像從原始高解析度的影像中裁切而出，作為後續偵測模型的實際輸入。如此既能維持原始的解析度，而子區域的影像作為模型的實際輸入，因為影像尺寸較小也降低了所需的 GPU 記憶體容量，得以在不提高硬體上限的情況下進行模型訓練。如此這般先提出子區域後再進行偵測的方法作法被稱為專注偵測

doi:10.6342/NTU202404011

(Focus-and-Detect)(Cheng et al., 2023)。本研究認為此種方法適合應用在無人機影像上，原因其一是無人機影像自帶高解析度，相對尺度較小的目標物也有一定量的像素含量。其二是無人機影像中的物件在相對尺度上是均小的，先天上不容易出現目標物比子區域還大，此種不適合 Focus-and-Detect 的情況。在過往相關的研究中，ClusDet(Yang et al., 2019) 與 DMNet(Li et al., 2020) 透過先訓練一專門預測子區域的模型，並以子區域預測模型找出影像的子區域後再進行後續的物件偵測任務。ClusDet 與 DMNet 都需要額外的標註資料來訓練預測子區域的模型，並且子區域模型與後續的物件偵測模型並不是端到端 (end-to-end) 的，子區域預測模型與物件偵測模型的最佳化會走向各自的最佳化，而非整體的最佳化。Deng et al. (2020) 提出 global-local self-adaptive network (GLSAN)，該研究方法只使用一個物件偵測模型，但將偵測分為兩次進行來實現 Focus-and-Detect。該方法首先會進行一次低解析度的偵測，並將物件偵測結果透過該研究提出的 self-adaptive region selecting algorithm(SARSA) 轉換成子區域的結果。第二次則依照子區域的結果裁切後進行高解析度的物件偵測，以此來得到更好的結果。雖然此研究提出的方法不需要額外的模型與標註資料，且也是端到端的模型，但此方法在第一次的低解析度的偵測上依然會面對到小物件偵測的問題。

在過去實現 Focus-and-Detect 的研究中，都是使用卷積神經網路去進行子區域的提出，因此不是需要額外的標註資料與模型來輔助，就是無法完全繞過無人機影像所面臨的小物件困境。本研究根據此缺口提出修改版 Selective Search(Modified Selective Search) 演算法，一種以 Selective Search(Uijlings et al., 2013) 演算法為基礎修改的候選區域演算法，修改的部分使 Selective Search 提出的候選區域維持在適當的大小之下，並盡可能地使候選區域內能包含多個目標物。本研究提出的方法不是卷積神經網路，不需要額外的標註資料，也因為並非透過卷積神經網路計算，所以不會面臨到小物件偵測的困境。本研究方法在子區

doi:10.6342/NTU202404011

域選取的部分不佔用 GPU 的記憶體容量，但另一方面來說也無法靠 GPU 進行加速，是以子區域的提出所費時間較長。演算法透過提出適當大小的子區域，達到在不降低影像解析度的同時減少後續物件偵測部分所需要的 GPU 記憶體容量。本研究認為 Focus-and-Detect 的概念更近似於影像擴增技術，因此有別於主流上將子區域提出方式融合在模型架構之中 (Girshick (2015); Ren et al. (2015))，本研究所提出的子區域提出方法獨立於偵測模型之外，是以如同影像擴增技術能使用在偵測模型上，受惠於任何 State-of-The-Art(SoTA) 模型的進步，但與之相對的本研究方法在運算時間上也費時較久。

小結

本研究分析無人機影像中小物件進行卷積神經網路類物件偵測所面臨到的限制與挑戰。限制在於 GPU 的記憶體容量不堪高解析度影像負荷。而為降低 GPU 的記憶體容量負荷，需降低輸入影像的尺寸大小，但透過降解析進行，會面臨到小物件偵測的挑戰。本研究提出一採用 Focus-and-Detect 概念的優化方法（圖1.6)。透過非卷積神經網路的方式提出子區域，並裁切子區域來減少輸入影像的大小，在降低 GPU 記憶體需求的同時不犧牲影像解析度，迴避掉小物件偵測的困境。

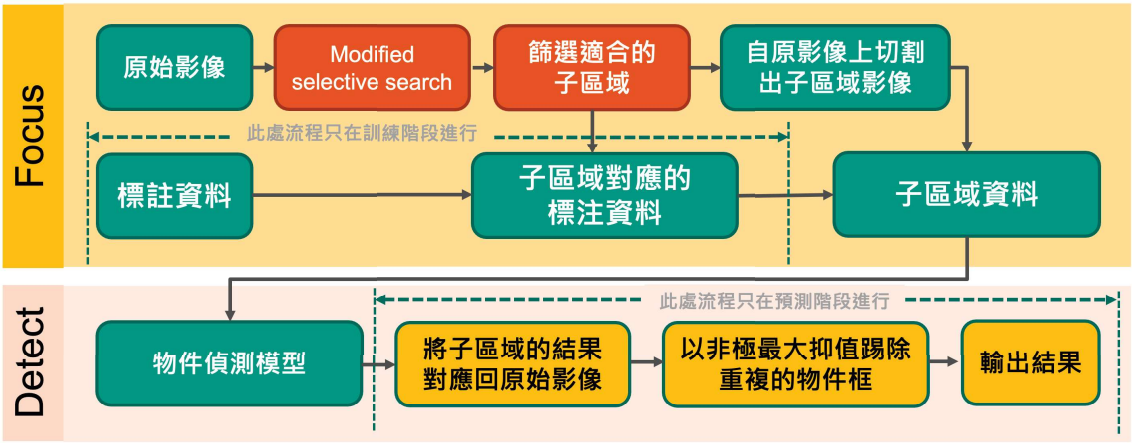


圖 1.6、本研究方法之流程圖。

doi:10.6342/NTU202404011

第二章　研究方法設計

本章節詳細的介紹所提出的 Focus-and-Detect 優化辦法，辦法旨在降低 GPU 記憶體的需求與提高針對小物件的偵測精度。研究的重點在 Focus-and-Detect 中的前部分，如何提出子區域。後續的偵測部分則可以套用任一模型進行，在本研究中偵測部分以 Yolov8(Ultralytics, 2023) 物件偵測模型進行實驗。

在子區域提出的部分，最直觀且顯而易見的作法是以圖磚的方式將影像等比例的裁切到適宜的大小，然而圖磚的分割方法與影像內容及資料都無關，無法避免所提出來的子區域邊界剛好畫在目標物上導致目標物的完整性遭破壞的情況。本研究方法所提出的 Modified Selective Search 修改自 Selective Search 這一候選區域演算法，使演算法所提出的候選區域能盡量符合理想的子區域（圖2.1），理想的子區域有二個特性：

1. 影像大小符合卷積神經網路所適合的尺寸大小，避免訓練所需的 GPU 記憶體大小不易負擔。
2. 在避免破壞目標物的完整性時，盡可能的使子區域內包含多個目標物，以有效率的方式涵蓋影像內所有的目標物，降低子區域的數量以減少後續偵測所需的時間。

doi:10.6342/NTU202404011

圖 2.1、圖磚法與理想的子區域提出方法間的差異。左圖圖磚法提出的子區域的邊界劃分在目標物上，也需較多的子區域來涵蓋全部的目標物。右圖為較為理想的子區域提出方法，一個子區域包含多個完整的目標物。

由於子區域是後續模型的實際輸入，因此在測試階段時也會是以子區域進行偵測，而由於子區域的涵蓋範圍會有所重複，同一目標物可能被多個子區域涵蓋，因此在將所有子區域的偵測結果合併回母影像的結果時，本研究以慣用的 Non-Maximum Suppression(NMS) 處理多個子區域對同一目標物重複框選的問題 (Girshick et al., 2014)。

2.1 子區域提出的方法

2.1.1 Selective Search 的介紹

Selective Serach 是Uijlings et al. (2013) 於 2013 年所提出的候選區域演算法，結合了影像分割 (image segmentation) 與竭盡式搜尋 (exhausting search) 的優點，使最後提出的候選區域是資料驅動 (Data Driven)、類別獨立 (Class-independent) 且位置精確的候選區域。Selective Search 作為區域提出演算法，早在 2014 年就已經被使用於深度學習類的物件偵測上了，Girshick et al. (2014) 提出經典的 R-CNN 模型，先以 Selective Search 演算法對影像進行區域候選的提出，約莫提出兩千餘個物件可能存在的候選區域，後續再以卷積神經網路對候選區域進行影像辨識，預

doi:10.6342/NTU202404011

測該候選區域的類別，如此便能進行影像中多種物件類別的定位與類別判斷，是為物件辨識。Selective Search 的步驟可以劃分為三步驟：

1. 利用Felzenszwalb and Huttenlocher (2004) 進行影像分割並根據分區結果提出最初的候選區域。
2. 計算候選區域間的相鄰情況，而相鄰的候選區域計算彼此之間的相似度。
3. 從最相似的兩區域開始，合併所有相鄰的候選區域。

詳細的演算法如圖2.2所示，其中關於相似度 (similarity) 的計算Uijlings et al. (2013) 使用了四種相似度值。分別是顏色相似度、材質相似度、尺寸相似度與合併後形狀變化的相似度。每個合併後的區域同時也會反過來計算與該新區域相鄰的候選區域，以及兩個區域之間的相似度。Selective Search 永遠挑遠最相似的兩個區域進行合併，透過貪婪演算法 (greedy algorithm) 不斷合併直到沒有區域可以進行合併為止。

原本的 Selective Search 的目標是針對物件的候選區域，與本研究所希望子區域有些許的不同。首先在演算法的設計上Uijlings et al. (2013) 希望 Selective Search 能應付多樣尺寸的目標物，因此對於候選區域的大小並無限制，而本研究所希望的子區域則是要在適當的大小之下。另外 Selective Search 提出的是為物件辨識 (Object Recognition) 服務的候選區域，對於Uijlings et al. (2013) 來說，理想的候選區域是以單一目標物為範圍，與本研究所希望的包含多個目標物的子區域有所不同，因此本研究以 Selective Search 為基礎進行調整，使其符合本研究所希望的子區域。

doi:10.6342/NTU202404011

Algorithm 1 Original Selective Search (Uijlings et al., 2013)

Input: (colour) image

Output: Set of object location hypotheses L

Obtain initial regions $R = r_i, ...r_n$ using Felzenszwalb and Huttenlocher (2004)
Initialise similarity set $S = \{\}$;
foreach Neighbouring region pair (r_i, r_j) **do**
  Calculate similarity $s(r_i, r_j)$;
  $S = S \cup s(r_i, r_j)$;
end for

while $S \neq \{\}$ **do**
  Get highest similarity $s(r_i, r_j) = max(S)$;
  Merge corresponding regions $r_t, r_i \cup r_j$;
  Remove similarities regarding $r_i : S = S \setminus s(r_i, r_*)$;
  Remove similarities regarding $r_j : S = S \setminus s(r_*, r_j)$;
  Calculate similarity set S_t between r_t and its neighbours;
  $S = S \cup S_t$;
  $R = R \cup r_t$;
end while

Extract object location boxers L from all regions in R;

圖 2.2、原始 Selective Search 的演算法架構。

2.1.2 針對研究目標所調整的修改版 Selective Search

為使演算法所提出的候選區域符合理想子區域的兩個特性：大小適宜與包含多個目標物。本研究提出在 Selective Search 為基礎的演算法上新增幾個條件的修改版 Selective Search（Modified Selective Search），演算法架構如圖2.5。為使演算法能提出多個目標物包含在內的子區域，當 Modified Selective Search 在計算候選區域的相鄰情況時，會擴增原本的候選區域邊界作為假想的邊界，使其能與更多的候選區域相鄰，圖2.3展示了在不同的擴增值底下，子區域提出的最終結果，擴增邊界能增加在目標物分佈較為分散的情況下依然能提出理想的子區域的機率，圖2.3a在邊界擴增寬度為 0 的情況下，相鄰的候選區域無法產出一大小足夠，且包含圖中較為分散的目標物的子區域。圖2.3c在邊界擴增寬度為 20 的情況下，因為更多的候選區域產生相臨，更容易在合併後產生大小足夠且包含分散目標物的

doi:10.6342/NTU202404011

子區域。假想的邊界僅作用在計算相鄰候選區域上，後續的相似度計算依然是以原始的候選區域進行計算。

(a) 邊界擴增寬度為 0px　(b) 邊界擴增寬度為 10px　(c) 邊界擴增寬度為 20px

圖 2.3、在不同的邊界擴增寬度底下，同一張影像子區域的最終提出結果。綠色區域為所提出的子區域。圖 (a) 中邊界擴增寬度為 0 px，因所有子區域都未合併成夠大的區域，所以沒有子區域被提出。圖 (c) 則是在邊界擴增寬度為 20 px 的情況，有更多的候選區域產生相鄰並合併成大區域。

除邊界擴增之外，為使提出的候選區域的大小能符合所希望的子區域大小，當演算法將最相似的兩個相鄰區域 r_i 與 r_j 合併成 r_t 後，判斷 r_t 的尺寸大小，若大小超過所希望的子區域大小則將 r_t 自相鄰集合 S 中排除，不再將 r_t 與其相鄰候選區域加入相鄰集合 S 中。透過上述針對大小的限制，能將候選區域的大小朝希望尺寸靠攏（圖2.4），但不會完全限制大小在希望尺寸之下，因為若原先的候選區域已接近尺寸大小，則合併後便有機會超出希望尺寸大小，但理論不會超出兩倍的希望尺寸大小。

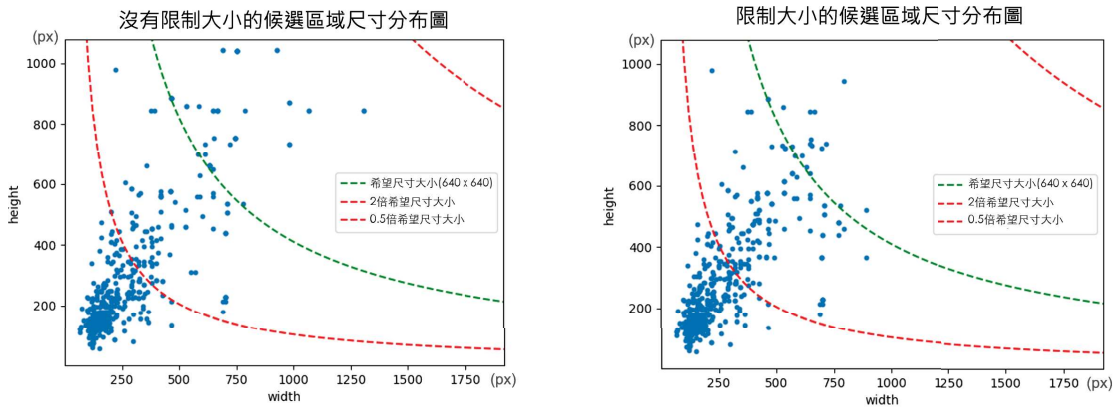


圖 2.4、影像經過子區域提出方法後得到的候選區域的尺寸分布圖及後選區域經過尺寸大小限制後的尺寸變化比較。左圖為未增加限制前的候選區域的尺寸分布；右圖為增加限制後的候選區域的尺寸分布。相較於左圖，經過限制後的尺寸更接近希望尺寸。

doi:10.6342/NTU202404011

Algorithm 2 Modified Selective Search

Input: (colour) image

Output: Set of sub-regions hypotheses R_s

```
Obtain initial regions R = r_i, ...r_n using Felzenszwalb and Huttenlocher (2004)
Initialise similarity set S = {};

for r_i in R do
    Increase r_i border by b as abstract bound;
end for
Get all neighbouring region pair (r_i, r_j) from abstract bound

for each Neighbouring region pair (r_i, r_j) do
    Calculate similarity s(r_i, r_j);
    S = S ∪ s(r_i, r_j);
end for

while S ≠ {} do
    Get highest similarity s(r_i, r_j) = max(S);
    Merge corresponding regions r_t, r_i ∪ r_j;
    Remove similarities regarding r_i : S = S \ s(r_i, r_*);
    Remove similarities regarding r_j : S = S \ s(r_*, r_j);

    if Size of r_t > preferred region size then
        continue
    end if

    Calculate similarity set S_t between r_t and its neighbours;
    S = S ∪ S_t;
    R = R ∪ r_t;
end while

Extract sub-regions boxers R_s from all regions in R;
```

圖 2.5、Modified Selective Search 的演算法架構。修改的部分在第 4 至第 7 行的 for 迴圈及第 20 行至第 23 行的 if 判斷。

2.2 子區域的篩選方法

演算法所提出的區域為候選區域，經篩選後成為最後的子區域。篩選候選區域為子區域的目的是為了減少最後所提出的子區域數量，減少後續偵測模型的時間花費。由於提出越多的子區域代表後續的偵測模型有越多的輸入，因此子區域

doi:10.6342/NTU202404011

的數量與偵測一張影像所花費的時間成正比關係。雖然在訓練階段能透過地真資料作為額外資訊來篩選提出的子區域，但在測試 (testing) 階段中是不會有地真資料來判斷子區域的理想與否，子區域提出方法所提出的所有子區域都會作為後續偵測模型的輸入，因此在子區域提出的方法上以演算法本身來盡可能的排除掉無用的子區域對於後續偵測來說是有所幫助的，尤其是對於測試階段。

2.2.1 篩選掉尺寸過小的候選區域

經過 Modified Selective Search 演算法所產生的候選區域數量眾多，同時包含許多小尺寸的候選區域，依照子區域的概念這些小尺寸的候選區域並非理想的子區域，因此在候選區域的提出後以希望尺寸大小的一半作為閾值，僅保留大於尺寸大於閾值的候選區域（方程式2.1），如此可以大幅減少候選區域的數量，留下適合作為子區域的候選區域（圖2.6）。

$$R_s = \{r_s \; for \; S_{r_s} > \frac{S_{ps}}{2}\} \tag{2.1}$$

其中:

R_s 是子區域的集合。

r_s 是子區域。

S_{r_s} 是子區域的尺寸。

S_{ps} 是子區域的希望尺寸。

doi:10.6342/NTU202404011

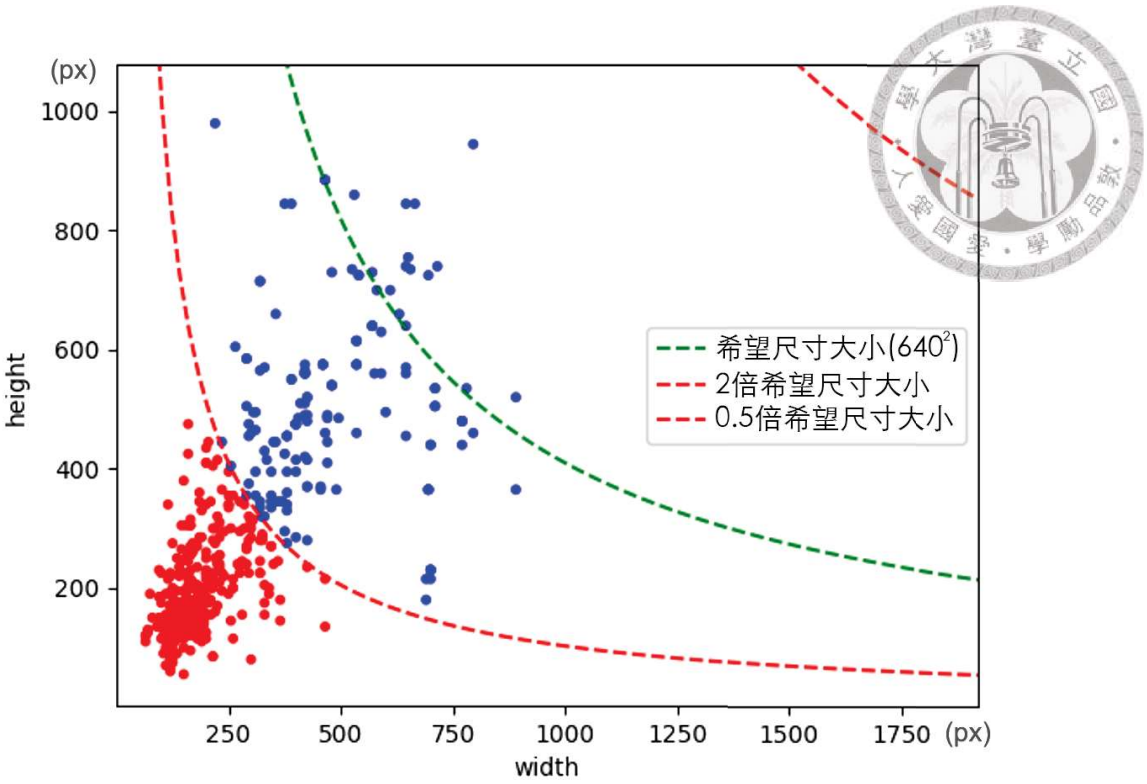


圖 2.6、候選區域經過尺寸閾值篩選後的結果。紅色點為低於閾值的候選區域。

2.2.2 篩選掉第一階段產生就已超過希望尺寸的候選區域

在 Modified Selective Search 演算法所產生的候選區域中，排除掉在最開始以Felzenszwalb and Huttenlocher (2004) 影像分割產生的區域時，尺寸就已經超過希望尺寸的大小的候選區域。這些候選區域的來源有大概率是大範圍且顏色相近的背景區域（圖2.7），譬如天空、河川、樹叢或柏油道路，而這些背景區域有較高的機率不包含任何的目標物，本研究更專注在那些由物件彼此之間相鄰合併而來的候選區域，由此而來的候選區域有更高的機會符合理想子區域的屬性。

doi:10.6342/NTU202404011

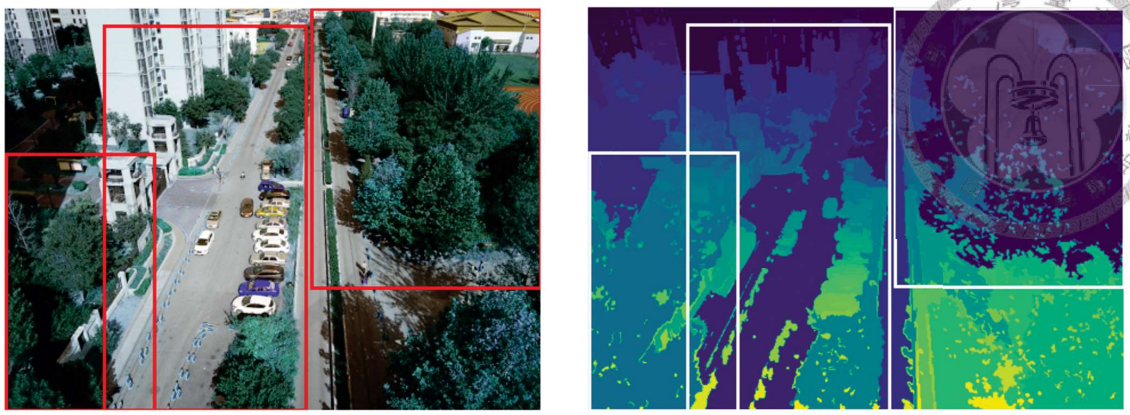


圖 2.7、在Felzenszwalb and Huttenlocher (2004) 影像分割產生的區域中就已經大於希望尺寸的區域範例。左圖為候選區域在原始影像上的位置與內容物，是長條柏油路與行道樹的樹叢區域。右圖為在影像分割結果上的位置，可見構成大尺寸候選區域的群類（圖中相同顏色為同一群）。

2.2.3 篩選掉高重複的候選區域

在候選區域的提出中，候選區域間彼此有高重疊率是常見的結果，因為 Modified Selective Search 在計算合併時使用的相鄰集合 S 只包含第一次候選區域產生時的所有相鄰對 (r_i, r_j)。在 r_i 與 r_j 合併成 r_t 後，也僅與相鄰集合 S 內的元素進行新的相鄰判斷（圖2.5），所以在合併完後有高重疊的相鄰區域是可被演算法接受的。為了降低最後子區域的數量，這些高重複的候選區域會在挑選候選區域為子區域時進行排除，若子區域集合 R_s 中舊有子區域 r_{os} 與新的候選區域 r_{ns} 間的重疊率大於己身區域的閾值則 r_{ns} 不放入子區域集合 R_s 之中（方程式2.2），在本研究中重疊率的閾值設定為 0.7。

doi:10.6342/NTU202404011

$$\begin{cases} r_{ns} \in R_s & \text{if } overlap(r_{ns}, r_{os}) \leq 0.7 \text{ for } r_{os} \in R_s, \\ r_{ns} \notin R_s & \text{otherwise} \end{cases} \tag{2.2}$$

其中:

r_{ns} 是新的候選區域。

R_s 是子區域的集合。

r_{os} 是子區域集合中已存在的子區域。

關於重疊率的計算如方程式2.3。

$$overlap(r_a, r_b) = max(\frac{r_a \cap r_b}{r_a}, \frac{r_a \cap r_b}{r_b}) \tag{2.3}$$

2.3 子區域提出方法的評估指標

2.3.1 平均地真資料的涵蓋率

偵測模型的實際輸入是子區域，換言之如果是子區域未涵蓋到的目標物，則不論偵測模型的能力再好都無法被偵測到。是以地真資料的涵蓋率是一個子區域提出方法好壞的重要依據之一，同時也會是 Focus-and-Detect 偵測方法的瓶頸(bottleneck)。本研究中地真資料的涵蓋率的計算方式如圖2.8。將每一筆標註資料與提出的所有子區域計算該標註資料在每一子區域內的面積比例，取最高的涵蓋面積比例作為該筆標註資料的地真資料的涵蓋率。最後加總所有標註資料的地真資料的涵蓋率並除以標註資料的筆數，得到的值為該影像在子區域提出方法中的地真資料的涵蓋率。而平均地真資料的涵蓋率則是計算整份資料集經過子區域提出方法後，所有影像平均的地真資料的涵蓋率。作為模型偵測的瓶頸，平均地真資料的涵蓋率不該低於後續偵測模型的 mAP。

doi:10.6342/NTU202404011

```
Algorithm Ground Truth Coverage
Input: annotations & sub-regions
Output: ground truth coverage
  Total-ground-truth-coverage = [];
  foreach annotation do
    foreach sub-region do
      Calculate the overlap rate between annotation and sub-region;
    end for
    Coverage = the highest overlap rate for this annotation;
    Add coverage to Total-ground-truth-coverage array;
  end for
```

$$\text{Ground truth coverage} = \frac{\sum(\text{Total-ground-truth-coverage})}{\text{len}(\text{Total-ground-truth-coverage})}$$

圖 2.8、Ground Truth Coverage 的計算方式。

2.3.2 量化子區域的理想程度

本研究提出 Score of Sub-region(SOS) 來量化子區域的理想程度，SOS 計算子區域內包含了多少個完整的目標物，SOS 分數越高則區域內包含越多個目標物，反之則子區內的目標物越少，越接近無效或無用的子區域（圖2.9）。SOS 分數可以將演算法的結果再作進一步的篩選來減少子區域數量，僅保留 SOS 分數較高的子區域來進行訓練可以顯著的降低子區域的數量，減少訓練所需時間，而這樣的篩選方法是否有助於整體偵測的結果將在下章實驗結果與討論中進行討論。值得一提的是 SOS 分數的計算使用了標註資料作為額外資訊，因此這樣的篩選方式僅能應用在訓練階段，於測試階段是無法計算 SOS 分數的。

doi:10.6342/NTU202404011

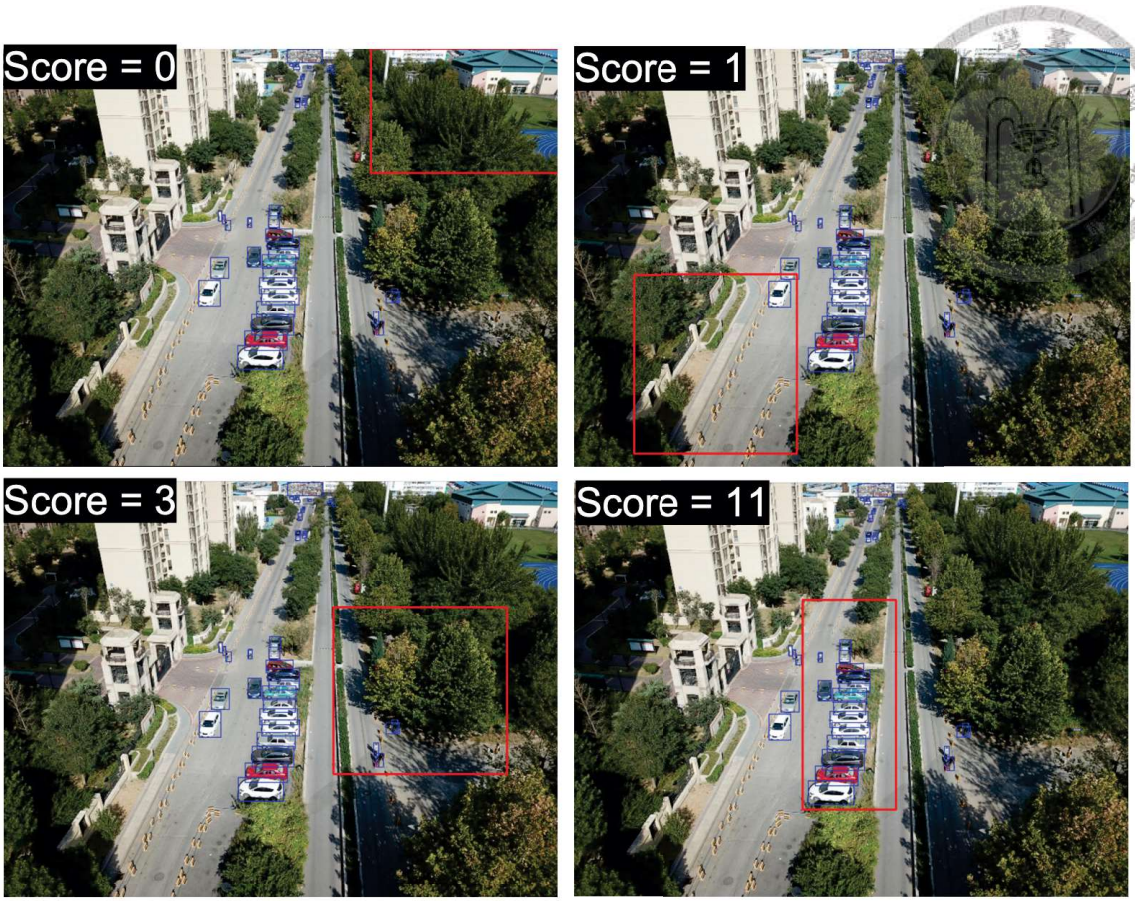


圖 2.9、子區域的 SOS 分數範例。區域內包含越多目標物則分數越高。反之分數為 0 的代表區域內無任何目標物，屬於無效的子區域。

小結

在本研究所提出的 Focus-and-Detect 的方法中，在子區域提出的部分以 Selective Search 為基礎，透過候選區域大小的限制與擴增候選區域的邊界這兩項修改，使演算法提出的候選區域符合理想子區域的特性－大小適宜與包含多個物件。而為了降低子區域的數量，演算法篩掉候選區域中尺寸過小、第一階段產生的過大區域與高度重複的區域，保留較為符合理想子區域特性的候選區域為最終的子區域。在子區域提出方法上，本研究提出以地真資料涵蓋率量化子區域提出方法的好壞，並以 SOS 分數來進行子區域的篩選。後續章節將以本小節所提之研究方法，對 VisDrone 無人機影像資料集 (Du et al., 2019) 進行物件偵測的評估。

doi:10.6342/NTU202404011

第三章　實驗結果與討論

本章節依序研究方法，使用 VisDrone2019(Du et al., 2019) 資料集，將影像依上章節所提之方式進行子區域的提出，以 yolov8(Ultralytics, 2023) 針對子區域影像進行訓練。最後在測試資料集上 (testing dataset) 的影像同樣依照研究方法進行子區域的提出，並以訓練的模型進行物件偵測，最後合併子區域的結果至母影像之上進行評估 (evaluation)，並以測試資料集上的評估結果作為評量依據。本研究針對底下四種不同的情況進行實驗與評估：

- 本研究所提出的 Focus-and-Detect 方法。
- 本研究所提出的 Focus-and-Detect 方法並以 SOS 分數做子區域的篩選。
- 影像降解析後進行物件偵測。
- 影像經圖磚法切分後進行物件偵測。

3.1　VisDrone 資料集中的高解析度影像

本研究使用 VisDrone(Du et al., 2019) 資料集中高解析度的影像為基礎資料集進行後續的實驗。VisDrone 資料集中囊括了多樣的無人機影像，資料集中影像的尺寸在 960×540 到 2000×1500 之間。資料集提供的類別共有十種，如表3.1。

doi:10.6342/NTU202404011

表 3.1、VisDrone 所提供的十種類別。

類別編號	類別名稱
1	行人 (pedestrian)
2	人 (people)
3	腳踏車 (bicycle)
4	車 (car)
5	貨車 (van)
6	卡車 (truck)
7	三輪車 (tricycle)
8	帶蓬三輪車 (awning-tricycle)
9	公車 (bus)
10	摩托車 (motor)

本研究所提出的方法是在高解析度的無人機影像底下的優化辦法，因此將 VisDrone 的資料集依照影像尺寸進行挑選，將原始資料集中 HD 以上的影像 (1920×1080 與 2000×1500) 挑出作為本研究使用之資料集，VisDrone 高解析度影像資料集的詳細情況如表3.2，後續會以此資料集為基礎，進行降解析處理、圖磚法處理或是本研究所提出的研究方法來產生四種實驗的資料集。

表 3.2、高解析度影像 (1920×1080 與 2000×1500) 的 VisDrone 資料集。

資料集	影像張數
訓練資料集 (training dataset)	1,648 張
驗證資料集 (validation dataset)	19 張
測試資料集 (testing dataset)	187 張

doi:10.6342/NTU202404011

3.2 訓練階段的專注偵測

3.2.1 子區域對應的標註資料

在專注偵測（Focus-and-Detect）的流程如研究流程圖1.6所示，在訓練階段進行子區域提出時還需要將原先的標註資料對應到提出的子區域上。在判斷標註物件框是否該被歸納於子區域內時，本研究以物件框是否有過半的面積被子區域涵蓋為基準，以上則納入，以下則不納入，如方程式3.1。

$$\begin{cases} bbox \in GT_{r_s} & \text{if } \dfrac{Area(bbox \cap r_s)}{Area(bbox)} \geq 0.5 \\ bbox \notin GT_{r_s} & \text{otherwise} \end{cases} \tag{3.1}$$

其中:

r_s 是子區域。

$bbox$ 是原始影像的標註物件框。

GT_{r_s} 是子區域 r_s 的地真資料集合。

3.2.2 修改版 Selective Search 的參數設定

修改版 Selective Search(Modified Selective Search) 的過程中會經過Felzenszwalb and Huttenlocher (2004) 的影像分割流程，因此輸入影像的解析度以及演算法中使用的參數 k 會影響影像分割的結果，進而影響 Modified Selective Search 的結果。再者是 Modified Selective Search 中所設定的子區域希望尺寸與子區域邊界擴增的寬度也會影響演算法中候選區域合併的結果，進而影響最後的結果。將上述影響 Modified Selective Search 的最終子區域的參數歸納如表3.3。其中輸入影像的解析

doi:10.6342/NTU202404011

度影響影像的細緻程度，細緻程度影響影像分割的精細度，越細緻的影像可以分得越細，但同時也比較花時間。影像分割參數 k 同樣影響分割的細緻程度，越小的 k 分割的越細緻，但所費時間也越久。子區域希望尺寸的設定大小影響最後子區域的數量，但對時間影響不大，本研究建議以後續模型與具備的 GPU 性能來考量希望尺寸的大小設定。邊界擴增的寬度會讓更多的小區域合併成大區域，進而使區域大小超過閾值的子區域數量變多，因此對於演算法最後的結果來說，邊界擴增的寬度設定的越寬，最終的子區域的數量越多。在本章節的實驗中針對這些參數的設定如表3.4，設定的依據如下述，子區域提出的範例如圖3.1：

- 子區域希望尺寸設定為 yolov8n 使用 COCO Dataset 的影像進行預訓練 (pre-train) 時使用的尺寸大小 640×640 pixels。
- 在邊界擴增的寬度大小設定上，以不同的邊界擴增寬度對訓練資料集進行測試，統整平均地真資料涵蓋度、平均每張所費時間與子區域數量（表3.5）。可以看見有無邊界擴增對於地真資料涵蓋度有大幅度的提升。但隨著寬度增加對於地真資料涵蓋度的增長有邊際效益，且會進一步的增加子區域的數量，並花費更久的時間，因此本研究的實驗以 10 像素作為設定。
- 輸入影像的解析度與影像分割參數 k 則使用預設值 (AlpacaTechJP, 2017)。

doi:10.6342/NTU202404011

表 3.3、Modified Selective Search 中的參數與其意義。

Modified Selective Search 的參數	**其代表的意義**
輸入影像的解析度 (相比於原解析度)	進行影像分割時的影像細緻程度。高解析做得較慢，但分割的越細緻。解析度對演算法所費時間影響顯著。
影像分割參數 k	k 是Felzenszwalb and Huttenlocher (2004) 演算法中的觀察尺度 (scale of observation)。k 的數值越大分割的越粗糙，對演算法所費時間影響顯著。
子區域希望尺寸的大小	在 Modified Selective Search 演算法架構（圖2.5）中子區域的希望大小閾值 (preferred size)。小尺寸會有較多的子區域被提出，對演算法所費時間影響不大。
邊界擴增的寬度	在 Modified Selective Search 演算法架構（圖2.5）中擴增的假想邊界寬度。擴增的越寬最後會有越多的子區域數量，稍微增加演算法所費時間。

doi:10.6342/NTU202404011

表 3.4、實驗使用的 Modified Selective Search 中的參數。

Modified Selective Search 的參數	**實驗設定**
輸入影像的解析度 (相比於原解析度)	0.4
影像分割參數 k	500
子區域希望尺寸的大小	640 × 640 pixels
邊界擴增的寬度	10 pixels

表 3.5、在其餘設定維持表3.4下，僅對邊界擴增的寬度調整，其對訓練資料進行子區域提出的平均地真資料涵蓋度、所費時間與子區域數量的統計。

邊界擴增的寬度設定	**平均地真資料涵蓋度**	**平均每張所費時間**	**子區域數量**
0 pixels	0.820	2.52秒/張	21,283
10 pixels	0.980	2.88秒/張	42,484
20 pixels	0.997	3.50秒/張	65,938
30 pixels	0.999	4.26秒/張	89,264
40 pixels	0.999	5.16秒/張	111,427
50 pixels	0.999	6.05秒/張	131,215

doi:10.6342/NTU202404011

圖 3.1、本研究所提出 Modified Selective Search 演算法的子區域提出結果範例。

3.2.3 修改版 Selective Search 的子區域提出結果

依照上述所說之設定，在訓練階段分別對訓練資料集、驗證資料集與測試資料集進行子區域的提出，在訓練資料集的 1648 張影像上，Modified Selective Search 共費時約 80 分鐘，在平均地真資料的涵蓋率上達到 0.981，產生子區域影像共 42484 張。驗證資料集 19 張上，演算法共費時不到 1 分鐘，在平均地真資料的涵蓋率上達到 0.999，產生子區域影像共 866 張。測試資料集 187 張上，演算法共費時約 8 分鐘，產生子區域影像 5158 張。

訓練資料集與驗證資料集的子區域提出結果再以 SOS 分數進行篩選，本研究以每張影像的平均子區域分數 (Average Score of Sub-regions, ASOS) 作為閾值，ASOS 的算法如方程式3.2。

doi:10.6342/NTU202404011

$$ASOS(R_s) = \frac{\sum^{R_s} SOS(r_s)}{\text{len}(R_s)} \tag{3.2}$$

其中:

R_s 是影像經演算法所提出的子區域的集合。

$ASOS(R_s)$ 是 R_s 的平均子區域分數。

r_s 是子區域。

$SOS(r_s)$ 是 r_s 的子區域分數。

$len(R_s)$ 是 R_s 內的元素總數。

而每張影像經 Modified Selective Search 提出子區域後，僅保留其中 SOS 分數高於該影像 ASOS 分數的子區域，具體的算法如方程式3.3。則經篩選後的訓練資料集共 13196 張，驗證資料集 145 張。

$$\begin{cases} r_s \in FR_s & \text{if } SOS(r_s) \geq ASOS(R_s) \\ r_s \notin FR_s & \text{otherwise} \end{cases} \tag{3.3}$$

其中:

r_s 是子區域。

R_s 是影像經演算法所提出的子區域的集合。

FR_s 是最後保留的子區域集合。

$SOS(r_s))$ 是子區域 r_s 的子區域分數。

$ASOS(R_s)$ 是子區域集合 R_s 的平均子區域分數。

有鑒於驗證資料集與訓練資料集之間的數量差異，在經過 Modified Selective Search 進行子區域提出後，不論有無經過 SOS 篩選，都將訓練資料集的子區域影

doi:10.6342/NTU202404011

像以 8 : 2 的比例隨機切分，並將其中兩成歸給驗證資料集來補足數量差距，經數量調整後的實驗組資料集如表3.6所示。未經 SOS 篩選的資料集中，訓練資料集共有 33988 張，驗證資料集 9362 張，測試資料集共 5158 張。經 SOS 篩選後的資料集中，訓練資料集共 10057 張，驗證資料集共 2784 張，測試資料集則無法經篩選，因此跟未經 SOS 篩選的資料集相同，5158 張。

表 3.6、高解析度影像 (1920×1080 與 2000×1500) 的 VisDrone 資料集經子區域提出後的子區域資料集的影像數量。分別是有經過 SOS 分數篩選與未經過 SOS 分數篩選的兩組資料集。

資料集	未經 SOS 篩選的資料集	經 SOS 篩選後的資料集
訓練資料集	33,988 張	12,410 張
驗證資料集	9,362 張	3,254 張
測試資料集	5,158 張	5,158 張

3.3 評估階段的專注偵測

專注偵測 (Focus-and-Detect) 如研究流程圖1.6所示，在評估階段完成物件偵測後，子區域的偵測結果會整合回母影像之上，由於同一物件可能被多個子區域同時偵測到，因此在合併時會採取 NMS 來處理重複的物件框。NMS 的程序如圖3.2所示，從信心度 (confidence score) 最高的偵測結果開始保留，若有其餘偵測結果與被保留的偵測結果間的 IoU 大於設定閾值，則刪除該偵測結果的物件框。在本研究中閾值設定為 0.5。

doi:10.6342/NTU202404011

Algorithm Non-Maximum Suppression(Brodmann17, 2018)

Input: $B = \{b_1, ..., b_n\}$, $S = \{s_1, ..., s_N\}$, N_t

B is the list of initial detection boxes.
S contains corresponding detection scores.
N_t is the NMS threshold.

```
D = {}
while B ≠ {} do
    m = argmax S
    M = b_m;
    D = D ∪ M;
    B = B \ M;
    for b_i in B do
        if iou(M, b_i) ≥ N_t then
            B = B \ b_i;
            S = S \ s_i;
        end if
    end for
end while

return D, S;
```

圖 3.2、Non-Maximum Supression 演算法的架構。

3.4 物件偵測模型的訓練與測試資料的評估結果

3.4.1 四種實驗使用的資料集

四種實驗情況分別是經過本研究所提出的 Focus-and-Deect 方法處理、經過本研究所提出的 Focus-and-Detect 方法處理後再以 SOS 分數篩選、圖磚法處理與影像降解析處理。其中前兩項的資料集的產生與結果如上述所述（表3.6）。而圖磚法處理的資料集與影像降解析處理的資料集產生方式如下述。首先在前處理所費時間上，兩者所費時間都可忽略不計，影像降解析在模型訓練時可一併進行，而圖磚法的前處理處理訓練資料集 1648 張只需 36 秒。而在資料集的影像張數上，降解析的資料集與原本的資料集相同，惟解析度統一降低至 640 為最大邊長。影像降解析的訓練資料集 1648 張、驗證資料集 19 張與測試資料集 187 張，同樣有

doi:10.6342/NTU202404011

鑒於訓練資料集與驗證資料集的數量差，將訓練資料集以 8 : 2 隨機切分，並將其中的兩成歸給驗證資料集。調整後的訓練資料集共 1319 張、驗證資料集 348 張與測試資料集 187 張。而圖磚法處理的資料集在處理後，同樣將訓練資料集的兩成歸給驗證資料集後，最後圖磚法處理的訓練資料集共 11616 張、驗證資料集 3018 張與測試資料集 1122 張。

四種情況的資料集整理如表3.7，測試資料如有經過切分處理，在偵測完成後會依照3.3章節所述之方法，合併回原始的 187 張測試影像上。

表 3.7、所有實驗各自所使用的資料集詳細情況。

實驗情況	訓練資料集	驗證資料集	測試資料集
本研究所提出的 Focus-and-Detect	33,988 張	9,362 張	5,158 張
本研究所提出的 Focus-and-Detect + SOS 篩選	12,410 張	3,254 張	5,158 張
圖磚法	11,616 張	3,018 張	1,122 張
影像降解析	1,319 張	348 張	187 張

doi:10.6342/NTU202404011

所有實驗運行在 NVIDIA® RTX™ 4000 Ada 上，記憶體為 20 GB GDDR6。相關的訓練設定如表3.8，模型使用參數總量為 3.2M 並預訓練 (pre-train) 在 COCO Dataset 的 Yolov8n 模型 (Ultralytics, 2023)、影像尺寸設定為 640 × 640 pixels、epochs 設定為 50 次和 batch size 設定為 16，其餘設定使用Ultralytics (2023) 所設定的預設數值。

表 3.8、本研究實驗步驟所使用的訓練參數。

訓練參數	參數設定
模型架構	yolov8n(3.2M)
影像尺寸	640 × 640
epochs	50
batch size	16

3.4.2 四種實驗訓練所需時間與 GPU 記憶體使用量

四種實驗在訓練上所花費的時間與 GPU 記憶體使用量如表所示3.9。時間花費上 Focus-and-Detect 的方法費時最久，基本上與訓練用的影像張數成正比關係，子區域的數量越多時間越久。影像降解析則是最快的。而在 GPU 記憶體使用量上，輸入影像尺寸相同且 Batch Size 都一樣的情況下，四者的情況都差不多，約 5-6GB。不過值得提起的是，在本研究的 VisDrone 高解析度資料集上，若想直接原始解析度的 1920 × 1080 影像尺寸且 Batch Size 為 16 去進行訓練，則會發生記憶體空間不足的問題，無法運行。

doi:10.6342/NTU202404011

表 3.9、四種實驗情境下訓練所需的時間與所占用的 GPU 記憶體大小。

實驗情境	訓練費時	前處理費時	佔用記憶體
本研究所提出的 Focus-and-Detect	3.5 小時	1.4 小時	5-6GB
本研究所提出的 Focus-and-Detect ＋SOS 篩選	1.4 小時	2.6 小時	5-6GB
圖磚法	1.1 小時	0 小時	5-6GB
影像降解析	0.2 小時	0 小時	5-6GB

3.4.3 四種實驗在測試資料的評估結果

統計四種實驗在 187 張測試資料上的結果，以物件偵測常用的 mAP 為指標，結果如圖3.3。其中 mAP0.5、mAP0.75 分別代表在物件偵測的結果中，如果預測框與地真資料的物件框的 IoU 分數在 0.5 內與在 0.75 內，則該物件框視為真陽(True positive)。而 mAP0.5:0.95 則是以 0.05 為步距大小，從 mAP0.5 到 mAP0.95 間計算每步的 mAP 值並平均。從結果上可以看出本研究所提出的方法有最好的表現結果，mAP0.5 比進行影像降解析的方法高了 12%，比圖磚法高了 4%。根據評估結果，Focus-and-Detect 方式對於無人機影像的物件偵測是有顯著幫助的，而本研究所提出的子區域提出方式又比最直觀的子區域提出方式－圖磚法來的更有助益。而在有無經過 SOS 篩選這一部分，可以看見經過篩選後的模型表現與未經篩選的模型表現相差無已，若再考量在訓練時所需要花費的時間，透過 SOS 篩選的

doi:10.6342/NTU202404011

方法可以用較少的時間成本達到相似的模型表現。圖3.4展示了一些本研究所提出的方法在測試資料上的結果與影像降解析在測試資料上的結果的比較，可以看到在整體的物件偵測上，本研究所提出的方法能有更好的表現。能抓到更多的目標物，即便是重疊的兩不同目標物也能辨識，同時在較困難的項目上辨識的效果也比較好。但子區域的辨識模式的小缺點則是同一物件上可能會有多個物件框，而這與 NMS 所設定的閾值相關。

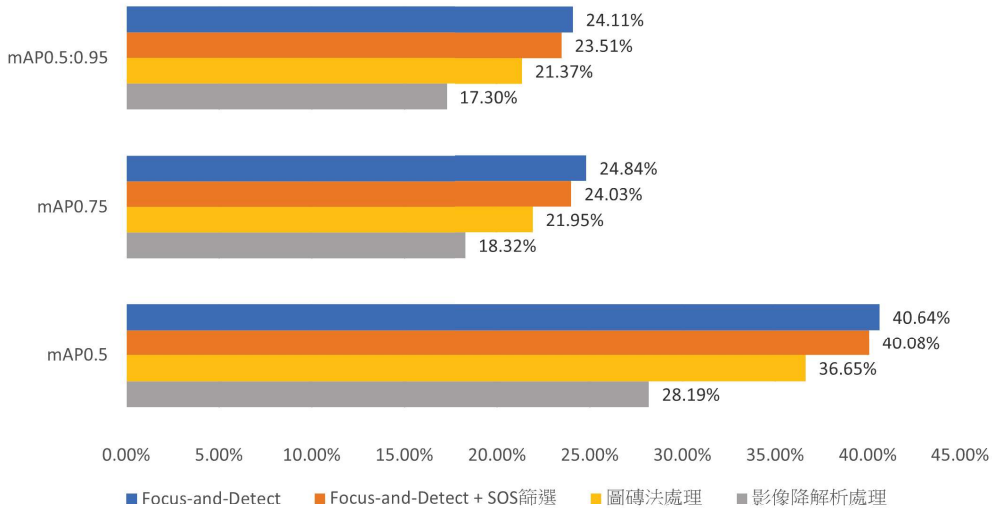


圖 3.3、本研究所提出的 Focus-and-Detect 方法、本研究所提出的 Focus-and-Detect 方法在透過 SOS 分數篩選、圖磚法處理與影像降解析處理的模型在測試資料的評估結果。

doi:10.6342/NTU202404011

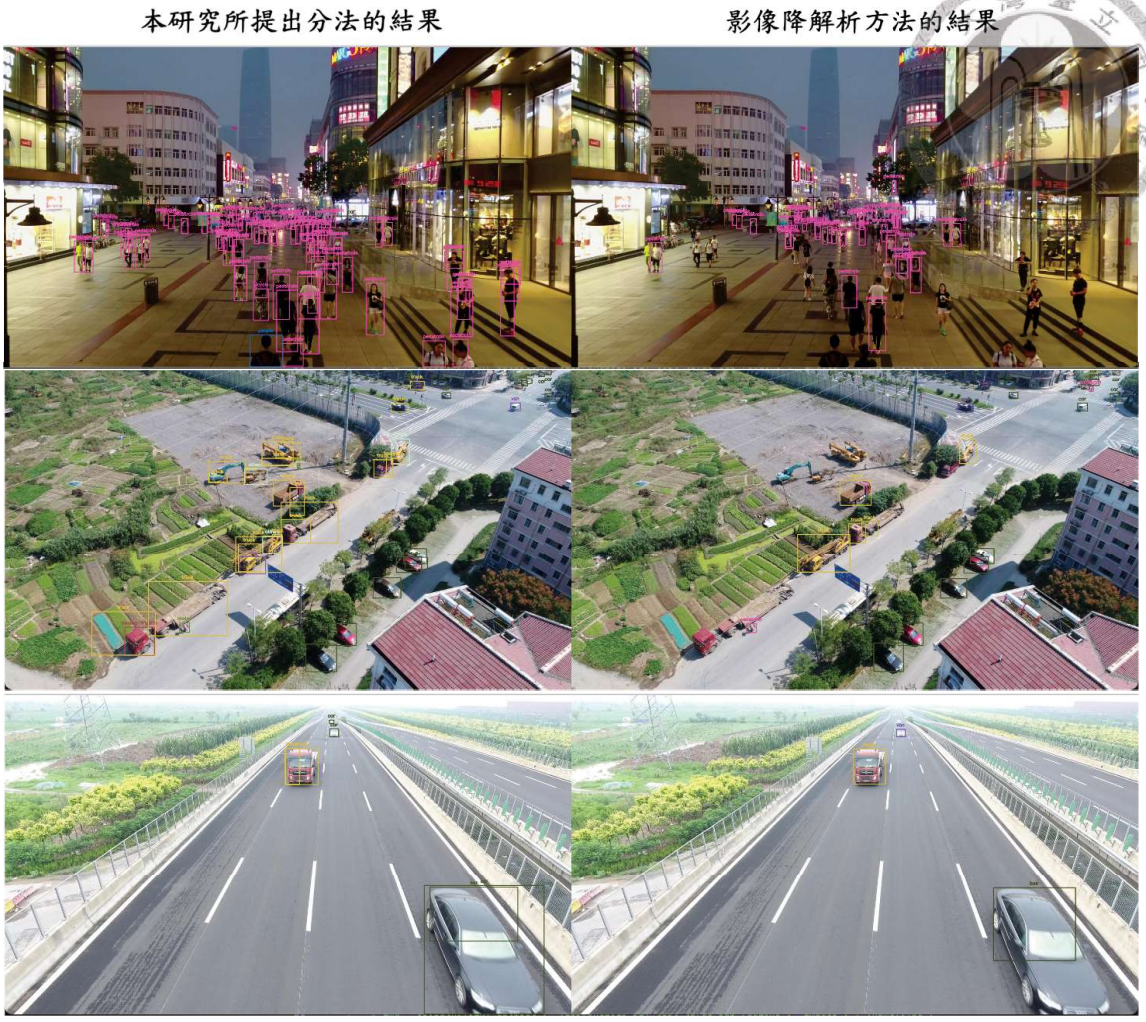


圖 3.4、本研究所提出的方法與影像降解析方法在測試資料集上的偵測結果比較。左邊為本研究所提出的方法之偵測結果。右邊為影像降解析方法的偵測結果。

在 VisDrone 的十個類別上，各自的評估結果如圖3.5所示。從各類別的表現上可看出本研究所提出的方法能幫助模型在相似物件的辨別上有更好的成果。在車、卡車與貨車這三項類別的表現上，四種實驗對於車的表現相去不遠，皆在 70% 以上。但影像降解析處理的模型在貨車與卡車的表現上則遠低於透過 Focus-and-Detect 與圖磚法處理的模型的表現，可見在經過降解析後，車、卡車與貨車這三者的特徵已被模糊的相似，不容易區分。但子區域方式訓練的模型則因為沒有降解析這一步驟，所以能更好的去區分這些特徵相似的物件。

doi:10.6342/NTU202404011

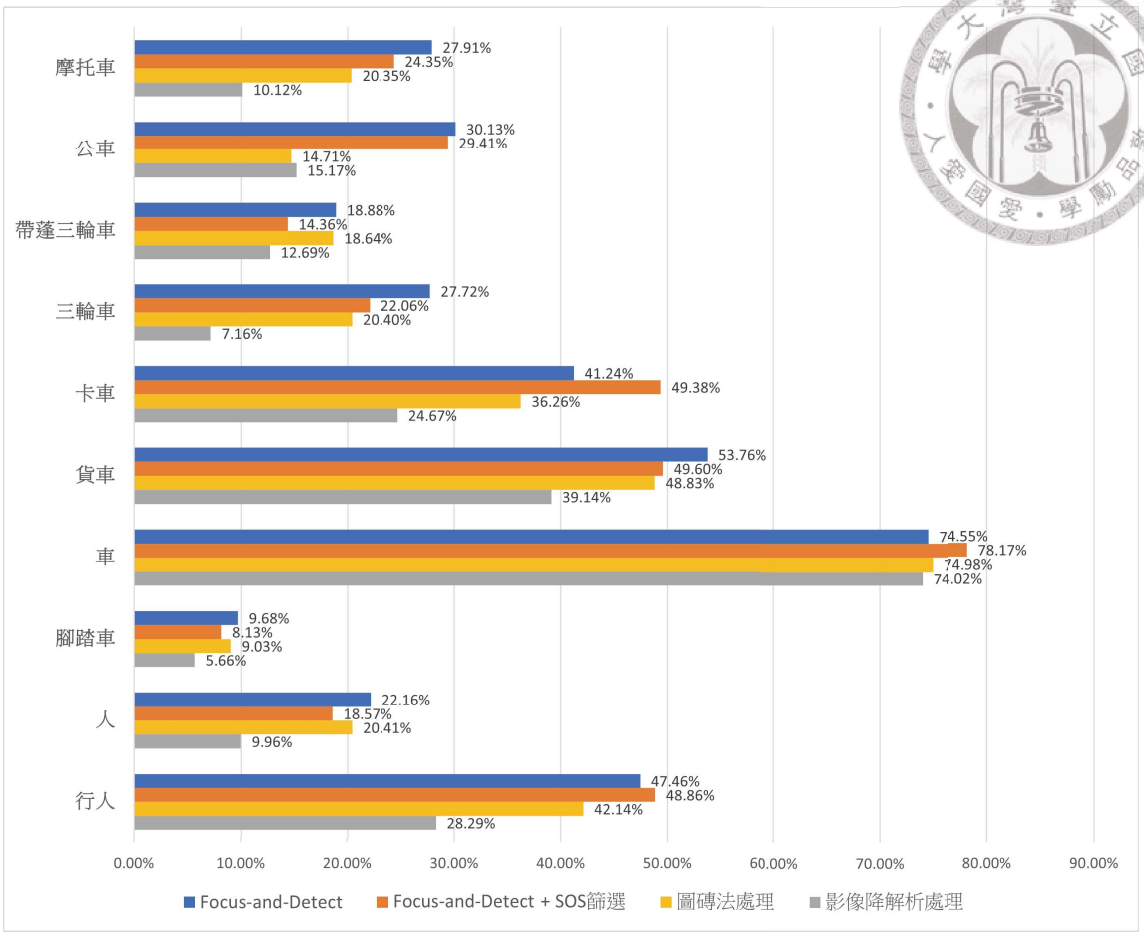


圖 3.5、本研究所提出的 Focus-and-Detect 方法、本研究所提出的 Focus-and-Detect 方法在透過 SOS 分數篩選、圖磚法處理與影像降解析處理的模型在測試資料中各類別的評估結果。

小結

本章節探討了本研究所提出的研究方法的實作結果。實驗以 VisDrone 資料集裡的高解析度影像進行，並分別採取影像降解析處理、圖磚法處理、本研究所提出的 Focus-and-Detect 方法處理、本研究所提出的 Focus-and-Detect 方法處理並以 SOS 分數進行子區域篩選，以上這四種實驗，並以其結果探討本研究所提出的方法是否相比於影像降解析處理能有更好的結果。從結果上，圖磚法與兩種 Focus-and-Detect 方法的表現相都有顯著的提升，而本研究所提出的 Focus-and-Detect 方法有最好的表現，更甚於圖磚法。而若再考量時間花費的話，以本研究所提出的 Focus-and-Detect 方法處理並以 SOS 分數進行子區域篩選的形

doi:10.6342/NTU202404011

式能夠在較少的訓練時間內達到相近的成果。總體而言，本研究所提出的方法能對無人機影像應用卷積神經網路類的物件偵測模型有所幫助，但須花費更多的時間在模型訓練與前處理上，是一個以時間換取表現的交換。

doi:10.6342/NTU202404011

第四章　結論與未來展望

4.1　成果討論

本研究深入研究無人機影像應用卷積神經網路類物件偵測模型的限制與挑戰，指出無人機影像因其高解析度與影像內目標物相對尺寸較小的特性，導致在進行物件偵測時會面臨 GPU 記憶體不足與衍伸而出的小物件偵測困境。本研究根據無人機影像的特性提出一依照 Focus-and-Detect 概念的優化辦法，先對影像進行子區域提出，後再針對子區域偵測。利用了無人機影像的高解析度優勢，以子區域提出的手段迴避掉小物件偵測的困境，而子區域的方式能在避免降解析的同時減少 GPU 記憶體的需求，在不提高硬體上限的情況下進行模型訓練，但因一張影像會被提成多個子區域，因此也提高了訓練所需的時間。

本研究在子區域提出上以 Selective Search 為基礎，透過限制 Selective Search 所提出的候選區域的大小，以及擴增候選區域的假想邊界來增加相鄰區域這兩項概念，延伸發展了 Modified Selective Search 作為子區域提出的演算法，在保留 Selective Search 的資料驅動與類別獨立的優點下，將原先演算法提出的適合物件識別的候選區域改為適合進行 Focus-and-Detect 的子區域，具備大小適合且內涵多個目標物。

而在子區域提出的方法上，本研究提出以平均地真資料涵蓋率來評估子區域提出方法的好壞，並以 SOS 分數來量化子區域的理想程度，區分出包含多個物

doi:10.6342/NTU202404011

件的理想子區域與不包含任何物件的無效子區域，使其在訓練階段的子區域提出中，能依靠 SOS 分數進行子區域的篩選，降低在訓練階段所提出的子區域數量，減少後續模型訓練所增加的時間。

最後在測試資料的評估上，除比較本研究所提出的方法與降解析處理的方法外，本研究也與最直觀的 Focus-and-Detect 的方法－圖磚法相比較。從實驗結果上證明，Focus-and-Detect 方法對於無人機影像的物件偵測能有顯著的提升，而本研究所提出的方法能獲得比圖磚法更好的表現結果，說明花費心力尋求更好的子區域提出能幫助整體的偵測結果。

總體而言，本研究所提出的方法是一個以時間換取表現的交換，也因為本研究的方法能降低 GPU 的記憶體需求，降低訓練所需的門檻，所以也是一個以時間換取資源的交換，用更長的訓練時間與更廉價的空間資源（子區域影像的儲存，一般硬碟空間）換取能使訓練運行在更便宜的 GPU 上，減少較昂貴的空間資源需求 (GPU 記憶體容量)。

4.2 未來展望

4.2.1 研究方法的改進

從研究成果來看，依照 Focus-and-Detect 概念進行的模型都獲得了更好的結果，Focus-and-Detect 能顯著的幫助無人機影像的物件偵測是被結果證實的。但去細究本研究提出的方法與圖磚法間的差異，一是在訓練階段本研究所提出的方法盡可能地去挑選理想的子區域來進行訓練，二是在測試階段影像中的目標物會被這些理想子區域多次的覆蓋，進而形成多餘觀測。這兩者對於物件偵測模型是否具有實質的幫助還可以再進行實驗分析，若在訓練階段其實子區域提出的方式幫

doi:10.6342/NTU202404011

助不大，或許訓練時以簡單、快速的圖磚法進行子區域提出即可，僅需在測試階段用更好的子區域提出方法來形成多餘觀測，如此能降低前處理所需的時間。

4.2.2 研究方法的延伸應用

雖然實驗上本研究因硬體限制無法使用原始解析度的無人機影像進行訓練，但若在 GPU 能負荷高解析度的情況下，透過 Focus-and-Detect 的方式去訓練是否還是能幫助模型表現，是一個值得考慮的問題。

另外本研究提出的方法也並不侷限應用在無人機上，但凡高畫質與影像中目標物較小的情況都適用，與無人機情況相似的衛星影像是否也能依照本研究所提出的概念去優化卷積神經網路的應用情形，也是值得繼續深究的問題。

doi:10.6342/NTU202404011

參考文獻

Akar, Ö. (2018). The rotation forest algorithm and object-based classification method for land use mapping through uav images. Geocarto international, 33(5):538–553.

AlpacaTechJP (2017, july 16). selective search python implementation. https://github.com/alpacatechjp/selectivesearch.

Chen, P.-C., Chiang, Y.-C., and Weng, P.-Y. (2020). Imaging using unmanned aerial vehicles for agriculture land use classification. Agriculture, 10(9):416.

Cheng, G., Yuan, X., Yao, X., Yan, K., Zeng, Q., Xie, X., and Han, J. (2023). Towards large-scale small object detection: Survey and benchmarks. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., and Zhang, L. (2021). Dynamic head: Unifying object detection heads with attentions. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 7373–7382.

Delavarpour, N., Koparan, C., Nowatzki, J., Bajwa, S., and Sun, X. (2021). A technical study on uav characteristics for precision agriculture applications and associated practical challenges. Remote Sensing, 13(6):1204.

Deng, S., Li, S., Xie, K., Song, W., Liao, X., Hao, A., and Qin, H. (2020). A global-local

doi:10.6342/NTU202404011

self-adaptive network for drone-view object detection. IEEE Transactions on Image Processing, 30:1556–1569.

Du, D., Zhu, P., Wen, L., Bian, X., Lin, H., Hu, Q., Peng, T., Zheng, J., Wang, X., Zhang, Y., et al. (2019). Visdrone-det2019: The vision meets drone object detection in image challenge results. In Proceedings of the IEEE/CVF international conference on computer vision workshops, pages 0–0.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. International journal of computer vision, 59:167–181.

Gao, Y., Liu, Y., Zhang, H., Li, Z., Zhu, Y., Lin, H., and Yang, M. (2020). Estimating gpu memory consumption of deep learning models. In Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pages 1342–1352.

Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al. (2018). Recent advances in convolutional neural networks. Pattern recognition, 77:354–377.

Iizuka, K., Itoh, M., Shiodera, S., Matsubara, T., Dohar, M., and Watanabe, K. (2018). Advantages of unmanned aerial vehicle (uav) photogrammetry for landscape analysis

doi:10.6342/NTU202404011

compared with satellite data: A case study of postmining sites in indonesia. Cogent Geoscience, 4(1):1498180.

Kandel, I. and Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. ICT express, 6(4):312–315.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

Lattanzi, D. and Miller, G. (2017). Review of robotic infrastructure inspection systems. Journal of Infrastructure Systems, 23(3):04017004.

Li, C., Yang, T., Zhu, S., Chen, C., and Guan, S. (2020). Density map guided object detection in aerial images. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pages 190–191.

Li, Z., Liu, F., Yang, W., Peng, S., and Zhou, J. (2021a). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12):6999–7019.

Li, Z., Liu, X., Zhao, Y., Liu, B., Huang, Z., and Hong, R. (2021b). A lightweight multi-scale aggregated model for detecting aerial images captured by uavs. Journal of Visual Communication and Image Representation, 77:103058.

Lin, R. (2022). Analysis on the selection of the appropriate batch size in cnn neural network. In 2022 International Conference on Machine Learning and Knowledge Engineering (MLKE), pages 106–109. IEEE.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and

doi:10.6342/NTU202404011

Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer.

Liu, M., Wang, X., Zhou, A., Fu, X., Ma, Y., and Piao, C. (2020). Uav-yolo: Small object detection on unmanned aerial vehicle perspective. Sensors, 20(8):2238.

Micheal, A. A., Vani, K., Sanjeevi, S., and Lin, C.-H. (2021). Object detection and tracking with uav data using deep learning. Journal of the Indian Society of Remote Sensing, 49:463–469.

Osco, L. P., de Arruda, M. d. S., Gonçalves, D. N., Dias, A., Batistoti, J., de Souza, M., Gomes, F. D. G., Ramos, A. P. M., de Castro Jorge, L. A., Liesenberg, V., et al. (2021). A cnn approach to simultaneously count plants and detect plantation-rows from uav imagery. ISPRS Journal of Photogrammetry and Remote Sensing, 174:1–17.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

Silvagni, M., Tonoli, A., Zenerino, E., and Chiaberge, M. (2017). Multipurpose uav for search and rescue operations in mountain avalanche events. Geomatics, Natural Hazards and Risk, 8(1):18–33.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Tijtgat, N., Van Ranst, W., Goedeme, T., Volckaert, B., and De Turck, F. (2017). Embedded real-time object detection for a uav warning system. In Proceedings of the IEEE international conference on computer vision workshops, pages 2110–2118.

doi:10.6342/NTU202404011

Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. International journal of computer vision, 104:154–171.

Ultralytics (2023, july 5). Yolov8 Documentation. https://github.com/ultralytics/ultralytics/tree/main?tab=readme-ov-file.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2023). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 7464–7475.

Xu, Y., Yu, G., Wang, Y., Wu, X., and Ma, Y. (2017). Car detection from low-altitude uav imagery with the faster r-cnn. Journal of Advanced Transportation, 2017(1):2823617.

Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into imaging, 9:611–629.

Yang, F., Fan, H., Chu, P., Blasch, E., and Ling, H. (2019). Clustered object detection in aerial images. In Proceedings of the IEEE/CVF international conference on computer vision, pages 8311–8320.

Yu, X., Gong, Y., Jiang, N., Ye, Q., and Han, Z. (2020). Scale match for tiny person detection. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 1257–1265.

Zhai, X., Huang, Z., Li, T., Liu, H., and Wang, S. (2023). Yolo-drone: an optimized yolov8 network for tiny uav object detection. Electronics, 12(17):3664.

周巧盈, 巫思揚, and 陳琦玲 (2020). 無人飛行載具之航拍影像應用於水稻倒伏災損判釋. 台灣農業研究, 69(1):25–45.

doi:10.6342/NTU202404011