EFFICIENT SHIP DETECTION ON LARGE OPEN SEA AREAS

Aitor J. García
Satellogic & Universitat Oberta de Catalunya
aitor.garcia@satellogic.com

Javier Marín
Satellogic & Universitat Oberta de Catalunya
javier.marin@satellogic.com

ABSTRACT

The ability to track and process large images while selectively focusing on areas of interest has become extremely important over the past few years. This capability enables a significant reduction in computational resources, which ultimately results in a decrease in costs when operating in the cloud or in energy when working with edge computing devices. A remote sensing task that can clearly benefit from such reduction consists in detecting ships in open seas. In such scenario, thousands of miles may only contain water without any vessels in sight. In this work, we introduce an efficient cascade architecture that can be effortlessly deployed in any machine. Our proposed approach builds upon prior related work and evaluates the current state-of-the-art image-based machine learning algorithms to accomplish our research objectives.

1 INTRODUCTION

Over the past few years, there has been a growing interest in using computer vision software and advanced image acquisition hardware to address the challenges of remote sensing Tahir et al. (2022). In this context, the development of faster and more accurate models has become a critical milestone in this field Sancho et al. (2021). These challenges are mainly driven by the need to keep pace with the rapid advancements in image acquisition hardware and to reduce costs by saving energy and computing power. Moreover, the increasing urgency to extract relevant information from the planet's surface and to take global actions highlights the prevailing need of using multi-and-hyper-spectral vision equipment, which generates voluminous and cumbersome data Hu et al. (2022). Consequently, the machine learning community is focusing its efforts on improving the computational efficiency and performance of current models to overcome these bottlenecks. This is particularly interesting because some of the tasks generally performed by these models are related to segmentation Oksuz et al. (2021)and object detection challenges Qin et al. (2022), which are computationally expensive tasks. To address this issue, a highly efficient model has been designed and built in this work, which can perform real or near real-time ship detection in large open sea areas. This model leverages state-of-the-art deep learning architectures for multi-spectral images, making it effective and precise.

2 METHODOLOGY

2.1 WORKFLOW OVERVIEW

This section aims to showcase the overall workflow of the system, which seeks to automatically find and identify ships in large scale maritime data. Some depictions will be done to show what the output data looks like as input data goes through each one of the modules of the pipeline. These modules are the following:

A sliding window algorithm (SW) that breaks down the large input data into smaller frames, similar to Akyon et al. (2022). A black pixel pruning algorithm (BP) that removes frames with too many black pixels. A water pruning algorithm (WP) that keeps only frames with a certain amount of water. A ship pruning algorithm (SP) that predicts if a frame has at least one big ship Wightman

(2019)Tan et al. (2020), and a ship detection algorithm (SD) that identifies the exact location and category of a ship in a frame Wang et al. (2022b).

All these modules work together to quickly locate and track ships in large images. The system uses simpler and faster algorithms to remove non-relevant images before using more complex algorithms for object detection, improving previous systems and real-time applications with satellite imagery Ozdemir & Polat (2020) Nalepa (2021) Paoletti et al. (2019). To build this system, we collected and processed large satellite rasters, in the form of **Red Green Blue (RGB)**, see Fig. 1, and **RGBN** imagery, which includes the near-infrared channel.

Figure 1: Large capture over the Mediterranean sea.

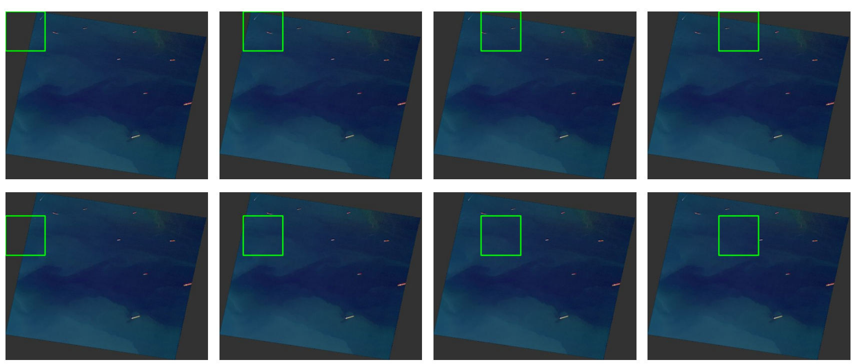

Figure 2: Sliding window using a 1024 x 1024 window size and a stride of 614 pixels.

To start, step one involves using a sliding window, see Fig. 2, to cut a large raster into smaller frames for easier handling. However, some frames located at the edge of the raster may be entirely black, which is addressed in step two.

By using the module BP, frames are kept only if they have a certain percentage of non-black pixels. A percentage of around 90-95% is suitable, as it prevents the discarding of frames that might contain objects despite the black pixels at the edge of the raster. For the purpose of illustration, a **Canny Edge** detector Ding & Goshtasby (2001) has been used, as can be seen in Fig. 2, to emphasize the difference between the left part of the edge, which is the end of the raster, and the part at the right of the edge, which is where the fully black pixels are.

In step three, **Near Infrared (NIR)** sensors are utilized to take advantage of water's light absorption properties and discard non-water areas.

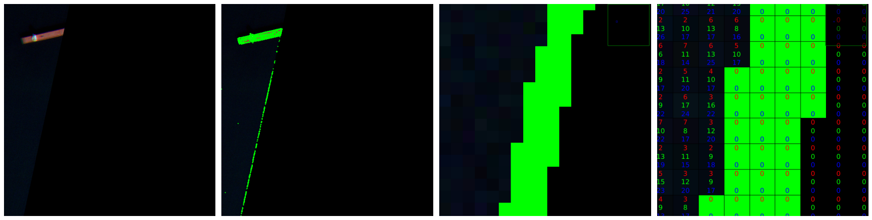

Figure 3: The frame contains a ship located at the edge of the raster

In step four, a state-of-the-art **Convolutional Neural Network (CNN)** Albawi et al. (2017) algorithm is utilized to predict whether a frame corresponds to the class ”ship” or ”no ship.” The algorithm is trained using an **EfficientNet** Koonce & Koonce (2021) architecture, see Fig. 4, which automatically extracts the relevant features of a ship and classifies it as one of the two possible classes. If a frame class happens to be ”ship”, it is kept for the final step.

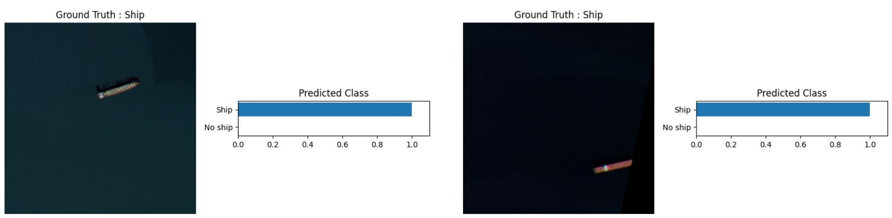

Figure 4: Frames labeled as ”ship” with EfficientNet

Finally, among the different methods available in the literature used for ship detection (e.g. ReDet Han et al. (2021), Oriented R-CNN Xie et al. (2021), RR-CNN Liu et al. (2017)), we adapted YOLOv7 Wang et al. (2022a) to locate ships using rotated bounding-boxes in step five. This decision was made after most recent improvements authors reported in their paper. As can be seen in Fig. 5, this fork is able to predict rotated bounding boxes (x1,y1, x2,y2, x3,y3, x4,y4), unlike the original version, and processes independent frames rapidly. This module is initialized at the beginning of the execution and waits for a frame before providing a real-time prediction.

Figure 5: Detected ships with YOLOv7.

2.2 Training, Validation and Testing

Except for the ship detector that was trained and validated using the full **DOTA v2** dataset Xia et al. (2018) (see Fig. 6), the rest of the stages within the cascade were trained using Satellogic multispectral imagery, L1 product[1], at one meter resolution. EfficientNet was trained and validated on a custom dataset containing around 500 ships, 80% for training and 20% for validation. To assess the full cascade vs YOLOv7, we carefully split several captures into validation and testing, leaving a total of 3,291 km² for testing. Quantitative results are shown in Table 1. Regarding training times, EfficientNet took 20 minutes, while YOLOv7 took 33 hours, in both cases, using a single Nvidia RTX A5000.

[1]https://satellogic.com/products/multispectral-imagery/

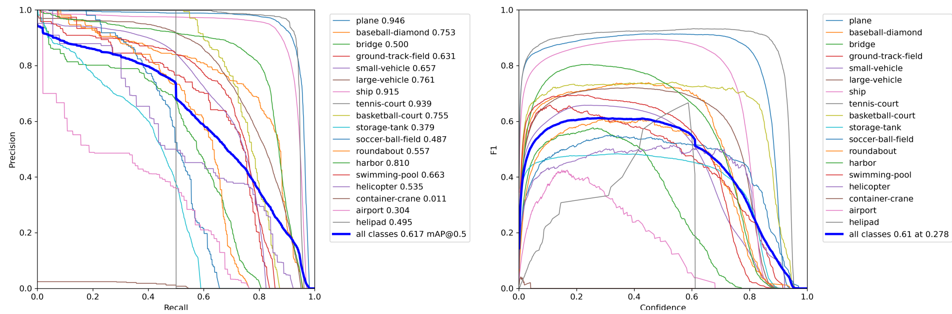

Figure 6: Precision vs Recall and F1-Score vs Confidence curves after having trained YOLOv7.

2.3 PROPOSED (MCPN) VS BASELINE (YOLOV7) PIPELINES

Outputs	**MCPN**	**YOLOv7**
Area (km²)	3,291	3,291
Sliding Window time (s)	0.51	0.40
Sliding Window input frames	8,980	8,980
Black Pruning time (s)	20.33	20.18
Black Pruning input frames	8,980	8,980
Water Pruning time(s)	3.72	-
Water Pruning frames	3,836	-
Ship Pruning time(s)	27.26	-
Ship Pruning input frames	3,440	-
Normalizations and conversions time (s)	34.00	37.90
Ship Detection time (s)	32.81	121.42
Ship Detection input frames	1,002	3,836
Total time (s)	**118.63**	**179.90**
Sliding Window frames per second	**75.69**	**49.91**

Table 1: *Test results with proposed cascade vs YOLOv7.*

The pipeline has included sea and ship pruning modules, resulting in an average frame-rate of 75.69 FPS, which represents a speed increase of approximately 51.65% compared to the baseline algorithm that solely utilizes YOLOv7 without EfficientNet and the WP module. However, there are still a small number of false negatives in some images when compared to the baseline algorithm. Out of the total of 336 cropped positives processed with YOLOv7, there are 23 false negatives using Maritime Candidate Pruning Network (MCPN), which accounts for roughly 6.8% of errors

3 CONCLUSIONS

As a result of comparing the baseline algorithm with the proposed algorithm, it can be concluded that the proposed algorithm has improved the performance in terms of speed by approximately 51.65% and has a slightly higher error of 6.8% in different weather conditions. This error originates exclusively from the ship pruning module based on EfficientNet, and it is currently being improved by increasing the amount of data in the current dataset. Efforts are also underway to address the impact of having clouds and sunglint in the images in the future. In the former case, we believe that these cases could be dealt in previous stages without the need of having a cloud detector. In the second case, we could either include more cases in the dataset, so the full pipeline could learn from them, or use image processing techniques such as filters or contrast enhancement to mitigate the sunglint. Maritime Candidate Pruning Network is an end-to-end solution that fully integrates and leverages different well-known computer vision techniques along with current state-of-the-art deep learning architectures to conduct real-time analysis.

REFERENCES

Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing aided hyper inference and fine-tuning for small object detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 966–970, 2022. doi: 10.1109/ICIP46576.2022.9897990.

Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pp. 1–6. IEEE, 2017.

Lijun Ding and Ardeshir Goshtasby. On the canny edge detector. *Pattern recognition*, 34(3):721–725, 2001.

Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. Redet: A rotation-equivariant detector for aerial object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2786–2795, June 2021.

Xing Hu, Chun Xie, Zhe Fan, Qianqian Duan, Dawei Zhang, Linhua Jiang, Xian Wei, Danfeng Hong, Guoqiang Li, Xinhua Zeng, Wenming Chen, Dongfang Wu, and Jocelyn Chanussot. Hyperspectral Anomaly Detection Using Deep Learning: A Review. *Remote Sensing*, 14(9):1973, April 2022. ISSN 2072-4292. doi: 10.3390/rs14091973. URL `https://www.mdpi.com/2072-4292/14/9/1973`.

Brett Koonce and Brett Koonce. Efficientnet. *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, pp. 109–123, 2021.

Zikun Liu, Jingao Hu, Lubin Weng, and Yiping Yang. Rotated region based cnn for ship detection. In *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 900–904. IEEE, 2017.

Jakub Nalepa. Recent Advances in Multi- and Hyperspectral Image Analysis. *Sensors*, 21(18):6002, January 2021. ISSN 1424-8220. doi: 10.3390/s21186002. URL `https://www.mdpi.com/1424-8220/21/18/6002`. Number: 18 Publisher: Multidisciplinary Digital Publishing Institute.

Kemal Oksuz, Baris Can Cam, Emre Akbas, and Sinan Kalkan. Rank & Sort Loss for Object Detection and Instance Segmentation, August 2021. URL `http://arxiv.org/abs/2107.11669`. arXiv:2107.11669 [cs].

Akin Ozdemir and Kemal Polat. Deep Learning Applications for Hyperspectral Imaging: A Systematic Review. *Journal of the Institute of Electronics and Computer*, 2:39–56, January 2020. doi: 10.33969/JIEC.2020.21004.

M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza. Deep learning classifiers for hyperspectral imaging: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 158:279–317, December 2019. ISSN 0924-2716. doi: 10.1016/j.isprsjprs.2019.09.006. URL `https://www.sciencedirect.com/science/article/pii/S0924271619302187`.

Zheng Qin, Zeming Li, Zhaoning Zhang, Yiping Bao, Gang Yu, Yuxing Peng, and Jian Sun. ThunderNet: Towards Real-time Generic Object Detection, March 2022. URL `http://arxiv.org/abs/1903.11752`. arXiv:1903.11752 [cs].

Jaime Sancho, Manuel Villa, Gemma Urbanos, Marta Villanueva, Pallab Sutradhar, Gonzalo Rosa, Alberto Martin, Guillermo Vazquez, Miguel Chavarrias, Ruben Salvador, Alfonso Lagares, Eduardo Juarez, and Cesar Sanz. An Embedded GPU Accelerated Hyperspectral Video Classification System in Real-Time. In *2021 XXXVI Conference on Design of Circuits and Integrated Systems (DCIS)*, pp. 1–6, Vila do Conde, Portugal, November 2021. IEEE. ISBN 978-1-66542-116-4. doi: 10.1109/DCIS53048.2021.9666171. URL `https://ieeexplore.ieee.org/document/9666171/`.

Arsalan Tahir, Hafiz Munawar, Junaid Akram, Muhammad Adil, Shehryar Ali, Abbas Kouzani, and M. Mahmud. Automatic Target Detection from Satellite Imagery Using Machine Learning. *Sensors*, 22(3):1147, February 2022. ISSN 1424-8220. doi: 10.3390/s22031147. URL `https://www.mdpi.com/1424-8220/22/3/1147`.

Mingxing Tan, Ruoming Pang, and Quoc V. Le. EfficientDet: Scalable and Efficient Object Detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10778–10787, Seattle, WA, USA, June 2020. IEEE. ISBN 978-1-72817-168-5. doi: 10.1109/CVPR42600.2020.01079. URL https://ieeexplore.ieee.org/document/9156454/.

Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022a.

Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, July 2022b. URL http://arxiv.org/abs/2207.02696. arXiv:2207.02696 [cs].

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3974–3983, 2018.

Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. Oriented r-cnn for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3520–3529, October 2021.