

# FPGA-Based Hardware Acceleration for Real-Time Maritime Surveillance and Monitoring Onboard Spacecraft

Giovanni Maria Capuano<sup>\*1,2</sup>, Salvatore Capuozzo<sup>1</sup>, Antonio G.M. Strollo<sup>1</sup>, Nicola Petra<sup>1</sup>

<sup>1</sup> *Università Degli Studi di Napoli Federico II, Naples, Italy*

<sup>2</sup> *TSD-Space, Pozzuoli, Italy*

Accurate vessel detection and timely information extraction from optical remote sensing imagery are essential for a wide range of maritime surveillance operations, both civilian and defense-related. These operations include vessel tracking, unauthorized fishing, illegal migration monitoring, and search and rescue missions. Although artificial intelligence (AI) is a key component for achieving reliable and accurate detection in satellite imagery, traditional AI-based remote sensing methodologies rely on ground-based image processing. This dependence leads to significant delays between data acquisition and the generation of actionable insights, which may hinder rapid decision-making during critical maritime situations such as sea disasters. To address this challenge, we propose a novel hardware design based on the Microchip PolarFire System-on-Chip for low-power and real-time vessel detection onboard spacecraft. Our design leverages Microchip's CoreVectorBlox, implemented on the programmable logic, to accelerate the inference process of SR-YOLOv5s, an enhanced YOLOv5s-based object detection framework. This detector incorporates a super-resolution backbone that allows the extraction of fine details and features of small targets of interest, thus improving detection accuracy. The results confirm the effectiveness of our approach, showcasing its potential as a solution to enable real-time alerts in the maritime surveillance domain through Earth Observation (EO) image processing at the edge.

## 1 Introduction

Maritime monitoring and timely detection of significant hazards, both natural and man-made, are crucial for emergency management and the prevention of further incidents at sea. Rapid and reliable information empowers civil protection organizations and authorities to promptly organize effective rescue operations, potentially saving lives and reducing damages [1]. Maritime surveillance also plays a critical role in detecting illegal activities like unregulated fishing or illegal immigration, allowing security forces to quickly

respond and enforce justice [2]. Satellite remote sensing (RS) through Earth Observation (EO) is invaluable for efficiently monitoring global oceans, with sensors on satellites capturing large areas with high spatial and temporal resolution [3]. With the rapid advancement in EO technology, a massive number of remote sensing optical images are now available.

Recent advances in Deep Learning (DL) have significantly boosted object detection (OD) in electro-optical satellite imagery, outperforming traditional methods and making it a key research area [4]. This technology has achieved notable results in detecting ships and vessels [5]. However, detecting small objects may be a complex task due to the small size of targets and limited spatial resolution of the imagery, often leading to minimal object pixelation and consequently poor feature extraction, unsatisfactory detection performance, and frequent misidentifications. To address this issue, researchers are exploring various strategies [6], including super-resolution techniques that enhance imagery beyond its native resolution to improve the detection accuracy [7].

Despite the recent advancements in OD, generating rapid alerts remains challenging due to traditional remote sensing methods that depend on ground-based image processing, resulting in delays between data acquisition and the delivery of actionable insights. Limited down-link bandwidth and ground station availability impede rapid analytics. Consequently, there is increasing interest in processing operations directly onboard spacecraft to reduce latency and data bandwidth needs, enabling near-real-time downlinks that focus solely on critical actionable data, such as alerts for vessel positions [8]. However, DL-based processing often demands significant power consumption and extensive memory due to model complexity, presenting challenges for space systems with strict resource constraints [9].

For ground applications, the advancement of AI was enabled by the availability of high-performance

<sup>\*</sup>Corresponding author. E-Mail: giovanni-maria.capuano@unina.it

GPUs. Due to their high power consumption, the use of GPUs is limited in energy-sensitive applications [10]. Newer power-efficient AI engines, like Embedded GPUs and ASIC-based platforms, allow for low-power inference at the edge with remarkable performance [11]. Commercial off-the-shelf (COTS) devices are cost-effective and significantly reduce development costs for missions. However, they are built with commercial silicon technology, which does not meet the stringent radiation-tolerant requirements of more demanding space missions [12]. FPGAs, instead, have been adopted in space for several decades; moreover, although in a very limited number, rad-hard space-grade FPGAs are now available on the market and are able to tolerate the space environment even in demanding missions, such as long-duration in GEO and above.

Microchip recently presented the Radiation Tolerant (RT) PolarFire SoC [13], an innovative device that integrates a real-time, Linux-capable, RISC-V based microprocessor with RT PolarFire FPGA fabric. It features a non-volatile flash-based technology that makes this device more tolerant to radiation-induced upset than SRAM-based devices [14]. The FPGA contains a matrix of configurable blocks that can be customized to perform specific computing tasks with high flexibility, including inference. Compared to other standalone AI solutions, the FPGA can be designed to simultaneously host VHDL modules for sensor interfacing and data handling functionalities, making it particularly suitable for low-power remote sensing applications. For deploying NNs on its devices, Microchip provides the CoreVectorBlox, an FPGA-based hardware accelerator, tailored to speed up convolutional layers. It is designed with an overlay architecture that allows the deployment of different neural networks onboard without the need to reconfigure the programmable logic resources [14]. Techniques such as pruning and quantization enable the efficient running of very deep NNs with high power efficiency and no significant loss of accuracy [14] [15]. In addition, the FPGA can be reprogrammed even in-flight, providing satellite operators with the ability to change processing algorithms after a satellite has been launched.

In this paper, we first explore the application of super-resolution (SR) techniques in the vessel detection task to improve detection accuracy on satellite imagery. We introduce SR-YOLOv5s, an enhanced YOLOv5s network [16] that incorporates a SR backbone, facilitating the extraction of fine details and small objects. Subsequently, we explore the Microchip VectorBlox SDK flow [17], a software development kit provided by Microchip to deploy neural networks on its FPGAs, through the CoreVectorBlox engine.

Finally, we deploy our model into Microchip's Po-

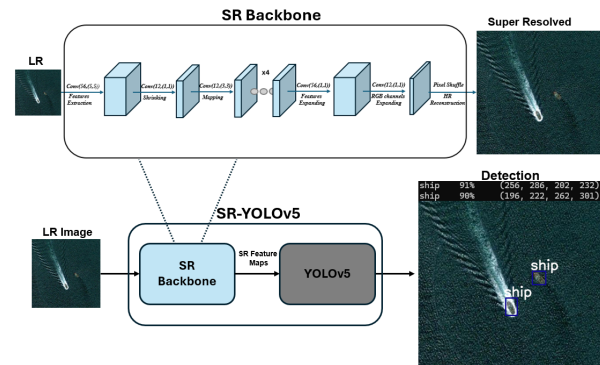


Figure 1: SR-YOLOv5 Architecture.

larFire SoC device to evaluate its inference performance, providing a benchmark with the embedded NVIDIA Jetson Orin Nano GPU and NVIDIA A100 GPU, in terms of inference accuracy, speed, and power consumption.

## 2 Methods

### 2.1 SR-YOLOv5s

OD enables the identification and localization of objects within an image by recognizing them from various classes and determining their boundaries with bounding boxes. Since its introduction in 2015, the YOLO (You Only Look Once) versions [18] have rapidly grown, with YOLOv5 being one of the most recent and widely used [19]. This detector allows solving classification and regression in a single pass, significantly reduces the computational demand. The model is available in a flexible range of sizes, and its smaller variant (YOLOv5s), with 7.5 million parameters, delivers impressive results thanks to its efficiency and accuracy [20].

SR techniques aim to enhance the resolution and visual quality of low-resolution (LR) images by reconstructing fine spatial details. DL has shown remarkable performance in SR tasks, surpassing traditional fixed kernel-based methods [21]. The SRCNN [22] is a pioneering deep learning model designed to enhance LR images by employing a fully convolutional network that reconstructs high-resolution details using bicubic interpolated input. The Fast SRCNN [23] improves the processing speed and enhances image restoration quality. This model directly takes an LR image as input and performs up-scaling at the final stage using a deconvolutional layer.

We introduce SR-YOLOv5s, an advanced version of the YOLOv5s object detector that integrates a super-resolution (SR) backbone into its architecture. The model reconstructs input image spatial details, sig-

nificantly improving its object detection capabilities without excessively increasing the inference time. The SR backbone is based on the Fast SRCNN architecture, but adopts the Rectified Linear Unit (ReLU) activation function and performs additional convolution and pixel-shuffle operations for maps aggregation and SR image reconstruction.

## 2.2 Dataset and Training Details

We developed a custom dataset from ShipRSImageNet [24], designed for ship detection in high-resolution (HR) optical remote sensing imagery. It consists of 3,435 RGB images from various sensors, satellite platforms, locations, and seasons. To create HR ground truth images for the SR backbone's training and evaluation, we resized the original images to 416x416x3 while preserving the aspect ratio. We generated low-resolution (LR) inputs (208x208x3) by applying a downsampling operation, which involved convolving the HR images with a Gaussian Filter [25]. Focusing exclusively on the 'ship' class, we adjusted the bounding boxes to accommodate the new dimensions. The dataset is allocated as follows: 80% for training, 15 % for validation, and 5% for testing.

Training, validation, and testing were conducted using the NVIDIA A100 GPU, available on the cloud-computing platform Google Colab. In more detail, we first tuned the SR Backbone's weights for a total of 150 epochs, following no observed improvement in the Mean Square Error (MSE) loss function. Finally, by applying transfer learning to the backbone, we subsequently trained the entire SR-YOLOv5s model for a total of 150 epochs, following the training specifications described by the authors of YOLOv5.

## 2.3 PolarFire SoC Deployment

The CoreVectorBlox is an FPGA coprocessor featuring a soft RISC-V microcontroller, VectorBloxMXP vector processor, and a 2D processing element grid for convolutional layers [26]. It offers various size configurations to balance FPGA utilization and performance. Our design utilizes the V1000 configuration for optimal parallel computation. The VectorBlox SDK translates neural network frameworks into a Binary Large Object (BLOB) [27], which is loaded into flash memory accessible by the core memory-mapped master. This core reads the BLOB and network inputs from memory, transfers them to DDR memory, processes the network, and outputs the results to a buffer. The inference process is managed by the hard-core RISC-V processor within the SoC, under the supervision of a Linux-based OS that controls, through application software, the data flow and manages the AI engine.

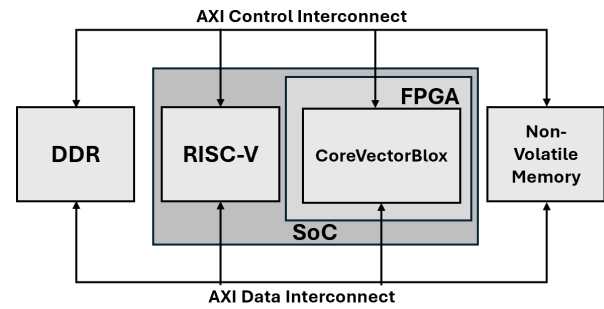


Figure 2: CoreVectorBlox-based Hardware Design

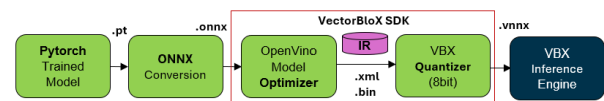


Figure 3: VectorBloxSDK Flow

Figure 2 illustrates the overall hardware design. Hardware validation is conducted using the Microchip PolarFire SoC Video Kit, a development board designed for vision-based applications and equipped with the PolarFire MPFS250TS. The entire design requires a total of 72% of the available logical elements (LEs), of which about 70% is allocated to implementing the functionalities of the CoreVectorBlox.

The architecture of YOLOv5s, which forms the SR-YOLOv5s network, has been modified by replacing the SiLU activation function with ReLU to make it compatible with VectorBlox SDK. Figure 3 shows the tool flow for generating a BLOB from a neural network model description. Initially, the SR-YOLOv5 model described in Pytorch is converted to ONNX format. The VectorBlox SDK optimizer then transforms the ONNX model into an intermediate representation (IR), removing layers like batch normalization and dropout, which are used only during training. The model is further optimized by converting from 32-bit floating-point (FP32) to 8-bit integer (INT8) precision. This quantization utilizes dynamic fixed-point quantization, which achieves memory efficiency with minimal precision loss.

## 3 Results

We evaluate the super-resolution capabilities of the SR Backbone on a test set using two prevalent SR metrics: Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [28]. Despite having significantly fewer parameters (98% less than the Enhanced Deep Residual Network (EDRN) [29]), the SR Backbone exhibits marginally lower performance compared to other models such as SRCNN, Fast SRCNN, and EDRN, the latter representing a state-of-

the-art model for the single image super-resolution task. Designed for compactness and computational efficiency, the SR Backbone minimally impacts the overall computational load of the SR-YOLOv5s algorithm. Table 1 details the performance comparison between the models.

Model	Param.	PSNR	SSIM
SRCNN	20.099	31.668	0.880
Fast SRCNN	24.683	31.978	0.889
EDRN	1.395.788	32.79	0.898
SR BackBone	16.960	30.64	0.884

**Table 1:** Super Resolution Results

Table 2 compares the detection performance of original YOLOv5s on low-resolution and high-resolution datasets using COCO metrics, including mAP 50 and mAP 50:95. It also presents the accuracy values achieved by the YOLOv5s architecture with ReLU activation, and the performance obtained by our SR-YOLOv5s network.

Model	Input Size	Param.	mAP50	mAP50:95
YOLOv5s Original	208	7.012.822	0.836	0.590
YOLOv5s Original	416	7.012.822	0.935	0.792
YOLOv5s ReLU	416	7.012.822	0.920	0.761
SR-YOLOv5s	208	7.029.782	0.909	0.746

**Table 2:** Vessel Detection Accuracy

The original YOLOv5s demonstrates superior object detection capability, but its performance declines with decreasing input resolution, specifically at high IoU thresholds. This result confirms that detecting small objects is more challenging at lower spatial resolutions. However, integrating the super-resolution backbone enhances YOLOv5s' detection performance on low-resolution images by 26.4%. Using SR-YOLOv5s, the detection accuracy for low-resolution images closely matches that of the original model, with only a slight decrease in mAP. Modifying the SiLU activation function in ReLU results in a minimal mAP reduction.

Table 3 compares the performance of SR-YOLOv5s deployed on the Embedded GPU Jetson Orin Nano, NVIDIA GPU A100, and SoC PolarFire. The table

Device	Precision	mAP50	Inference Speed	Avg. Power Consumption	Avg. Energy Consumption
A100	FP32	0.909	10.92 ms	~ 53.04 W	~ 579.2 mJ/frame
Jetson Orin Nano	FP32	0.909	31.84 ms	~ 9.6 W	~ 305.7 mJ/frame
PolarFire SoC	INT-8	0.810	85 ms	~ 2.46 W	~ 209.1 mJ/frame

**Table 3:** AI Engines Comparative Performance Analysis



**Figure 4:** SR-YOLOv5 vessel detection

focuses on accuracy, inference speed, and estimated power and energy consumption.

Despite the quantization process, detection results on the test set closely match those with full precision with a slight drop in accuracy. The quantized network operates at approximately 12 fps but with significantly lower power consumption compared to the A100 GPU and the embedded Jetson Orin Nano GPU. GPUs typically operate at very high clock frequencies, 1-2 GHz or higher, enabling several operations per second. They are optimized for floating-point operations, with many parallel processing units and high-bandwidth memory, though at the cost of increased power consumption. In contrast, FPGAs operate at lower clock frequencies. Our FPGA design includes two clocks for the CoreVectorBlox: a system clock and a double-frequency clock for the internal math blocks and LSRAM of 125 MHz and 250 MHz, respectively. Despite reducing model complexity through quantization and configuring the programmable logic to accelerate neural networks, the lower clock frequency limits the number of operations per timeframe, resulting in longer inference times. However, the PolarFire SoC's low power consumption still ensures lower energy use per frame compared to the GPUs.

Figure 4 shows the vessel detection performed by SR-YOLOv5s on the PolarFire SoC.

## 4 Discussion

In this paper, we introduce the SR-YOLOv5s object detection algorithm, enhancing YOLOv5s' accuracy for

detecting small vessels in remote sensing images. The original model exhibits different detection results for two images with the same objects and features but different sizes. Integrating an SR backbone into the YOLOv5s architecture, we improve feature extraction, and small vessel detection with minimal impact on detection speed. Additionally, we evaluated the benefits of an FPGA-based architecture for accelerating CNN inference, offering a balanced trade-off between inference time, accuracy, and power consumption, suitable for edge computing. The VectorBlox SDK and CoreVectorBlox IP offer a user-friendly and flexible method for deploying neural networks on the programmable logic of PolarFire SoCs. Furthermore, the latest RT SoC PolarFire provides a valuable solution for deploying AI applications in space, even for missions requiring high reliability in the space environment, making these devices efficient processors for AI deployment onboard spacecraft.

## References

1. Denis, G. *et al.* The evolution of Earth Observation satellites in Europe and its impact on the performance of emergency response services. *Acta Astronautica* **127**, 619–633 (2016).
2. Kanjir, U., Greidanus, H. & Oštir, K. Vessel detection and classification from spaceborne optical images: A literature survey. *Remote sensing of environment* **207**, 1–26 (2018).
3. Melet, A. *et al.* Earth observations for monitoring marine coastal hazards and their drivers. *Surveys in Geophysics* **41**, 1489–1534 (2020).
4. Gui, S., Song, S., Qin, R. & Tang, Y. Remote Sensing Object Detection in the Deep Learning Era—A Review. *Remote Sensing* **16**, 327 (2024).
5. Bo, L., Xiaoyang, X., Xingxing, W. & Wenting, T. Ship detection and classification from optical remote sensing images: A survey. *Chinese Journal of Aeronautics* **34**, 145–163 (2021).
6. Cheng, G. *et al.* Towards large-scale small object detection: Survey and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).
7. Shermeyer, J. & Van Etten, A. The effects of super-resolution on object detection performance in satellite imagery in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2019), 0–0.
8. Del Prete, R., Meoni, G., Longépé, N., Graziano, M. D. & Renga, A. First Results of Vessel Detection with Onboard Processing of Sentinel-2 Raw Data by Deep Learning in IGARSS 2023–2023 IEEE International Geoscience and Remote Sensing Symposium (2023), 6262–6265.
9. Ghiglione, M. *et al.* Machine learning application benchmark for in-orbit on-board data processing in European Workshop on On-Board Data Processing (2021).
10. Giuffrida, G. *et al.* CloudScout: A deep neural network for on-board cloud detection on hyperspectral images. *Remote Sensing* **12**, 2205 (2020).
11. Furano, G. *et al.* Towards the use of artificial intelligence on the edge in space systems: Challenges and opportunities. *IEEE Aerospace and Electronic Systems Magazine* **35**, 44–56 (2020).
12. Rapuano, E. *et al.* An fpga-based hardware accelerator for cnns inference on board satellites: benchmarking with myriad 2-based solution for the cloudscout case study. *Remote Sensing* **13**, 1518 (2021).
13. Microchip. RT PolarFire SoC <https://www.microchip.com/en-us/products/fpgas-and-plds/radiation-tolerant-fpgas/rt-polarfire-soc> (2024).
14. Dadà, M., Zulberti, L., Nannipieri, P., Fanucci, L. & Moranti, S. Inference and Evaluation of Deep Convolutional Neural Networks on Microchip's Hardware Accelerator VectorBlox in 2023 European Data Handling & Data Processing Conference (EDHPC) (2023), 1–8.
15. Dettmers, T. 8-bit approximations for parallelism in deep learning. *arXiv preprint arXiv:1511.04561* (2015).
16. Ultralytics. YOLOv5 <https://github.com/ultralytics/yolov5> (2024).
17. O'Neill, K., Severance, A. & Nandi, D. Using the VectorBlox software development kit to create programmable AI/ML applications in radiation-tolerant RT PolarFire FPGAs in *Proc. Eur. Workshop Board Data Process.(OBDP)* (2021), 1–8.
18. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, real-time object detection in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 779–788.
19. Hussain, M. YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. *Machines* **11**, 677 (2023).
20. Jiang, P., Ergu, D., Liu, F., Cai, Y. & Ma, B. A Review of Yolo algorithm developments. *Procedia computer science* **199**, 1066–1073 (2022).
21. Lepcha, D. C., Goyal, B., Dogra, A. & Goyal, V. Image super-resolution: A comprehensive review, recent trends, challenges and applications. *Information Fusion* **91**, 230–260 (2023).
22. Dong, C., Loy, C. C., He, K. & Tang, X. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence* **38**, 295–307 (2015).
23. Dong, C., Loy, C. C. & Tang, X. Accelerating the super-resolution convolutional neural network in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II* **14** (2016), 391–407.
24. Zhang, Z., Zhang, L., Wang, Y., Feng, P. & He, R. ShipRSImageNet: A Large-Scale Fine-Grained Dataset for Ship Detection in High-Resolution Optical Remote Sensing Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **14**, 8458–8472 (2021).
25. Shi, W. *et al.* Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 1874–1883.
26. Microchip. MicrosemiCoreVectorBloxIPHandbook [https://github.com/Microchip-VectorBlox/VectorBlox-SDK/blob/master/docs/Microsemi\\_CoreVectorBlox\\_IP\\_Handbook.pdf](https://github.com/Microchip-VectorBlox/VectorBlox-SDK/blob/master/docs/Microsemi_CoreVectorBlox_IP_Handbook.pdf) (2024).
27. Microchip. VectorBlox Programmer Guide [https://github.com/Microchip-VectorBlox/VectorBlox-SDK/blob/master/docs/Microsemi\\_CoreVectorBlox\\_IP\\_Handbook.pdf](https://github.com/Microchip-VectorBlox/VectorBlox-SDK/blob/master/docs/Microsemi_CoreVectorBlox_IP_Handbook.pdf) (2024).
28. Wang, P., Bayram, B. & Sertel, E. A comprehensive review on deep learning based remote sensing image super-resolution methods. *Earth-Science Reviews* **232**, 104110 (2022).



29. Lim, B., Son, S., Kim, H., Nah, S. & Mu Lee, K. *Enhanced deep residual networks for single image super-resolution* in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (2017), 136–144.