Improving YOLOv8 with Scattering Transform and Attention for Maritime Awareness

1st Borja Carrillo-Perez 2nd Angel Bueno Rodriguez 3rd Sarah Barnes 4th Maurice Stephan
German Aerospace Center (DLR), Institute for the Protection of Maritime Infrastructures, Bremerhaven, Germany
{Borja.CarrilloPerez, Angel.Bueno, Sarah.Barnes, Maurice.Stephan}@dlr.de

2023 International Symposium on Image and Signal Processing and Analysis (ISPA) | 979-8-3503-1536-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISPA58351.2023.10279352

***Abstract*—Ship recognition and georeferencing using monitoring cameras are crucial to many applications in maritime situational awareness. Although deep learning algorithms are available for ship recognition tasks, there is a need for innovative approaches that attain higher precision rates irrespective of ship sizes, types, or physical hardware limitations. Furthermore, their deployment in maritime environments requires embedded systems capable of image processing, with balanced accuracy, reduced latency and low energy consumption. To achieve that, we build upon the foundations of the standard YOLOv8 and present a novel architecture that improves the segmentation and georeferencing of ships in the context of maritime awareness using a real-world dataset (ShipSG). Our architecture synergizes global and local features in the image for improved ship segmentation and georeferencing. The 2D scattering-transform enhances the YOLOv8 backbone by extracting global structural features from the image. The addition of convolutional block attention module (CBAM) in the head allows focusing on relevant spatial and channel-wise regions. We achieve mAP of 75.46%, comparable to larger YOLOv8 models at a much faster inference speed, 59.3 milliseconds per image, when deployed on the NVIDIA Jetson Xavier AGX as target embedded system. We applied the modified network to georeference the segmented ship masks, with a georeferencing distance error of 18 meters, which implies comparable georeferencing performance to non-embedded approaches.**

***Index Terms*—Real-time instance segmentation, YOLOv8, scattering transform, attention, georeferencing, maritime awareness**

I. Introduction

Based on the critical role of maritime infrastructures in global trade, monitoring their security, integrity, and operational safety is paramount [1]. Research in maritime safety and security focuses on developing, testing, and validating innovative systems to assess the status of infrastructures for real-time protection and security counter-measures [2], [3].

Optical monitoring cameras enhance maritime situational awareness [4], but personnel may require support to track all pertinent details with multiple cameras, given the large volume of data generated [5]. Image processing on port monitoring video enhances real-time maritime situational awareness by enabling automatic ship detection, classification, and calculation of geographic position to be presented on a map. [6]. This allows maritime operators for faster assessment of the situation compared to the Automatic Identification System (AIS), which relies on intermittent ship transmissions every 2 to 10 seconds while underway. Additionally, it helps overcome the limitations of AIS, such as operational disruptions or cyber threats which may compromise its effectiveness [7]. Moreover, using cameras for ship monitoring provides cost-effectiveness, non-intrusiveness, and improved visual operational efficiency compared to radar-based systems [8].

Ship georeferencing using optical monitoring cameras is more accurately computed from segmented masks rather than surrounding bounding boxes which may include irrelevant background information [9]. Deep learning-based instance segmentation enables refined information extraction from the recognized ship, including the georeferencing of their position on a geographic coordinate system [9]. The development of image processing methodologies for maritime environments, requires the use of real-world ship monitoring datasets and precise algorithmic solutions. The practical application of general deep learning models in maritime domains requires effective approaches that can utilize training data and feature information regardless of the ship size or position within the image. To alleviate this, the use of the first-order 2D scattering transform, a wavelet-based mathematical operator, provides a hierarchical and sparse representation of the input data, thus being amenable as an enhancement for computer vision tasks [10], [11]. Additionally, attention models improve real-time instance segmentation by selectively focusing on informative regions, improving accuracy and efficiency of object recognition [12].

Employing a GPU-accelerated embedded system, equipped with a monitoring camera, enables deep-learning ship segmentation and georeferencing directly at maritime infrastructures [13]. Processing images locally on the deployed system offers reduced network bandwidth, minimized latency, cost efficiency, and improved security. The NVIDIA Jetson AGX Xavier[1] in particular, is a high-performance and energy-efficient embedded system for deep learning, with an energy consumption from 10 to 30 Watts. Its compact size, optimized neural network processing, and deployment flexibility make it ideal for real-time inference in resource-constrained environments. Supported by a comprehensive developer ecosystem, it simplifies computer vision applications deployment. Ships recognized and georeferenced within the embedded system can be seamlessly accessed via web services to the situational awareness system for their display on a map to the operator,

[1]https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/, accessed on 9 June 2023

enhancing real-time visualization and enriching overall maritime situational awareness [6].

Using a real maritime dataset, this work improves the current state-of-the-art in embedded real-time ship segmentation and georeferencing for maritime situational awareness. The contributions and results of this work are summarized as follows:

1) We modify the lightest YOLOv8 instance segmentation architecture [14], YOLOv8n, by adding:
   a) ScatBlock, a 2D scattering-transform-based block at the beginning of YOLOv8 backbone, replacing the first convolutional block.
   b) Convolutional Block Attention Module (CBAM) to the head.
2) The improved architecture achieves comparable mean Average Precision (mAP) as larger YOLOv8 models, at a faster inference speed, with ShipSG dataset [9].
3) The georeferencing of the segmented ship masks, is consistent with the state-of-the-art results on ShipSG.
4) We deploy the architecture in the NVIDIA Jetson Xavier AGX, acting as embedded system.

The remainder of this paper is organized as follows: Section II provides a literature overview of maritime datasets, embedded ship segmentation, and georeferencing. Section III describes YOLOv8 and our proposed architecture using scattering transform with attention on the ShipSG dataset in an embedded system. Section IV presents the implementation, training, and validation details with experimental results. Section V concludes the paper and outlooks future work.

II. Related works

A. Ship datasets for maritime monitoring

As deep learning algorithms for ship segmentation rely on supervised machine learning, it is necessary to use domain-specific training datasets. Real-world maritime monitoring requires image data with precise mask annotations for a broad range of ships and ship classes. The available, general-purpose, segmentation datasets such as COCO [16] or PASCAL VOC [17], therefore, do not suit the task of ship segmentation and georeferencing as benchmark datasets for maritime awareness. Several datasets in the literature for ship detection on video monitoring cameras are the Singapore Maritime Dataset [4], Seaships7000 [18], and a dataset introduced by Chen et al. [19]. However, these datasets lack a variety of ship classes in their annotations and do not provide ship masks, necessary for ship georeferencing. The MarSyn dataset [20] is a synthetic ship dataset that contains images rendered from synthetic 3D scenes for instance segmentation in six ship classes, without georeference from the ships annotated. A publicly available real-world dataset, for ship segmentation and georeferencing, is ShipSG [9]. This dataset includes seven classes of ships and two distinct views of a port location. The ShipSG dataset was collected to develop and evaluate instance segmentation and georeferencing methods for real-world maritime applications and therefore is used in this work.

B. Embedded ship segmentation and georeferencing

State-of-the-art methods for real-time instance segmentation with COCO [16], evaluated using mean Average Precision (mAP) and inference time in milliseconds, include RTMDet-Int-X with mAP 44.6% and 5.31 ms [21], YOLOv5x-seg with mAP 41.4% and 4.50 ms [22] and YOLOv8x-seg with mAP 43.4% and 4.02 ms [14]. We observe that YOLOv8 provides the fastest configuration at a high mAP, which is useful for maritime applications. For real-time ship segmentation, as seen in [9], the use of YOLACT [23] and Centermask-Lite [24] on the ShipSG dataset [9] are proposed. However, the use of embedded systems is not reported in these studies. Real-time embedded instance segmentation has been explored for its application in traffic videos [25], where the NVIDIA Jetson AGX Xavier as system is used to deploy a modified version of YOLACT [23] and SOLO [26]. Ship georeferencing from maritime monitoring video is proposed by [27] where a homography is created with reference pairs of latitude, longitudes, and pixel coordinates. The work of [9] analyses homography for georeferencing on the ShipSG dataset.

III. Methods

A. Overview of YOLOv8 for instance segmentation

YOLOv8 [14] is a state-of-the-art real-time model on COCO dataset [16], that builds upon previous YOLO versions. YOLOv8 supports a full range of vision tasks, including detection, instance segmentation, pose estimation, tracking, and classification, with instance segmentation being the task of interest for this work. The model utilizes a convolutional neural network with a modified version of the CSPDarknet53 [28] architecture as the backbone, which includes the novel C2f module that contains two convolutional blocks, a channel split and a CSP bottleneck [28] (see Fig. 1(f)). The backbone is followed by three segmentation heads, which learn to predict the segmentation masks for the input image (see Fig. 1(g)). YOLOv8 also offers customizable architecture and five model sizes, these being, from the lightest and fastest to the deepest and most accurate: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x.

B. Scattering-enhanced YOLOv8 with Convolutional Block Attention Module

We propose a lightweight architecture, depicted in Fig.1(a), based on YOLOv8n instance segmentation configuration [14], for its deployment in the NVIDIA Jetson AGX Xavier. The additions performed in our proposed architecture to modify YOLOv8n are the following:

1) 2D scattering transform block (ScatBlock): The 2D scattering transform is a specialized mathematical operator that extracts invariant feature representations by decomposing the input image data into a set of scattering coefficients. Each scattering coefficient is a translation-invariant feature map representation that captures local variations in an image [10]. We blend the 2D scattering transform in the backbone of YOLOv8 to enhance the input image for instance segmentation. To achieve this, we conceive a ScatBlock (see Fig. 1(b)) that

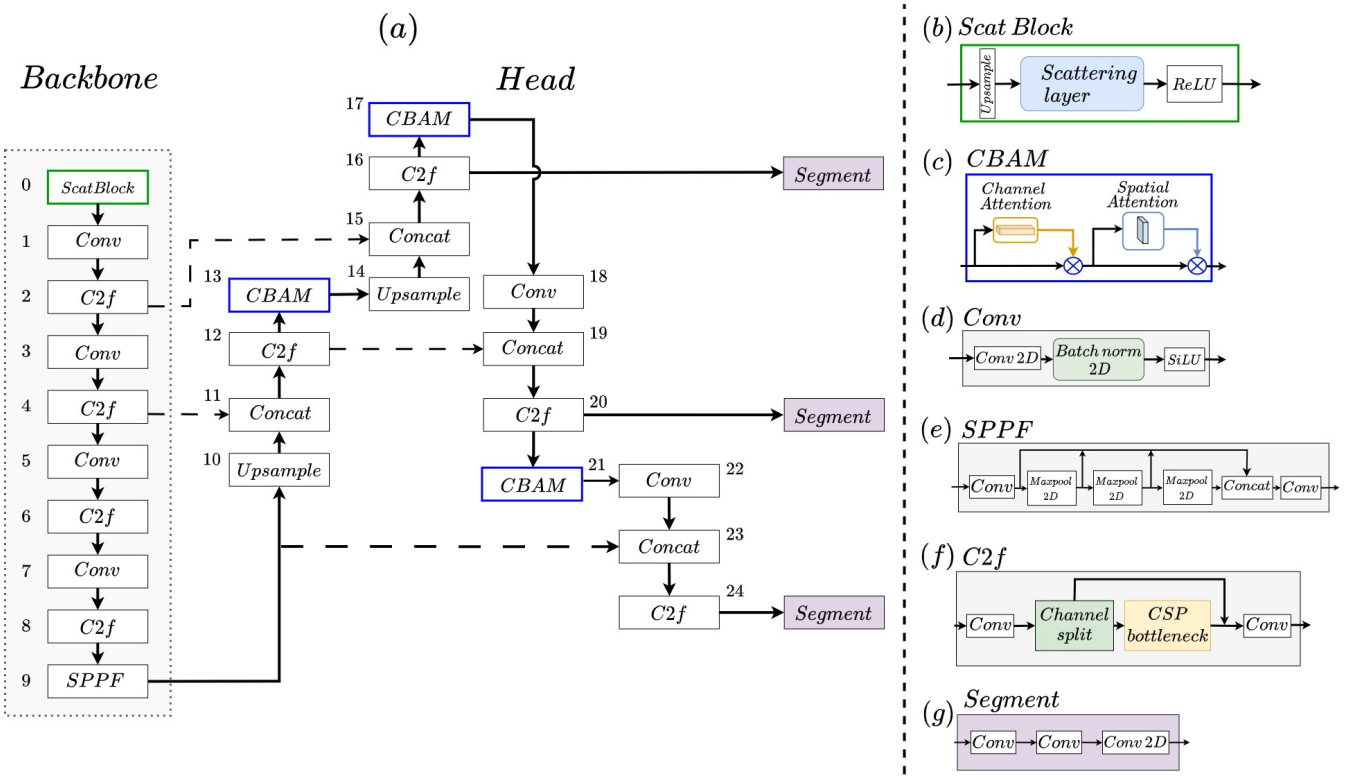

Fig. 1. Our proposed architecture with the ScatBlock in the backbone of YOLOv8n and CBAM modules in the head. (a) We place ScatBlock at the beginning of the YOLOv8n backbone to render a set of sparse feature maps, replacing the first Conv block of the original YOLOv8 backbone. The CBAM module is placed at the output of the head blocks of YOLOv8 to distill valuable object shape information for the segmentation task. The numbers next to every block represent the sequential order followed by the implementation, from input to output (b) The scattering block contains an upsample operation, followed by the scattering layer and a ReLU activation (c) CBAM module depicting the channel and spatial attention mechanisms [15] (d) Standard YOLOv8 convolutional block (e) Spatial Pyramid Pooling Fast module of YOLOv8 (f) C2f with split channel operation and a CSP Bottleneck. (g) Segment block of YOLOv8 that performs segmentation. In-depth description of SPPF, C2f and Segment modules is found in the original YOLOv8 [14].

contains the first-oder 2D scattering transform of the input image X. The ScatBlock computes the scattering transform from a set of dilated and rotated versions of a mother wavelet ψ, and a low-pass filter ϕ_J, with J being the spatial scale of the scattering transform, and L, the number of mother wavelet rotations. The computation of the scattering transform requires the convolution of the input image X with a set of pre-defined filter banks ψ_λ, which are the scaled and rotated version λ of the mother wavelet ψ. The convolved image is then subjected to an element-wise complex modulus operation. The resulting feature maps are smoothed using the low-pass filter ϕ_J, with a down-sampling factor of 2^J, to ensure invariance to small translations [10]. The scattering transform is similar to convolutional neural networks (CNNs) in that it involves an iterative process to compute multiple coefficients on the input data to extract hierarchical features [10]. In contrast, CNNs focus on spatially local features and are trainable models, while the 2D scattering transform explicitly captures global structural information and is a fixed transform. At layer ℓ, the wavelet filter-bank produces the representation $U_\lambda^{(\ell)}$ which corresponds to the application of each filter followed by application of taking the modulus, $|.|$, as:

$$U^{(\ell)}_{\lambda^{(\ell)},\lambda^{(\ell-1)},\ldots,\lambda^{(1)}} = |(\psi_\lambda \star U^{(\ell-1)}_{\lambda^{(\ell-1)},\ldots,\lambda^{(1)}})|, \quad (1)$$

with $U^{(0)}$, the initialization at the first scattering layer or 0th-order coefficients. The output of each layer ℓ is obtained with a smoothing operation using the low-pass filter as:

$$S^{(\ell)}_{\lambda^{(\ell)},\ldots,\lambda^{(1)}} = U^{(\ell)}_{\lambda^{(\ell)},\ldots,\lambda^{(1)}} \star \phi_J, \quad (2)$$

with $S^{(\ell)}_{\lambda^{(\ell)},\ldots,\lambda^{(1)}}$ the scattering representation at layer ℓ, of size $H \times W$, with $J \times L + 1$ channels representing each scattering order. These sparse feature maps are then forwarded to a $ReLU$ activation function. Our ScatBlock computes the first-order scattering coefficients. Albeit the computation of subsequent orders, i.e., second or third order, is achievable by incorporating more layers, only first-order coefficients hold significant energy for mainstream tasks [10]. The ScatBlock performs an upsampling operation to the input image X, to $(2\times H)\times(2\times W)$, to achieve $H\times W$ as the output resolution of the first-order coefficient maps before the computation of the scattering transform. With this, the ScatBlock maintains the image proportionality for the successive YOLOv8 backbone blocks.

2) Convolutional Block Attention Module (CBAM): The CBAM (Convolutional Block Attention Module), introduced by [15], enhances CNNs by incorporating attention mechanisms at a very low computational cost. The CBAM consists of two components (see Fig. 1(c)). The first component, the channel attention module, captures interdependencies among feature channels, that is, emphasizes informative channels while suppressing irrelevant ones. The second component, the spatial attention module, selectively highlights important spatial regions in the feature maps. By integrating both attention mechanisms, CBAM allows the network to focus on relevant spatial regions whilst emphasizing informative channels, thus leading to improved feature representation and better localization. The work of [15] incorporates the CBAM block into a broad range of deep learning models across diverse classification and detection datasets; with significant performance improvements observed. For instance segmentation tasks, CBAM successfully refined object boundaries and enhanced the accuracy of segmented individual objects within an image.

3) ScatYOLOv8n + CBAM: Our proposed architecture for real-time ship segmentation, as depicted in Fig.1(a), replaces the first convolutional block of the YOLOv8n backbone with a ScatBlock that uses the first-order 2D scattering transform. This block increases the number of channels in the feature map from 3 (RGB) to $3 \times (J \times L + 1)$, with $J = 1$ and $L = 6$. The ScatBlock is used in forward mode, since it contains fixed filters and therefore, no filter parameter update is backpropagated during training. In the second modification, inspired by [29], where YOLOv5 is used for aerial object detection, we incorporate the CBAM block after C2f blocks of the YOLOv8 head. The aim of incorporating the CBAM block is then twofold: to help the network to find regions of interest, that is, ships, and use those selected regions as input for the consecutive blocks in the head.

C. Dataset for maritime awareness and embedded device

The maritime dataset selected for this work is the ShipSG dataset [9], which is publicly available[2]. The ShipSG dataset consists of 3505 images and ship annotations of two different views of part of the port of Bremerhaven, Germany. The images were acquired during daylight hours with varying weather conditions. The annotations of ShipSG contain 11625 annotated ship instances, divided into seven ship classes (cargo, law enforcement, passenger/pleasure, special 1, special 2, tanker, and tug). Moreover, each image contains one geographic annotation of one of the ship masks (latitude and longitude) present in the image. An example image of ShipSG can be seen in Fig. 2(a).

The selected target system for deploying our proposed ship segmentation and georeferencing architecture is the NVIDIA Jetson AGX Xavier. The AGX Xavier is an embedded computing module with low power consumption (30 Watts) that contains: an octa-core ARM CPU, 512-core Volta GPU, and support for a range of deep learning frameworks, making it a versatile platform for developing and deploying vision-based systems.

[2]https://www.dlr.de/mi/shipsg, accessed on 9 June 2023

IV. Results

A. Network implementation

We use the ShipSG dataset [9] to evaluate our implemented model. Following the common practice in the field, we report mAP as performance metric, computed as the mean of all average precisions for Intersection over Union (IoU) from 0.5 to 0.95 with a step size of 0.05. We implement our proposed model with Pytorch 2.0, CUDA 11.7 and YOLOv8 Ultralytics v8.0.51. For the measurement of inference times, we use the NVIDIA Jetson AGX Xavier with JetPack 5.0 as the target embedded system for deployment.

We trained all models using a NVIDIA A100 GPU with random weight initialization for all models. The number of training epochs is 300, with the default settings provided by YOLOv8 [14]. The input size used for all models is 640×640 pixels. We implemented the ScatBlock (see block number 0 of Fig.1(a)) using the 2D Wavelet transformations by [30], with the Python module $pytorch\_wavelets$, which has CUDA support for GPU operations. The scattering layer we selected uses Symlet wavelets with 6 convolutional calls or filters (L) and 1 scale (J). Given that the resulting number of channels from an RGB image is $3 \times (J \times L + 1)$, the output number of channels of our ScatBlock is 21.

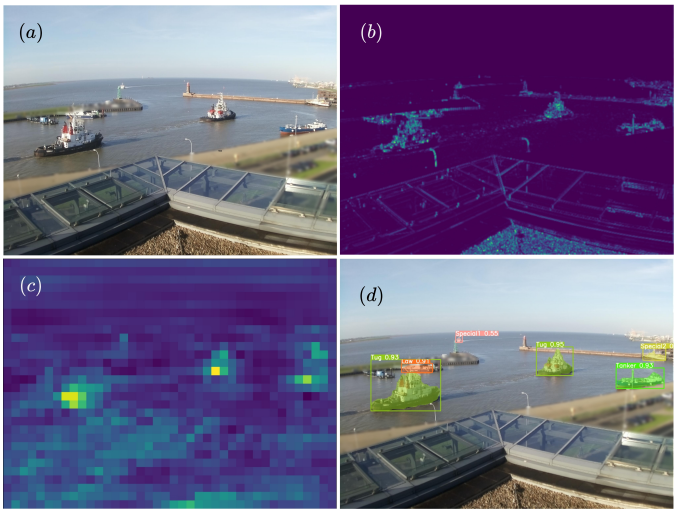

Fig. 2. Instance segmentation process of our proposed architecture on ShipSG. (a) ShipSG input sample image. (b) Output of the ScatBlock (here, for visualization, the mean of output channels). (c) Output of CBAM (module number 21 in Fig. 1). (d) ShipSG image with segmented and classified ships using our proposed architecture ScatYOLOv8n + CBAM.

An example of ship segmentation inference on ShipSG using our architecture is shown in Fig. 2. We observe that the ships present in the image are effectively enhanced by employing the 2D scattering transform (wavelets). This transformation improves the visual quality and perceptibility of the edges, resulting in clearer and more prominent delineation of the

present ships. As for the output of the CBAM, we observe that the attention mechanisms implemented (channel- and spatial-wise) learned to synergize with the scattering transform and leverage the location of the ship within the image whilst ignoring the background. The sensitivity of the rotated wavelet filters to the image information that is meaningful for ship segmentation, makes the ships prominent against the image background. CBAM takes each feature map and extracts local ship mask details to generate more refined attention maps in which the background appearance is mitigated.

TABLE I
COMPARISON OF STATE-OF-THE-ART REAL-TIME SEGMENTATION PERFORMANCES ON SHIPSG WITH YOLOV8N AND OUR ARCHITECTURE

Segmentation model	Input Size (pixel)	mAP (%)
YOLACT$_{700}$ [9]	700×700	58.20
Centermask-Lite$_{V39}$ [9]	800×600	64.40
YOLOv8n	640×640	70.15
ScatYOLOv8n + CBAM (this work)	**640×640**	**75.46**

Both YOLOv8n and our proposed ScatYOLOv8n + CBAM, as seen in Table I, improve significantly the mAP of previous approaches for ship segmentation on ShipSG. The baseline YOLOv8n shows 5.75% improvement compared to the Centermask-Lite implementation proposed by [9] and our ScatYOLOv8n + CBAM achieves a 11.06% improvement.

B. Ablation study

We evaluate the significance of the proposed additions in our architecture, namely ScatBlock and CBAM. To assess their impact, Table II presents individual performance results, including mAP and inference times on NVIDIA Jetson AGX Xavier. The first part of Table II shows the performance of each YOLOv8 model. The second part details the individual and combined additions of this work. The increments show the difference when compared with the YOLOv8n model. We list here the observed effects of the added modules:

- **Convolutional Block Attention Module (CBAM)**. The addition of attention to the end of the prediction heads produces an improved mAP for the network, with and without scattering, at a minimal increase in inference time. This proves that the use of CBAM is, in this case, worth the computational cost.
- **Scattering-enhanced backbone (ScatYOLOv8)**. The use of the ScatBlock at the beginning of the backbone, instead of the first $Conv$ block of CSPDarknet53 [28], produces a mAP improvement of 4.07% when compared to YOLOv8n at a small increase in inference time for the embedded system of 29.5 ms. Together with the CBAM modules on the head, the ScatBlock and CBAM leverage the performance of the network with a mAP improvement 5.11% and an increase in inference of 30.6 ms.

As shown in the ablation study, our proposed architecture ScatYOLOv8 + CBAM, provides a mAP comparable to the deeper and heavier YOLOv8l (75.46% vs 75.89%). Yet, our model has a much lower inference speed (59.3ms vs 127.1ms) on the NVIDIA Jetson AGX Xavier.

TABLE II
ABLATION STUDY OF YOLOV8 SEGMENTATION MODELS AND OUR PROPOSED IMPLEMENTATIONS AFTER TRAINING ON SHIPSG AND INFERENCE TIMES ON THE NVIDIA JETSON AGX XAVIER.

Segmentation model	mAP (%)		Inference (ms)	
YOLOv8n	70.35	-	28.7	-
YOLOv8s	71.99	(↑1.64)	32.2	(↑3.5)
YOLOv8m	74.84	(↑4.49)	72.4	(↑43.7)
YOLOv8l	75.89	(↑5.54)	127.1	(↑98.4)
YOLOv8x	76.45	(↑6.10)	196.6	(↑167.9)
ScatYOLOv8n	74.42	(↑4.07)	58.2	(↑29.5)
YOLOv8n + CBAM	70.75	(↑0.40)	29.9	(↑1.2)
ScatYOLOv8n + CBAM	**75.46**	**(↑5.11)**	**59.3**	**(↑30.6)**

C. Ship georeferencing for maritime situational awareness

We calculate the georeference of ships from ShipSG images using segmented masks from our proposed instance segmentation architecture. The georeferencing method [9] defines a homography matrix (H) to calculate the latitude and longitude of the segmented ship masks, as given in equation (3),

$$\begin{bmatrix} Latitude \\ Longitude \\ 1 \end{bmatrix} = H \begin{bmatrix} C_x \\ C_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} C_x \\ C_y \\ 1 \end{bmatrix} \quad (3)$$

The coefficients of H are calculated using the georeferencing annotations of ShipSG [9]. The variables C_x and C_y express the pixel coordinates of the ship from which the latitude and longitude are computed. For this, the lowest pixel on the mask in the vertical direction, representing the ship hull and water intersection, is used for georeferencing [9]. The Georeferencing Distance Error (GDE), measured in meters, assesses the performance by comparing true ShipSG annotations with the georeferences derived from the mask and homography. To compute the GDE, we use the haversine equation to compare the geolocation of the segmented ship and ground truth.

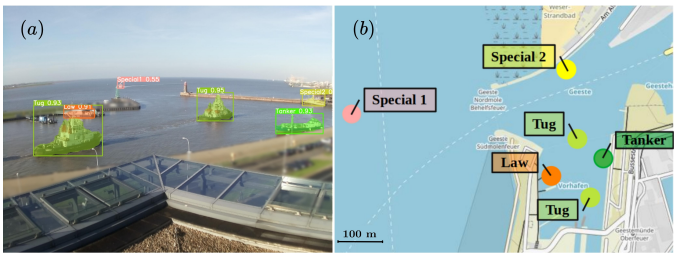

Fig. 3. The segmented ship masks are georeferenced as geographic map locations using a homography matrix to improve maritime awareness. (a) ShipSG image with segmented and classified ships using our ScatYOLOv8n + CBAM architecture. (b) Georeferenced ships displayed on OpenStreetMap [31] using the homography method by [9] on segmented masks.

Despite using an embedded system, our ScatYOLOv8n + CBAM model achieves 75.46% mAP and allows a GDE of 18±13 meters on ShipSG. These are consistent results for georeferencing at higher accuracy compared to [9], with a previously reported mAP 64.40% mAP and 22±10 meters. As seen in Fig. 3, when the georeferenced masks are accessed

using web services, our work can support real-time decision making, thus improving maritime awareness.

V. Conclusion

We presented an architecture, ScatYOLOv8n + CBAM, for ship segmentation and georeferencing that improves the state of the art in maritime awareness using a real-world ship segmentation and georeferencing dataset (ShipSG). Our architecture uses ScatBlock, a 2D scattering-transform-based block, to enhance the YOLOv8 backbone and attention (CBAM) in the head. We verified that our architecture, applied to the lightest YOLOv8, offers mAP of 75.46%, which is comparable to larger YOLOv8 models but offering a faster inference speed. The developed architecture is suitable for its use in an embedded system, such as the NVIDIA Jetson Xavier AGX, at an inference speed of 59.3 ms per image. We applied the modified network for georeferencing the segmented ship masks, attaining a georeferencing distance error of 18 ± 13 meters, thus leading to comparable georeferencing performance to non-embedded state-of-the-art approaches.

This work paves the way for new studies to design feature representations for light networks that synergize the scattering transform with attention modules to render refined feature maps, robust for computer vision applications, such as instance segmentation for maritime awareness. These models are paramount to embedded systems deployed in the field to lessen the computational burden of more sophisticated models often hosted in cloud computing servers.

References

[1] E. Engler, D. Göge, and S. Brusch, "Resiliencen–a multi-dimensional challenge for maritime infrastructures," *NAŠE MORE: znanstveni časopis za more i pomorstvo*, vol. 65, no. 2, pp. 123–129, 2018.

[2] K. Wang, M. Liang, Y. Li, J. Liu, and R. W. Liu, "Maritime traffic data visualization: A brief review," in *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*. IEEE, 2019, pp. 67–72.

[3] F. S. Torres, N. Kulev, B. Skobiej, M. Meyer, O. Eichhorn, and J. Schäfer-Frey, "Indicator-based safety and security assessment of offshore wind farms," in *2020 Resilience Week (RWS)*. IEEE, 2020, pp. 26–33.

[4] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 1993–2016, 2017.

[5] F. Li, C.-H. Chen, G. Xu, D. Chang, and L. P. Khoo, "Causal factors and symptoms of task-related human fatigue in vessel traffic service: A task-driven approach," *The Journal of Navigation*, vol. 73, no. 6, pp. 1340–1357, 2020.

[6] T. Flenker and J. Stoppe, "Marlin: An iot sensor network for improving maritime situational awareness," *MARESEC 2021*, 2021.

[7] M. Balduzzi, A. Pasta, and K. Wilhoit, "A security evaluation of ais automated identification system," in *Proceedings of the 30th annual computer security applications conference*, 2014, pp. 436–445.

[8] D. Bloisi, L. Iocchi, M. Fiorini, and G. Graziano, "Camera based target recognition for maritime awareness," in *2012 15th International Conference on Information Fusion*, 2012, pp. 1982–1987.

[9] B. Carrillo-Perez, S. Barnes, and M. Stephan, "Ship segmentation and georeferencing from static oblique view images," *Sensors*, vol. 22, no. 7, p. 2713, 2022.

[10] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[11] E. Oyallon, E. Belilovsky, S. Zagoruyko, and M. Valko, "Compressing the input for cnns with the first-order scattering transform," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 301–316.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[13] F. Sattler, B. Carrillo-Perez, S. Barnes, K. Stebner, M. Stephan, and G. Lux, "Embedded 3d reconstruction of dynamic objects in real time for maritime situational awareness pictures," *The Visual Computer*, pp. 1–14, 2023.

[14] G. Jocher, A. Chaurasia, and J. Qiu, "YOLOv8 by Ultralytics," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[15] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[16] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[17] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.

[18] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, "Seaships: A large-scale precisely annotated dataset for ship detection," *IEEE transactions on multimedia*, vol. 20, no. 10, pp. 2593–2604, 2018.

[19] X. Chen, L. Qi, Y. Yang, Q. Luo, O. Postolache, J. Tang, and H. Wu, "Video-based detection infrastructure enhancement for automated ship recognition and behavior analysis," *Journal of Advanced Transportation*, vol. 2020, 2020.

[20] M. Ribeiro, B. Damas, and A. Bernardino, "Real-time ship segmentation in maritime surveillance videos using automatically annotated synthetic datasets," *Sensors*, vol. 22, no. 21, p. 8090, 2022.

[21] C. Lyu, W. Zhang, H. Huang, Y. Zhou, Y. Wang, Y. Liu, S. Zhang, and K. Chen, "Rtmdet: An empirical study of designing real-time object detectors," *arXiv preprint arXiv:2212.07784*, 2022.

[22] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Z. Yifu, C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5: v7.0 - yolov5 sota realtime instance segmentation," Nov. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7347926

[23] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9157–9166.

[24] Y. Lee and J. Park, "Centermask: Real-time anchor-free instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 906–13 915.

[25] R. Panero Martinez, I. Schiopu, B. Cornelis, and A. Munteanu, "Real-time instance segmentation of traffic videos for embedded devices," *Sensors*, vol. 21, no. 1, p. 275, 2021.

[26] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "Solo: Segmenting objects by locations," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 2020, pp. 649–665.

[27] E. Solano-Carrillo, B. Carrillo-Perez, T. Flenker, Y. Steiniger, and J. Stoppe, "Detection and geovisualization of abnormal vessel behavior from video," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2193–2199.

[28] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[29] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 2778–2788.

[30] F. Cotter, "Uses of complex wavelets in deep convolutional neural networks," Ph.D. dissertation, University of Cambridge, 2020.

[31] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2017.