Ocean Engineering 286 (2023) 115647

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

A convolutional vision transformer for semantic segmentation of side-scan sonar data

Hayat Rajani *, Nuno Gracias, Rafael Garcia

Computer Vision and Robotics Research Institute (ViCOROB), University of Girona, Campus Montilivi, Edifici P4, Girona 17003, Catalonia, Spain

ARTICLE INFO

Dataset link: https://github.com/hayatrajani/s3seg-vit

Keywords:
Seafloor segmentation
Side-scan sonar
Vision transformer
Convolutional transformer
Real-time

ABSTRACT

Distinguishing among different marine benthic habitat characteristics is of key importance in a wide set of seabed operations ranging from installations of oil rigs to laying networks of cables and monitoring the impact of humans on marine ecosystems. The Side-Scan Sonar (SSS) is a widely used imaging sensor in this regard. It produces high-resolution seafloor maps by logging the intensities of sound waves reflected back from the seafloor. In this work, we leverage these acoustic intensity maps to produce pixel-wise categorization of different seafloor types. We propose a novel architecture adapted from the Vision Transformer (ViT) in an encoder–decoder framework. Further, in doing so, the applicability of ViTs is evaluated on smaller datasets. To overcome the lack of CNN-like inductive biases, thereby making ViTs more conducive to applications in low data regimes, we propose a novel feature extraction module to replace the Multi-layer Perceptron (MLP) block within transformer layers and a novel module to extract multiscale patch embeddings. A lightweight decoder is also proposed to complement this design in order to further enhance multiscale feature extraction. With the modified architecture, we achieve state-of-the-art results and also meet real-time computational requirements. We make our code available at https://github.com/hayatrajani/s3seg-vit.

1. Introduction

High-resolution acoustic maps of the seafloor are a central tool for a wide variety of operations underwater. Application scenarios include activities as varied as environmental monitoring, marine archaeology, geology surveying, structure inspection, security, search and rescue, and others. If these maps include topographical features and bottom types correctly identified and classified, they would become fundamental for the scientific, naval and economic exploration of the oceans. An aftermath of the surveys for creating such maps, however, typically involves a tedious and labour-intensive process of manually identifying and labelling those features, which is expensive in terms of time and cost. Automating this task of pixel-wise classification, in real-time, while the surveys are being conducted would not only alleviate this burden but also open new avenues for Autonomous Underwater Vehicles (AUVs) to help automate the process of mission planning and navigation, where real-time processing of data is all the more crucial.

Although the use of optical sensors for conducting such surveys has seen several advances, they are severely affected by light attenuation and colour shift caused by the variability in water conditions. This imposes limitations on the range of operation, restricting samples to be acquired only over small areas. The use of acoustic sensors, on the other hand, makes it possible to perceive the underwater environment even in zero-visibility conditions regardless of the depth, also allowing them to cover a much wider area in a single pass. The Side Scan Sonar (SSS) is one such acoustic sensor that is very widely used in marine surveys. It is very easily adaptable to numerous types of sea vessels without the need for specific configurations and consumes low power, making it very economical and easy to deploy. As such, this work is aimed at semantic segmentation of acoustic images acquired from a SSS.

Much of the prior work carried out in this area makes use of traditional image processing and pattern recognition approaches such as clustering strategies (Celik and Tjahjadi, 2011; Yao et al., 2000), Markov Random Field (MRF) (Mignotte et al., 1999, 2000) or active contouring (Lianantonakis and Petillot, 2007). These methods are based on hand-crafted features and either lack the efficiency to be used in real-time or the capacity for generalization, among other issues.

In this paper, we propose to leverage the capabilities that Deep Neural Networks (DNNs) have showcased in pixel-wise labelling in recent years. In particular, we adopt Vision Transformers (ViTs) for the above-mentioned task due to their ability to draw long-range associations among different regions of an image. The motivation behind using long-range associations comes from the fact that expert geophysicists often use global context to disambiguate among similarly looking classes. We

* Corresponding author.
E-mail addresses: hayat.rajani@udg.edu (H. Rajani), ngracias@silver.udg.edu (N. Gracias), rafael.garcia@udg.edu (R. Garcia).

https://doi.org/10.1016/j.oceaneng.2023.115647
Received 15 December 2022; Received in revised form 26 July 2023; Accepted 17 August 2023
Available online 31 August 2023

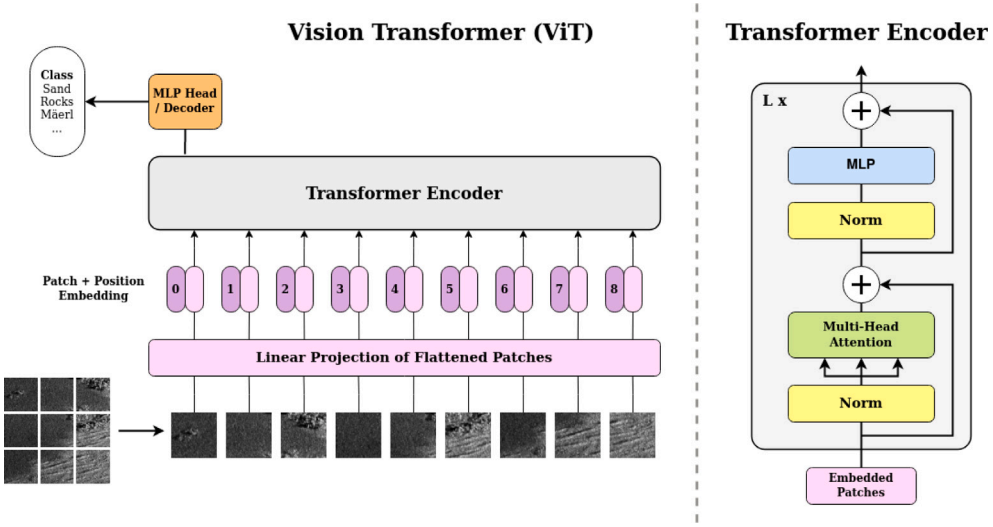

Fig. 1. The vanilla ViT architecture. *Source:* Adapted from Dosovitskiy et al. (2021).

believe that ViTs would enable the model to efficiently capture enough global context so as to make more informed decisions. This study, thus, also serves as a proof-of-concept of the feasibility of ViTs for tasks such as SSS segmentation. Given sufficient speed of computation, we can then use larger images (for instance, 512×512) to capture more global information and further improve results.

The Transformer was originally developed as a sequence transduction model, for tasks such as machine translation, where it is essential for the model to formulate a thorough understanding of the language by capturing how the different components in a text interact with each other, the different semantics they might adopt in different contexts and the various syntactical patterns that might arise. Vaswani et al. (2017) designed a computationally efficient mechanism, called the multi-head self-attention, to capture such global dependencies, which eventually led Transformer-based models to become the state-of-the-art in many Natural Language Processing (NLP) tasks. Later, Dosovitskiy et al. (2021) transferred these principles to computer vision, giving rise to an architectural paradigm called the Vision Transformer, as depicted in Fig. 1.

Since the original Transformer was designed to operate on a sequence of 1D word embeddings, as opposed to the 2D or 3D images that vision models usually deal with, ViT breaks down the input image into a sequence of patches and linearly projects the flattened patches into an embedding space. This is essentially what allows the multi-head self-attention mechanism to ingest the n-D input and draw associations between different regions of the image, thereby capturing global context. However, in doing so, the spatial correlations between patches are lost and the structure of the image is no longer preserved. This requires the use of positional encodings as a way to embed this structural information within the architecture. Unfortunately, positional encodings must be learnt from scratch as the model has no knowledge about the relative location of the patches to begin with. This increases the dependence of ViTs on large datasets, resulting in poor generalization otherwise. However, for domains such as marine robotics, where the data is typically very scarce, employing such architectures becomes infeasible, especially without any pre-training.

Convolutional Neural Networks (CNNs), on the other hand, have the grid-structure of the image built into their architectural design. This acts as a strong prior for the model resulting in properties such as shift invariance and equivariance. Moreover, although ViTs possess a global receptive field, convolutions tend to be more effective in extracting local features. We, therefore, propose two architectural modifications in an attempt to inject these characteristics into ViTs and to make them more suitable for applications that lack sufficiently large datasets.

- We replace the linear patch embeddings at the beginning of each transformer stage with multiscale patch embeddings inspired in part by the design choices of Inception-v2 (Szegedy et al., 2016).
- We replace the non-linear projections of the Multi-layer Perceptron (MLP) block within each transformer layer with a feature extraction block inspired by Ghost convolutions (Han et al., 2020).

The benefits are multifold. Not only does the use of convolutions within these modules introduce the notion of a grid-like structure, but posing them as stacked separable convolutions also reduces the number of parameters that the model needs to optimize. This further relaxes the need for position encodings while also enabling multiscale feature extraction. Consequently, this improves the capabilities of our modified architecture in capturing high-frequency details and generalizing well in the absence of huge datasets for training.

Furthermore, (hierarchical) ViT-based encoders, due to their ability to draw associations between different regions of an image, especially at multiple scales, tend to produce strong latent representations, which are readily suitable for the task of semantic segmentation. This permits the use of simple decoder designs without the need for computationally expensive modules. We model our decoder after the lightweight design proposed by Xie et al. (2021), by supplementing it with auxiliary output blocks inspired by DeepLab's Atrous Spatial Pyramid Pooling (ASPP) (Chen et al., 2018).

With our modified design, we surpass the results of previous state-of-the-art by a margin of over 3% in mIoU for our smallest model and over 10% for our largest model while also meeting the computational considerations for real-time implementation. Consequently, we demonstrate the applicability of ViTs for tasks such as semantic segmentation of the seafloor using SSS waterfalls, for which large datasets are seldom available. We believe that such hybrid ViT-based architectures have a large potential in underwater applications. To encourage further exploration of these approaches and to ease the reproducibility of our results, our code will be made available online at https://github.com/hayatrajani/s3seg-vit.

The remainder of this paper is organized as follows. Section 2 presents a brief review of previous works on SSS segmentation using DNNs and related literature in the context of semantic segmentation using ViTs. Then, Section 3 provides thorough details of our proposed architecture. Next, Section 4 presents an overview of the dataset and the experimental setup. Section 5 reports and visualizes our results. And finally, Section 6 concludes this study by outlining the planned efforts.

2. Related work

2.1. CNNs for SSS segmentation

The approach in this paper is, to the best of our knowledge, the first to demonstrate the applicability of ViTs to SSS segmentation. Nonetheless, a number of previous studies have adopted CNNs and attention-based mechanisms for this task. Wang et al. (2019) propose a U-Net (Ronneberger et al., 2015) like encoder–decoder architecture for real-time SSS segmentation. They employ a two-way branching structure exploiting depth-wise separable convolutions in their encoder for efficiency and a combination of pooling indices and direct skip connections to feed the lost spatial information back into the decoder. Wu et al. (2019), on the other hand, focus on dealing with different sources of noise in SSS imagery and the issue of class imbalance. They employ residual blocks in their encoder and propose the use of side-output blocks, in addition to the typical encoder–decoder design, to leverage multi-level information from each encoder. Similarly, Zhao et al. (2021) focus on dealing with different sources of noise in SSS imagery for real-time segmentation. Apart from carefully designing the encoder and decoder modules, they propose a novel DCblock employing dilated convolutions that sits between the encoder and the decoder to attain more context. Whereas, Burguera and Bonin-Font (2020) target the problem of building a semantic map of the seafloor, specifically to search for loop candidates in a SLAM context. They propose an end to end framework in this regard while employing a lightweight encoder–decoder architecture for online multi-class SSS segmentation. However, Wang et al. (2022b) argue that such simple encoder–decoder designs are vulnerable to noise interference and only work well for SSS images with simple backgrounds. They propose an adaptive receptive field mechanism on the skip connections between the encoder and decoder branches to improve target shape fit. They further supplement the encoder branch with dynamic multiscale dilated convolution blocks to extract multiscale target features, and supplement the decoder branch with attention-based feature fusion blocks to better fuse global and local features while suppressing background noise. Furthermore, they propose a tree structure optimization module to refine the produced segmentation masks, thus, reducing the rate of misclassifications. Wang et al. (2022b) also propose a boundary loss based on structural similarity and weighted binary cross-entropy to improve classification along the contour of the target. However, we believe these enhancements to be specifically directed towards the problem of target segmentation, which is the main purpose of their work, as opposed to our objective of seafloor segmentation. Moreover, from the reported results, the architecture seems to have a very large number of parameters, and consequently a much lower inference speed. Therefore, we do not draw direct comparisons with this approach. Yu et al. (2021), on the contrary, propose a novel approach for SSS segmentation by employing recurrent residual convolutions to capture global context followed by a self-guidance block for further refinement of results. The self-guidance block takes inspiration from the discriminator component of Generative Adversarial Networks (GANs) and serves to distinguish the generated segmentation mask from the ground truth. Although the authors claim a boost in segmentation results with their approach, the inference speed is significantly slow. Moreover, they mainly draw comparisons with models such as the U-Net (Ronneberger et al., 2015) and SegNet (Badrinarayanan et al., 2017), which are over 15 times larger than our proposed architecture. We, therefore, do not include this approach in our comparative study either. Yu et al. (2022), take this approach further in another publication directed towards target segmentation. They propose a dual-branch framework comprising a segmentation branch and a refinement branch. The segmentation branch adopts a MobileNetv2 (Sandler et al., 2018) backbone, additionally consisting of local attention and recurrent residual modules to dampen the effect of irrelevant features, thereby facilitating better emphasis on the target. This also addresses the overfitting caused by unbalanced datasets. The refinement branch further tunes this output with the help of holistic attention blocks for multi-level feature fusion. Both branches are further complemented by ASPP modules (Chen et al., 2018) for enlarged receptive fields and better contextual understanding. However, this results in an architecture that is parameter-heavy, being almost 6 times larger than ours. Further, since this work is also intended for target segmentation, as opposed to our objective of seafloor segmentation, we do not draw direct comparisons with it.

2.2. ViTs for semantic segmentation

The vanilla ViT, as proposed by Dosovitskiy et al. (2021), yields low resolution feature maps of uniform scale, making it less desirable for dense prediction tasks such as semantic segmentation. To address this, Wang et al. (2021) proposed as an alternative a hierarchical structure composed of a progressively shrinking pyramid capable of extracting features at multiple scales. This readily allowed ViTs to be plugged into standard dense prediction frameworks. They further apply spatial reduction to *key* and *value* embeddings before attention computation to handle the quadratic complexity associated with the traditional self-attention operation. Based on this hierarchical design, Ren et al. (2022) and Yao et al. (2022) propose different flavours of spatially-reduced attention to preserve image details. Where the former downsamples the *key* and *value* embeddings with different rates for different attention heads, the latter employs discrete wavelet transforms. Liu et al. (2021), on the other hand, proposed a window-based self-attention scheme to reduce memory and computational costs. Attention is computed among tokens within non-overlapping local windows and the windows are shifted by a certain amount between consecutive layers to facilitate cross-window connections. This approach to self-attention computation was followed up by several other works (Huang et al., 2021; Wang et al., 2022a; Dong et al., 2022; Wu et al., 2022) proposing different schemes of windowing and cross-flow of information among windows. However, these approaches are still restricted by the number of input tokens. To increase efficiency in processing high-resolution images, Ali et al. (2021) propose cross-covariance attention which computes self-attention among feature channels thereby making the computation linear in the number of tokens. Koohpayegani and Pirsiavash (2022) take this a step forward by replacing the softmax in self-attention with L1-normalization of *key* and *value* embeddings to further boost computational efficiency. In our proposed architecture, we use an adaptation of this approach, and draw comparisons with notable window-based and spatially-reduced self-attention mechanisms.

A related but independent line of work addresses the lack of CNN-like inductive biases in ViTs in order to improve their efficiency in capturing high-frequency details. There have been several studies (D'Ascoli et al., 2021; Srinivas et al., 2021; Wu et al., 2021; Guo et al., 2022; Li et al., 2022; Ma et al., 2022; Mehta and Rastegari, 2022; Si et al., 2022; Zhang et al., 2022) in this regard, which adopt one or a combination of diverse strategies such as introducing convolutions within the transformer blocks of ViTs, modifying the self-attention computation using convolutions, replacing the MLP blocks of ViTs with convolutions, or introducing transformer blocks within CNNs. The approach by Guo et al. (2022) is the closest to ours in that they also use a convolutional stem and adapt the MLP block of ViTs by introducing convolutions, apart from adopting convolutions inside their self-attention mechanism. However, in addition to inducing CNN-like inductive biases in the architecture, one of our primary objectives is to enhance the representability of objects of different scales, driving us to adopt distinct designs.

While the aforementioned studies focus on modifying the design of the ViT-based encoder counterpart of the framework, there have been a handful of efforts directed towards an efficient decoder design. For instance, Cao et al. (2021), taking inspiration from U-Net (Ronneberger et al., 2015), propose a Swin Transformer (Liu et al., 2021) based symmetric upsampling decoder, and Strudel et al. (2021) propose as

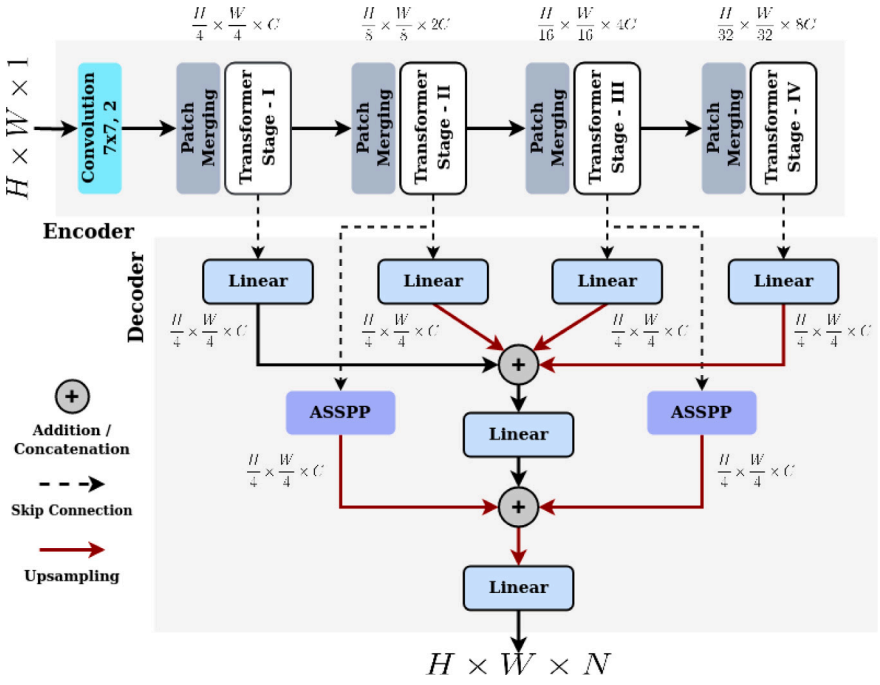

Fig. 2. Overview of the proposed architecture. H' and W' refer to the spatial dimensions of the input. C denotes the initial embedding dimension. N denotes the number of classes.

a decoder a mask transformer that jointly processes patch and class embeddings. Bousselham et al. (2021), on the other hand, propose an end-to-end trainable self-ensembling approach to leverage multi-scale features that are produced by different stages of the encoder, without the need of expensive feature fusion operations. However, Xie et al. (2021) propose a much simpler design, solely consisting of feature aggregating MLP blocks, that is capable of producing powerful representations while being computationally inexpensive. We model our decoder after an adaptation of this approach as detailed in Section 3.2.

3. Proposed architecture

Fig. 2 depicts an overview of the proposed architecture. It follows the general scheme of an encoder–decoder structure for semantic segmentation consisting of downsampling encoder blocks and upsampling decoder blocks with skip connections from the corresponding encoders. However, instead of building the decoder as a symmetric counterpart of the encoder, a much lighter-weight approach is adopted that is not only computationally more efficient but also yields better segmentation results, as discussed later in Section 5. The following two subsections describe the encoder and decoder modules in further detail.

3.1. A multiscale convolutional ViT encoder

The encoder adopts a modified version of a conventional 4-stage hierarchical ViT that begins with an initial patch size of 4×4 pixels, projected onto a C-dimensional embedding space. We set the initial length of embeddings to 24. For an input of size $H \times W$ pixels, this results in a sequence of $(H \cdot W) \,/\, 16$ 24-dimensional patch embeddings. Between two successive transformer stages, patches in non-overlapping 2×2 neighbourhoods are merged together while the length of their embeddings is doubled. This essentially downsamples each patch by a factor of 2 and, consequently, reduces the sequence length by a factor of 4. As a result, each transformer stage operates on a different scale, thereby making it possible for the encoder to construct a feature pyramid comparable to that of traditional CNN backbones.

Traditionally, the patch embedding and merging modules are composed of reshaping, flattening and linear projection operations. Our approach, on the other hand, leverages the local feature extraction capabilities of convolutions. We begin by applying a 7×7 convolution with a stride of 2 to the input image, resulting in a feature representation of size $\frac{H}{2} \times \frac{W}{2} \times 12$. Before each transformer stage, we then place a patch merging module that downsamples the input by a factor of 2 and doubles the number of feature channels. The produced feature representation is subsequently flattened along the spatial dimensions to make it suitable to be processed by the corresponding transformer stage. Thus, each transformer stage, $i \in \{1, 2, 3, 4\}$, operates on an input of size $\frac{H \cdot W}{2^{i+1}} \times 12 \cdot 2^i$.

The patch merging module, as illustrated in Fig. 3, was in part inspired by the design of Inception-v2 (Szegedy et al., 2016). It consists of four parallel branches of stacked depthwise convolutions of different receptive fields with appropriate padding to maintain spatial dimensions. Apart from introducing convolutional priors, the main rationale behind the patch merging module was to be able to adequately represent objects of different scales such as small pebbles to large boulders. The parallel convolutional branches enable feature extraction over various spatial footprints to generate multi-scale patch embeddings. Such a design further complements self-attention to draw finer global associations. Further, employing depthwise convolutions instead of full convolutions and factorizing convolutions with larger kernels by a stack of 3×3 convolutions, significantly saves on parameters while maintaining the effective receptive field. Each depthwise convolution operation is also followed by group normalization (Wu and He, 2018) with the number of groups set to 1. We then apply average pooling, preceded by a Hard Swish non-linearity (Howard et al., 2019), to downsample the generated feature representations by a factor of 2 followed by a pointwise convolution to project the aggregated multi-scale representations to twice the length of the input embeddings. Moreover, we introduce a residual connection, composed of a pointwise convolution and average pooling, for stability. The overall parameter count still remains lower as compared to employing a single full 3×3 convolution with a stride of 2 for patch merging, as proposed by Wu et al. (2021) and Dong et al. (2022), in lieu of linear projections.

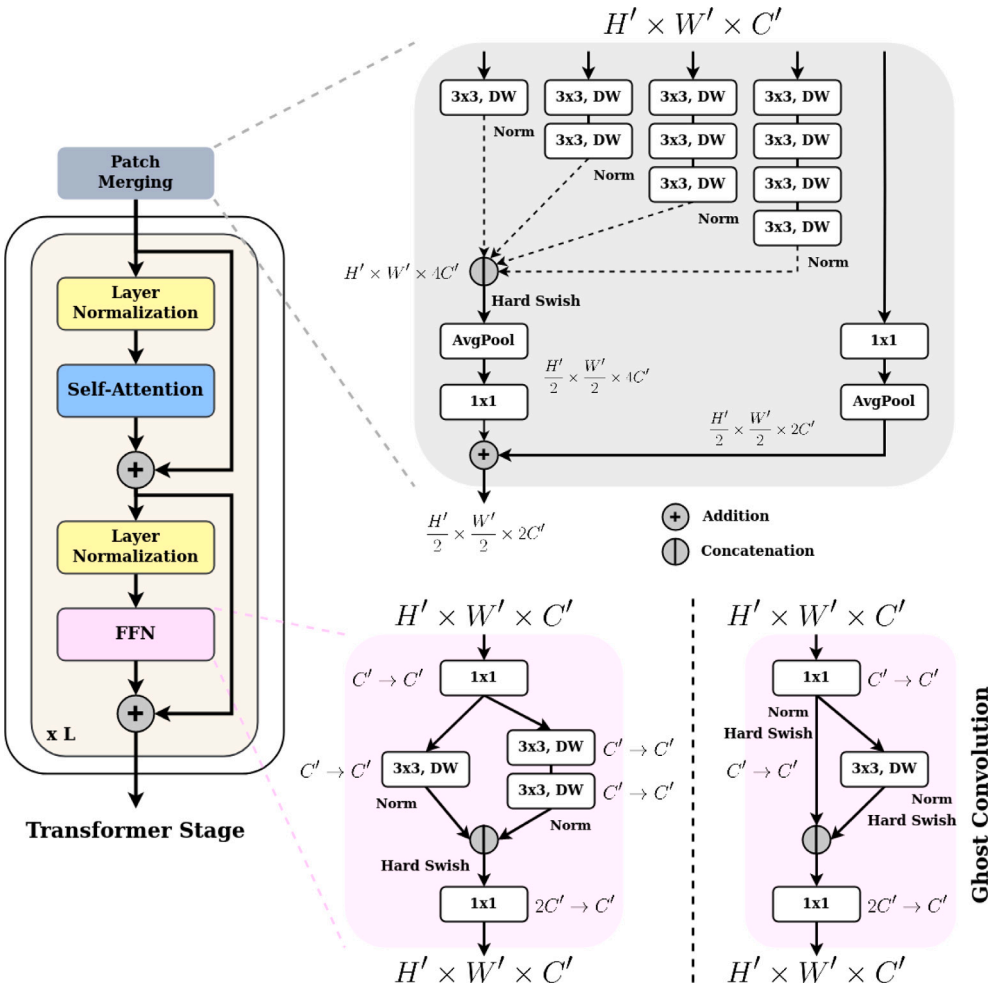

Fig. 3. Overview of a transformer stage together with illustrations of the patch merging module, the proposed FFN and ghost convolutions. H' and W' refer to the spatial dimensions of the input. C' denotes the number of input channels. DW denotes a depthwise convolution.

Fig. 3 also illustrates the transformer stage in more detail. Each transformer stage consists of $L \in \{3, 6, 12, 3\}$ transformer layers. Each transformer layer, as in the original ViT design (Dosovitskiy et al., 2021), is composed of a self-attention module and an MLP block, each preceded by layer normalization (Ba et al., 2016) and accompanied by a residual connection (He et al., 2016). Traditionally, the MLP block comprises two linear projections separated by a GELU non-linearity. Since naïvely replacing the linear projections by 3×3 convolutions increases the overall parameter count 9-fold, we instead adopted ghost convolutions as conceived by Han et al. (2020). They try to replicate the redundancy in feature maps generated by full convolutions through cheap linear operations. Specifically, a full convolution is split into two parts: a pointwise convolution in order to generate the primary feature maps followed by a 3×3 depthwise convolution (portrayed as a cheap linear operation) in order to generate secondary "ghost" features that add redundancy. We later extended this design by applying an additional 5×5 depthwise convolution, implemented as a stack of two 3×3 depthwise convolutions, to the primary features before concatenating them with the ghost features. We use this extended ghost convolution to replace the first linear projection in the original MLP. Again, after each convolution operation, we employ group normalization with a group size of 1. We also employ a Hard Swish non-linearity before the final linear projection. The resultant module not only introduces image-specific priors into the design but also reduces the parameter count when compared to a corresponding MLP block with an expansion ratio of 2.

Finally, due to the linear complexity in the number of patches, we use the attention mechanism as proposed by Ali et al. (2021). To further reduce computational cost, we remove the expensive softmax-based normalization of the attention matrix and, instead, L1-normalize the *key* and *query* embeddings before computing the attention scores as proposed by Koohpayegani and Pirsiavash (2022). To avoid ambiguity in comparison, we term this modified attention mechanism ***SimXCA***, a contraction of the names of the two referenced approaches. We set the number of attention heads to $\{2, 4, 8, 16\}$ for the four transformer stages respectively, and we do not use any kind of positional encoding.

3.2. A SegFormer-ASSPP decoder

Classical CNN-based approaches to semantic segmentation, such as U-Net (Ronneberger et al., 2015) and SegNet (Badrinarayanan et al., 2017), design the decoder as a symmetric counterpart to the encoder together with skip connections in order to add the lost spatial information back from the multi-resolution feature pyramid of the encoder. However, we observed that adopting such a symmetric design for ViT-based encoder–decoder frameworks yields quite poor results. This can be particularly attributed to the dependence of ViTs on large datasets to overcome their inherent lack of CNN-like inductive biases, which is further amplified by the symmetric decoder. Despite our modifications to induce CNN-like inductive biases in the design, we postulate that such complex architectures are not really necessary for the decoder.

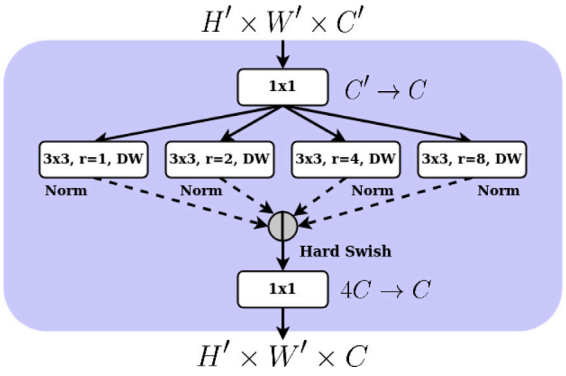

Fig. 4. Overview of the modified ASPP module. H' and W' refer to the spatial dimensions of the input. C' denotes the number of input channels. C denotes the initial embedding dimension. DW denotes a depthwise convolution.

The encoder, due to its ability to draw associations between different regions of the input image and extract multi-scale local features, already tends to produce strong latent representations that are readily suitable for the task of semantic segmentation. This permits the use of simpler designs for the decoder, which also drastically reduces the number of trainable parameters. As such, we base our decoder after the design proposed by Xie et al. (2021).

First, feature representations from each encoder stage are linearly projected to the initial embedding dimension, C, and are bilinearly upsampled to match the size of the initial patch embedding, $\frac{H}{4} \times \frac{W}{4}$. The projected representations are then fused together by addition, instead of concatenation as proposed by Xie et al. (2021). This is followed by another linear projection to the initial embedding dimension. This fused representation then undergoes a final linear projection to an N-dimensional space, where N is the number of classes, in order to predict the segmentation mask, which is subsequently upsampled by a factor of 4 to match the input image resolution.

However, we postulate that such a simple feature aggregating decoder is unable to completely leverage the multi-scale representations generated by the encoder, especially for small-scale objects. Therefore, contrary to Xie et al. (2021), we also employ two auxiliary blocks, based on the ASPP module (Chen et al., 2018), that operate on the feature representations from the second and the third encoder stages. The ASPP module can effectively enlarge the receptive field and incorporate multi-scale context, which also compensates for the lack of explicit modelling of global associations in the decoder, thereby further reducing the rate of misclassification. Since the bulk of the transformer blocks lie within the third stage of the encoder, the so produced feature representations are rich-enough for ASPP-decoding. Also, the spatial resolution is adequately high to not suffer significantly from the loss of information caused by downsampling. To further sharpen the results, we also adopt the feature representations from the second encoder stage for ASPP-decoding due to the relatively higher spatial resolution.

Our modified version of the ASPP module is illustrated in Fig. 4. First, we project the input feature representations onto the initial embedding dimension, C. The projected representation is then passed through the four parallel atrous convolution branches with dilation rates $r \in \{1, 2, 4, 8\}$, and a kernel size of 3. Each atrous convolution is implemented as a depthwise convolution and is followed by group normalization (Wu and He, 2018) with the number of groups set to 1. The feature representations from each atrous branch are then concatenated together and projected to the initial embedding dimension. This projection is preceded by a Hard Swish non-linearity (Howard et al., 2019). Finally, the aggregated representation from the two auxiliary blocks is upsampled to match the size of the initial patch embeddings. These upsampled representations are then concatenated with the previously fused representation from all four encoder stages, before predicting the final segmentation mask.

4. Experimental setup

4.1. Dataset

The datasets used in the course of this work were acquired with a Klein 3000 Side Scan Sonar during various surveys in the Balearic Sea. Approximately 52 km of coastal area was surveyed at an altitude varying from 4 to 21 m. Four categories of sediments were identified, namely Sand Ripples, Rocks, Maerl and Fine Sediments (such as silt and mud) covering approximately 50.60 %, 13.90 %, 12.06 % and 23.44 % of the total area respectively.

The raw SSS waterfalls were recorded in the eXtended Triton Format (XTF) and subsequently processed using SonarWiz for mosaicing. The mosaiced SSS waterfalls were then georeferenced and annotated by two expert geophysicists using ArcGIS. Fig. 5 depicts an example of a portion of the SSS mosaic and the corresponding annotation. We further processed the raw SSS waterfalls for blind zone removal and slat range correction. The available navigation data was then used to geocode each bin of the waterfall. This allowed us to fetch the corresponding annotations from the ArcGIS interpretations and automatically generate the ground truth for the SSS waterfalls. However, since the annotations were fetched from SSS mosaics while the ground truth was being generated for SSS waterfalls, misalignments in the mosaic may result in slight pixel-wise errors in the ground truth. Moreover, the annotations were made on a much coarser resolution than the SSS waterfalls and also suffered from human error resulting in ambiguous inter-class boundaries, missing labels and skewed or misaligned annotations for certain areas. However, despite the noisy ground truth, our model is able to generalize quite well, as discussed in Section 5.

The waterfalls and the corresponding ground truth were then partitioned in batches of 256 lines to generate images of size 256 × 256 with a 128 pixel-overlap along-track and across-track. This resulted in a total of 47,420 images, divided by an 80–20% split to form the training and validation sets respectively. Further, another set of images, equivalent to about 5% of the training set, was generated from a separate non-overlapping transect with similar class distribution as the training set to form a test set of 1800 additional images. The noisy ground truth of these test images was then manually corrected so as to be able to produce accurate metrics for evaluation. Fig. 6 illustrates some examples of the images and the different seabed types contained in the dataset.

4.2. Training and evaluation

All our models were trained on an NVIDIA A100 Tensor Core GPU for 100 epochs with a batch size of 64. We utilized the AdamW optimizer with a weight decay of $1e^{-2}$ and learning rate of $6e^{-5}$, decayed using a polynomial learning rate scheduler with a warm-up of 3 epochs. Weighted Cross Entropy was set as the loss function in order to tackle class imbalance. We also adopted standard train-time data augmentation techniques such as random rotation, random resized crop, random horizontal and vertical flip, random variations in contrast and/or sharpening and Gaussian blur. The models were implemented in PyTorch 1.11.0 and Python 3.8.10. The source code with all hyper-parameter configurations and pre-trained models will be made available at https://github.com/hayatrajani/s3seg-vit.

All trained models were then evaluated on a standard laptop equipped with an NVIDIA GeForce GTX 1650 Mobile GPU and an Intel Core i5-9300H CPU operating at 2.40 GHz running Ubuntu 20.04.5, Python 3.8.10 and PyTorch 1.9.0 + cu111. We report model performance in terms of mean Intersection over Union (mIoU) and inference speed in number of images processed per second (FPS).

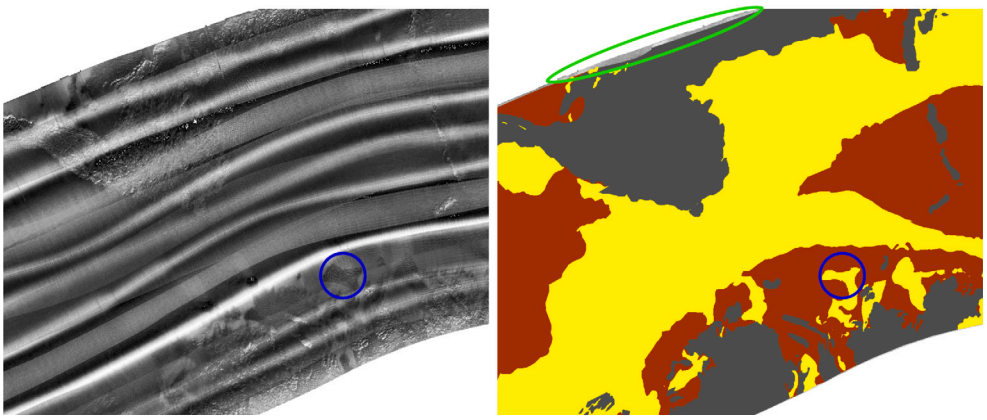

Fig. 5. SSS mosaic (left) and the corresponding interpretation (right). Note that in some areas the annotations are not accurate (as marked in blue) and may also be missing for certain areas (as marked in green).

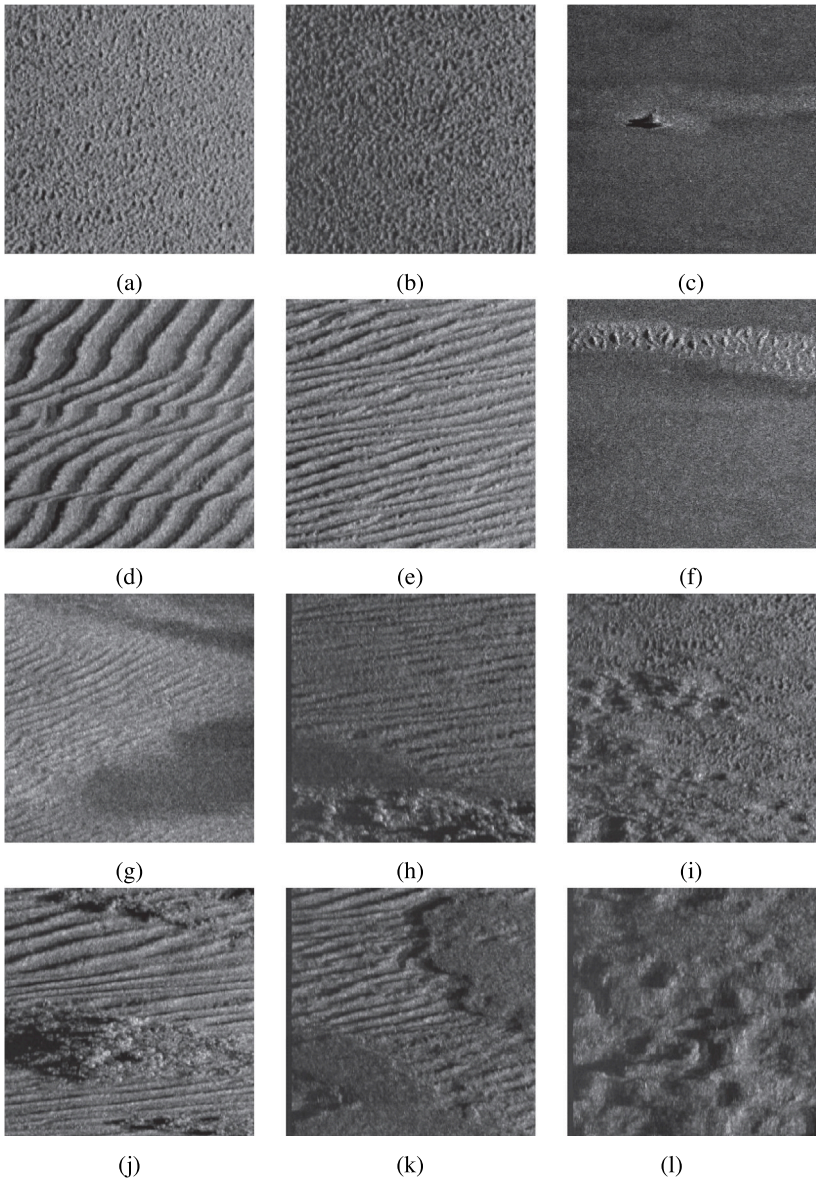

Fig. 6. Dataset overview: (a,b) gravel; (c) fine sediment with a chunk of rock; (d, e) rippled sand; (f) fine sediment with a strip of rippled sand; (g) fine sediment with rippled sand; (h) rocks, fine sediment and rippled sand; (i) rocks and gavel; (j, k) rocks and rippled sand; (l) rocks.

Table 1
Comparison between our proposed architecture and different state-of-the-art CNNs for SSS segmentation.

Method	mIoU (%)	Parameters (M)	FPS	Model size (MB)
ECNet	62.68	4.67	70	18.8
DCNet	78.02	0.09	260	**0.4**
RTSeg	77.75	0.74	190	3.1
Ours†	81.24	**0.08**	**525**	0.47
Ours††	84.65	0.26	311	1.2
Ours‡	88.58	0.97	188	4.1
Ours	**89.54**	1.91	112	7.9

Best results are indicated in bold. Second best results are underlined.

Table 2
Comparison among different self-attention mechanisms for SSS segmentation.

Method	mIoU (%)	Parameters (M)	FPS	Model size (MB)
Swin	84.94	2.35	160	10.3
CSWin	84.95	2.12	146	8.7
LSDA	84.96	2.12	186	8.7
LMHSA	81.31	2.15	176	**8.3**
Wavelet	85.33	4.37	152	17.3
SimXCA	85.97	**2.10**	**204**	8.5
SimXCA†	**86.23**	2.14	186	8.7

Best results are indicated in bold.

Table 3
Ablations of our proposed architecture.

Method	mIoU (%)	Parameters (M)	FPS	Model size (MB)
SimXCA	86.23	2.14	186	8.7
− MLP	88.18	1.98	136	8.1
+ Multimerge	89.28	1.90	116	7.9
+ ASSPP	89.54	1.91	112	7.9

5. Experiments and results

5.1. Comparisons with state-of-the-art CNNs

The CNN-based frameworks RTSeg (Wang et al., 2019), ECNet (Wu et al., 2019) and DCNet (Zhao et al., 2021) were adopted as baselines for primary comparison. We use our own implementation of these architectures since the authors' code was not made publicly available. All the architectural configurations were maintained as suggested in the original publications. However, we noticed certain inconsistencies in the architectures for RTSeg and DCNet. The former had a deviation of about 0.24 M between the number of parameters computed from the description of the architecture and that reported by Wang et al. (2019). Whereas in the latter, Zhao et al. (2021) do not specify the output feature dimensions for convolutions in their DCblock. We tried to set the parameters that we considered best to replicate the architectures as closely as possible. However, the results might deviate from the original implementations of the respective authors.

Table 1 presents a comparison of our proposed architecture with the above-mentioned approaches. Although our architecture has a significantly larger number of parameters than DCNet and RTSeg, the gain in mIoU is also significantly higher. Further, given that the ping rate of our SSS is 20 per second, it takes about 12.8 s to collect 256 swaths. Since each swath has about 1024 bins per side (port and starboard) and we generate images of size 256 × 256 with a 128 pixel-overlap, the minimum processing speed required for real-time segmentation is 14 images per 12.8 s, which equates to 1.01 frames per second. This is including the bins corresponding to the water column which we however do not take into account for segmentation. Even considering the overhead for I/O operations, pre-processing and stitching the segmented images back together into a coherent waterfall, our architecture is readily suitable for real-time segmentation.

In order to ensure a more fair comparison with the aforementioned approaches, Table 1 also presents the results of three variants of our proposed architecture with a significant reduction in parameters to match those of RTSeg and DCNet. These variants are indicated as Ours†, Ours†† and Ours‡. We reduce the lengths of the initial embeddings for the first two variants down to 8 and 12 respectively. We also reduce the number of attention heads down to $\{1, 2, 4, 8\}$ for each of their respective transformer stages. Furthermore, for the first variant, we set the number of layers in each transformer stage to $\{1, 1, 3, 1\}$, whereas for the second and third variants, we set the number of layers in each transformer stage to $\{1, 3, 7, 1\}$. All other configurations remain the same. Despite the significant reduction in parameters, our architectures perform substantially better. With this, we establish a new state-of-the-art for SSS segmentation.

5.2. Feasibility of ViTs for SSS segmentation

The secondary objective of our study is to evaluate the feasibility of ViTs for applications such as SSS segmentation, which typically lack sufficiently large datasets. We therefore draw comparisons of our modified architecture with certain notable self-attention mechanisms in the vanilla hierarchical ViT setting, as presented in Table 2.

Specifically, to switch back to a vanilla hierarchical ViT, we discard all our architectural modifications and simply employ the original MLP block within each transformer layer and use 3 × 3 convolutions with a stride of 2 for patch merging. Furthermore, we adopt the decoder as proposed by Xie et al. (2021) in its original form, without our modified ASPP module. We then train different architectures by replacing the self-attention modules in each transformer layer with the self-attention mechanism proposed by Liu et al. (2021) (Swin), Dong et al. (2022) (CSWin), Wang et al. (2022a) (LSDA), Yao et al. (2022) (Wavelets block) and Guo et al. (2022) (LMHSA block). Since, the approach to self-attention proposed by Wang et al. (2022a) works by alternately applying long-distance attention and short-distance attention in different layers, it suggests the use of an even number of transformer layers for each transformer stage. Therefore, in order to ensure a fair comparison, we set the number of transformer layers L to $\{2, 4, 16, 2\}$ for the four encoder stages for all flavours of attention mechanisms, unless stated otherwise. Also, we do not include positional encodings when employing the Wavelets block, the LMHSA block or SimXCA as the attention mechanism, as suggested by the respective works. All other architectural configurations remain the same. Additional hyperparameter configurations specific to each self-attention mechanism, are as given below:

- **LSDA:** Group size, $G = 8$
- **SWin:** Window size, $M = 8$
- **CSWin:** Stripe Width for each encoder stage, $sw = \{1, 2, 8, 8\}$
- **LMHSA block:** Spatial reduction scale of *key* and *value* embeddings for each encoder stage, $s = \{8, 4, 2, 1\}$
- **Wavelet block:** Spatial reduction scale of *key* and *value* embeddings for each encoder stage, $s = \{4, 2, 1, 1\}$

The LMHSA block generates quite poor segmentation masks due to the loss of information that results from spatial downsampling of *key* and *query* embeddings. The Wavelet block significantly recovers this drop in quality by leveraging wavelet transforms that allow invertible downsampling. However, this approach turns out to be too parameter-heavy and also quite computationally expensive. Although with a slightly lower mIoU, the window-based self-attention mechanisms are quite efficient. The variability in their performance stems from the adopted windowing mechanism, where LSDA performs the best due to the alternating structure of local and global attention that results in self-attention computation between fewer tokens than Swin or CSWin. However, window-based self-attention mechanisms are still a compromise to global self-attention. SimXCA, on the other hand,

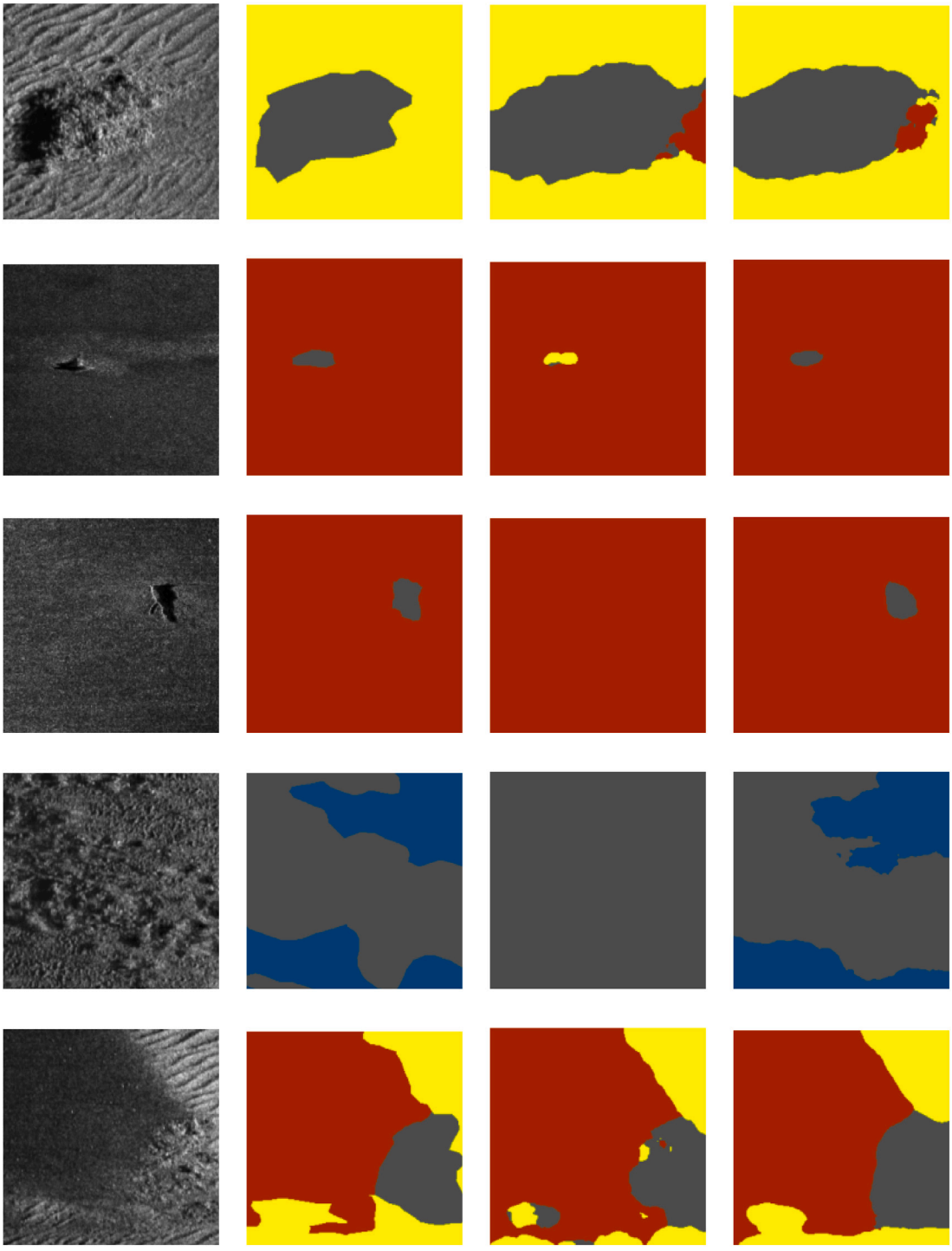

Fig. 7. Visualization of segmentation masks.

gives the best results owing to its transposed self-attention computation and absence of Softmax-based normalization. We also observed that redistributing the number of transformer layers L to $\{3, 6, 12, 3\}$ for SimXCA produces slightly better segmentation masks, while keeping the total number of transformer layers the same. We indicate this as SimXCA† in Table 2.

In Table 3, on the other hand, we ablate our modified architecture to present the individual performance gains from each of our architectural modifications. The top row represents SimXCA in the vanilla hierarchical ViT setting with the number of transformer layers L to $\{3, 6, 12, 3\}$. We then replace the MLP block within each transformer layer with our modified feature extraction block, denoted as "– MLP". Next, we replace the patch merging module in the resultant architecture with our proposed multiscale patch merging module, denoted as "+ Multimerge". Finally, we add the proposed ASSPP block to the decoder design, completing our modified architecture.

Furthermore, Figs. 7 and 8 illustrate the segmentation masks generated by our modified architecture as compared to those generated by

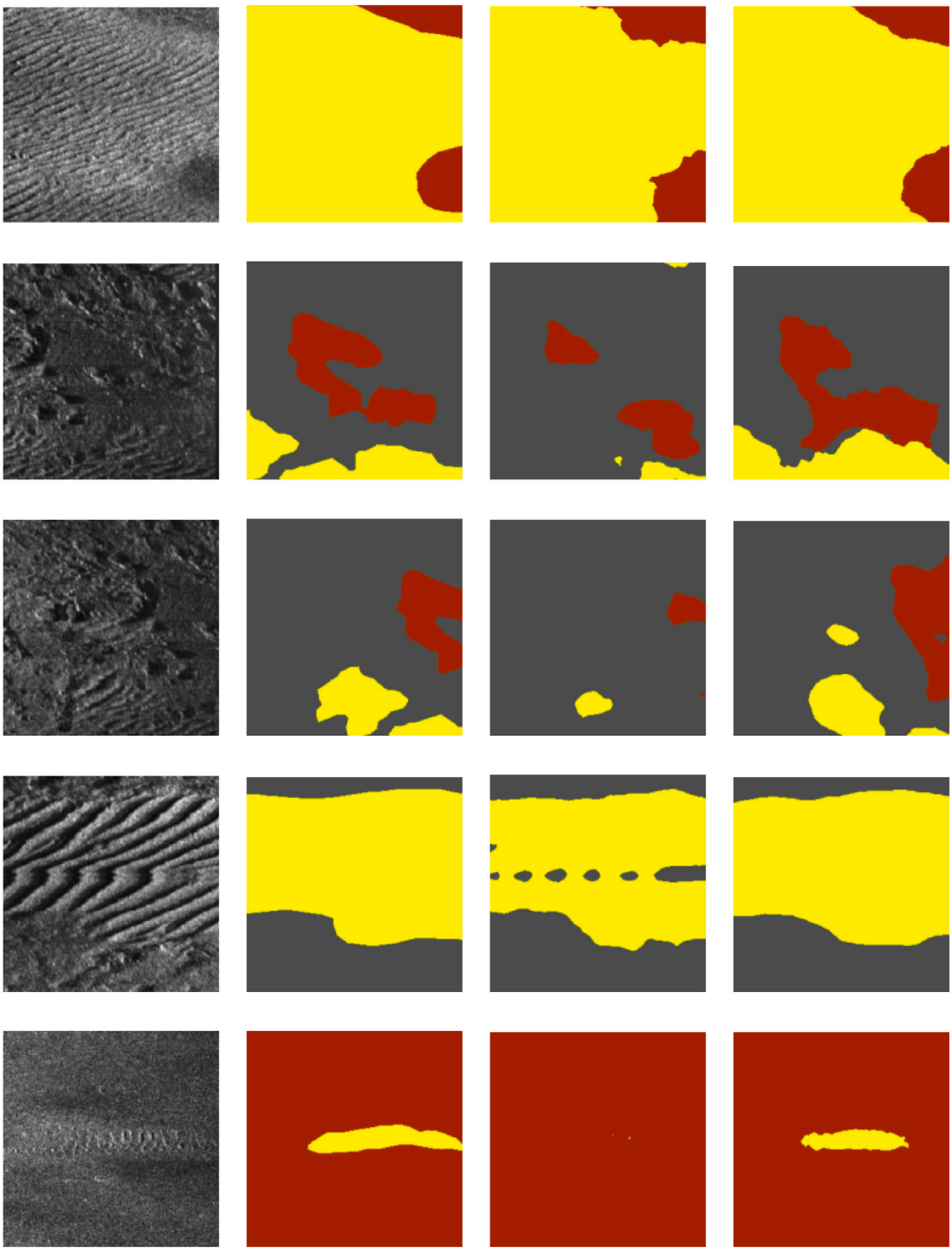

Fig. 8. Visualization of segmentation masks (continued).

SimXCA in the hierarchical ViT setting. Despite the noisy groundtruth, our modified architecture is able to generalize quite well as depicted in the top row of the figure. Moreover, our model is also effective in representing classes such as fine sand ripples and small pebbles of rocks, which the vanilla ViT misses in most cases.

6. Conclusion and future work

In this work we demonstrate the applicability of ViTs for semantic segmentation of the seafloor in SSS waterfalls. To the best of our knowledge, we are the first to employ ViTs for this task. Despite having a small dataset, through our modified design, we achieve results that surpass previous state-of-the-arts by a significant margin while also meeting the computational considerations for real-time implementation.

However, we are still constrained by the lack of precise ground truth to supervise model training. To overcome this weak supervision, we are currently investigating Self-Supervised pre-training followed by Weakly Supervised fine-tuning on image-level labels while also leveraging our noisy ground truth as pseudo masks to regularize training.

Moreover, with the help of the geophysical surveys being conducted by Tecnoambiente SL, we are in the midst of expanding our dataset with additional classes, pixel- and image-level annotations, navigation information and auxiliary metadata. We plan for an eventual release of a large-scale SSS dataset for seafloor segmentation to facilitate further research in this direction.

CRediT authorship contribution statement

Hayat Rajani: Conceptualization, Methodology, Software, Data curation, Writing – original draft. **Nuno Gracias:** Conceptualization, Project administration, Supervision, Writing – review & editing. **Rafael Garcia:** Conceptualization, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code and pretrained models have been made available through the following link: https://github.com/hayatrajani/s3seg-vit.

Acknowledgements

This study was supported by the DeeperSense project, funded by the European Union's Horizon 2020 Research and Innovation programme under grant agreement no. 101016958. The study was also supported in part by the SIREC project, funded by the Ministerio de Ciencia e Innovación, Gobierno de España under agreement no. PID2020-116736RB-IOO. We would further like to acknowledge the contribution of Tecnoambiente SL in the dataset generation effort and are grateful for the insightful discussions with Borja Martinez-Clavel on SSS imagery.

References

Ali, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al., 2021. Xcit: Cross-covariance image transformers. Adv. Neural Inf. Process. Syst. 34, 20014–20027.

Ba, J.L., Kiros, J.R., Hinton, G.E., 2016. Layer normalization. arXiv preprint arXiv:1607.06450.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (12), 2481–2495.

Bousselham, W., Thibault, G., Pagano, L., Machireddy, A., Gray, J., Chang, Y.H., Song, X., 2021. Efficient self-ensemble for semantic segmentation. arXiv preprint arXiv:2111.13280.

Burguera, A., Bonin-Font, F., 2020. On-line multi-class segmentation of side-scan sonar imagery using an autonomous underwater vehicle. J. Mar. Sci. Eng. 8 (8), 557.

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., Wang, M., 2021. Swin-unet: Unet-like pure transformer for medical image segmentation. arXiv preprint arXiv:2105.05537.

Celik, T., Tjahjadi, T., 2011. A novel method for sidescan sonar image segmentation. IEEE J. Ocean. Eng. 36 (2), 186–194.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 801–818.

D'Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L., 2021. ConViT: Improving vision transformers with soft convolutional inductive biases. In: Proceedings of the 38th International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 139, pp. 2286–2296, URL https://proceedings.mlr.press/v139/d-ascoli21a.html.

Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B., 2022. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12124–12134.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=YicbFdNTTy.

Guo, J., Han, K., Wu, H., Tang, Y., Chen, X., Wang, Y., Xu, C., 2022. Cmt: Convolutional neural networks meet vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12175–12185.

Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C., 2020. Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1580–1589.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al., 2019. Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1314–1324.

Huang, Z., Ben, Y., Luo, G., Cheng, P., Yu, G., Fu, B., 2021. Shuffle transformer: Rethinking spatial shuffle for vision transformer. arXiv preprint arXiv:2106.03650.

Koohpayegani, S.A., Pirsiavash, H., 2022. Sima: Simple softmax-free attention for vision transformers. arXiv preprint arXiv:2206.08898.

Li, Y., Yao, T., Pan, Y., Mei, T., 2022. Contextual transformer networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell..

Lianantonakis, M., Petillot, Y.R., 2007. Sidescan sonar segmentation using texture descriptors and active contours. IEEE J. Ocean. Eng. 32 (3), 744–752.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022.

Ma, H., Xia, X., Wang, X., Xiao, X., Li, J., Zheng, M., 2022. MoCoViT: Mobile convolutional vision transformer. arXiv preprint arXiv:2205.12635.

Mehta, S., Rastegari, M., 2022. MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=vh-0sUt8HlG.

Mignotte, M., Collet, C., Pérez, P., Bouthemy, P., 1999. Three-class Markovian segmentation of high-resolution sonar images. Comput. Vis. Image Underst. 76 (3), 191–204.

Mignotte, M., Collet, C., Perez, P., Bouthemy, P., 2000. Sonar image segmentation using an unsupervised hierarchical MRF model. IEEE Trans. Image Process. 9 (7), 1216–1231.

Ren, S., Zhou, D., He, S., Feng, J., Wang, X., 2022. Shunted self-attention via multi-scale token aggregation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10853–10862.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., 2018. MobileNetV2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Si, C., Yu, W., Zhou, P., Zhou, Y., Wang, X., Yan, S., 2022. Inception transformer. arXiv preprint arXiv:2205.12956.

Srinivas, A., Lin, T.-Y., Parmar, N., Shlens, J., Abbeel, P., Vaswani, A., 2021. Bottleneck transformers for visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16519–16529.

Strudel, R., Garcia, R., Laptev, I., Schmid, C., 2021. Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. ICCV, pp. 7262–7272.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2818–2826.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Wang, Q., Wu, M., Yu, F., Feng, C., Li, K., Zhu, Y., Rigall, E., He, B., 2019. Rt-seg: A real-time semantic segmentation network for side-scan sonar images. Sensors 19 (9), 1985.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L., 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 568–578.

Wang, W., Yao, L., Chen, L., Lin, B., Cai, D., He, X., Liu, W., 2022a. CrossFormer: A versatile vision transformer hinging on cross-scale attention. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=_PHymLIxuI.

Wang, Z., Zhang, S., Gross, L., Zhang, C., Wang, B., 2022b. Fused adaptive receptive field mechanism and dynamic multiscale dilated convolution for side-scan sonar image segmentation. IEEE Trans. Geosci. Remote Sens. 60, 1–17.

Wu, Y., He, K., 2018. Group normalization. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 3–19.

Wu, M., Wang, Q., Rigall, E., Li, K., Zhu, W., He, B., Yan, T., 2019. Ecnet: Efficient convolutional networks for side scan sonar image segmentation. Sensors 19 (9), 2009.

Wu, S., Wu, T., Tan, H., Guo, G., 2022. Pale transformer: A general vision transformer backbone with pale-shaped attention. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 2731–2739, (3).

Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L., 2021. Cvt: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22–31.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. In: Advances in Neural Information Processing Systems. vol. 34, pp. 12077–12090, URL https://proceedings.neurips.cc/paper/2021/file/64f1f27bf1b4ec22924fd0acb550c235-Paper.pdf.

Yao, K., Mignotte, M., Collet, C., Galerne, P., Burel, G., 2000. Unsupervised segmentation using a self-organizing map and a noise model estimation in sonar imagery. Pattern Recognit. 33 (9), 1575–1584.

Yao, T., Pan, Y., Li, Y., Ngo, C.-W., Mei, T., 2022. Wave-vit: Unifying wavelet and transformers for visual representation learning. In: European Conference on Computer Vision. Springer, pp. 328–345.

Yu, F., He, B., Li, K., Yan, T., Shen, Y., Wang, Q., Wu, M., 2021. Side-scan sonar images segmentation for AUV with recurrent residual convolutional neural network module and self-guidance module. Appl. Ocean Res. 113, 102608.

Yu, F., He, B., Liu, J., Wang, Q., 2022. Dual-branch framework: AUV-based target recognition method for marine survey. Eng. Appl. Artif. Intell. 115, 105291.

Zhang, Q., Xu, Y., Zhang, J., Tao, D., 2022. ViTAEv2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. arXiv preprint arXiv:2202.10108.

Zhao, X., Qin, R., Zhang, Q., Yu, F., Wang, Q., He, B., 2021. Dcnet: Dilated convolutional neural networks for side-scan sonar image semantic segmentation. J. Ocean Univ. China 20 (5), 1089–1096.