MERLIN-Seg: self-supervised despeckling for label-efficient semantic segmentation
Emanuele Dalsasso, Clément Rambour, Nicolas Trouvé, Nicolas Thome

To cite this version:
Emanuele Dalsasso, Clément Rambour, Nicolas Trouvé, Nicolas Thome. MERLIN-Seg: self-supervised despeckling for label-efficient semantic segmentation. 2023. hal-04163624

HAL Id: hal-04163624
https://cnam.hal.science/hal-04163624
Preprint submitted on 17 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
MERLIN-Seg: self-supervised despeckling for label-efficient semantic segmentation

Emanuele Dalsasso, Clément Rambour, Nicolas Trouvé, Nicolas Thome

Conservatoire National des Arts et Métiers, 75003 Paris, France
The French Aerospace Lab, ONERA, 91120 Palaiseau, France
Sorbonne Université, CNRS, ISIR, F-75005 Paris, France

ABSTRACT

This article has been submitted to CVIU and is currently under review. Remote sensing satellites acquire a continuous stream of data on a daily basis. As most of those data are unlabeled, the development of algorithms requiring weak supervision is of paramount importance. In this paper, we show that the need for annotation for Synthetic Aperture Radar data can be reduced by coupling a despeckling task (self-supervised) and a segmentation task (supervised). The proposed self-supervised learning framework, called MERLIN-Seg, has been trained for building footprint extraction and achieves favorable performances even with 1% of annotated data. We show that conditioning the network on despeckling without labels is beneficial for supervised segmentation. Our experiments demonstrate that the joint training of the two tasks achieves better performances than a vanilla segmentation network in terms of IoU, F1 score, and accuracy on both simulated and real SAR images.

1. Introduction

Earth observation satellites carry onboard different kind of remote sensors collecting information characterizing the Earth. Among these, Synthetic Aperture Radar (SAR) is an active system with imaging capabilities. As an active sensor, it can collect information at any time of the day and in (almost) all weather conditions, providing continuous and global coverage of the Earth’s surface. This powerful feature of SAR allows access to cloud-covered areas, such as the tropics and subtropics, subject to long wet seasons and frequent precipitation.

Deep learning algorithms play an essential role in the analysis of remote sensing data (Zhu et al., 2017). Once they are tuned and deployed, they can process data in a quick manner to extract useful information for Earth monitoring. To train such algorithms, one often needs a large set of labeled data. In remote sensing, the great abundance of images is guaranteed by the presence of numerous sensors embarked on satellites in orbit around the Earth and the increased adoption of open data policies. However, these images are generally not annotated, making data labeling one of the big open challenges in remote sensing (Wang et al., 2022b).

In the computer vision community, alternatives to vanilla supervised algorithms requiring labeled data have been proposed: so-called self-supervised learning (SSL) approaches have been shown to learn powerful representations for many tasks (Xian et al., 2017; Chen et al., 2020b; Lehtinen et al., 2018; Finn et al., 2017; Le-Khac et al., 2020). SSL has recently sparked great attention in remote sensing (Wang et al., 2022b), but still few works were proposed to deal with SAR data. In an SSL framework, a neural network is pre-trained on a self-supervised...
pretex task and then fine-tuned on the actual downstream task. This relieves the network of the burden of seeing many labeled data, reducing the need for annotation (Zheng et al., 2021). Contrastive learning approaches comprise another family of SSL methods where a network is trained to force a close latent representation between augmented views of the same image (and possibly push away representations from negative samples). To this aim, one may resort to techniques such as image rotation, mirroring, and image colorization. (Baranchuk et al., 2022; Brempong et al., 2022) have demonstrated that denoising pre-training learns useful semantic representations. As an alternative to this sequential scheme, a neural network can be trained simultaneously on both downstream task and pretext task, such as image denoising. In DenoiSeg (Buchholz et al., 2021), while learning to reduce noise from both unlabeled and labeled images, the network efficiently co-learns to segment even on a few labeled samples.

The application of SSL to SAR data is not straightforward (Wang et al., 2022b). Augmenting SAR data is indeed very sensitive: unlike optical images, SAR images are complex-valued, their pixels represent measurements with physical meaning, and the appearance of the scene is intimately related to the sensor position and acquisition mode. Moreover, SAR images are affected by speckle, a multiplicative perturbation with specific statistics.

In this paper, we present MERLIN-Seg, a general framework to address the shortage of annotations in SAR images, relying on self-supervised despeckling (Dalsasso et al., 2021a). As in (Buchholz et al., 2021), we simultaneously train our model on the self-supervised despeckling and supervised segmentation tasks. We empirically demonstrate that our SSL framework is beneficial for the segmentation of SAR images, especially when the availability of labels is low. We test our approach on the specific task of building footprint segmentation on simulated and real data. The main contributions of our work are the following:

- To the best of our knowledge, we are the first to propose to use despeckling as an SSL technique to extract semantic features from SAR data for label-efficient semantic segmentation. In particular, we propose a generalization of the MERLIN framework (Dalsasso et al., 2021a) for the simultaneous learning of despeckling and segmentation.
- The proposed MERLIN-Seg approach extracts semantically meaningful features from SAR images without the need of designing SAR-specific augmentation. We argue that SAR images can be seen as inherently augmented due to the presence of speckle. Thus, we exploit the nature of SAR data to devise a physically meaningful self-supervised task.
- We show that the simultaneous learning of SAR despeckling and segmentation is more effective than SSL pre-training on SAR despeckling, although both strategies are relevant and outperform a fully supervised learning on the footprint segmentation task when few labels are available.
- We validate MERLIN-Seg on both EMPIRE simulated SAR images and real TerraSAR-X data, highlighting the versatility of the proposed approach to different acquisition modalities and resolutions. Moreover, our approach shows to perform well both in dense and sparse urban environments.

2. Related Work

**SAR despeckling** SAR is a coherent imaging system. As such, the complex SAR signal $z$ is the result of the coherent summation of many elementary echoes coming from the same resolution cell. For rough surfaces at the wavelength scale and in absence of predominant scatterers, the measured intensity $I = |z|^2$ presents strong fluctuations that severely limits the exploitation of SAR images. Such fluctuations take the name of speckle phenomenon. The SAR intensity $I$, the underlying reflectivity $R$ and the speckle $S$ are linked by a multiplicative relation: $I = R \times S$ (Goodman, 2007). The aim of despeckling techniques is to suppress the speckle and restore the underlying reflectivity.

Previously unseen performances has been obtained by SAR despeckling approaches relying on a self-supervised training strategy (Fracastoro et al., 2021; Dalsasso et al., 2022). Indeed,
Fig. 1: Illustration of the proposed MERLIN-Seg approach for joint despeckling and segmentation

the difficulty in SAR imaging to produce the clean groundtruth signal $R$ limits the development of supervised methods. SSL for SAR despeckling stems from the advances in self-supervised image denoising (Lehtinen et al., 2018; Krull et al., 2019; Buchholz et al., 2021). Given two independent noisy images $y_i$ and $y_j$, representing the same underlying clean signal $x$, a neural network $f_\theta(.)$ can be trained to maximize the agreement between network’s output $f_\theta(y_i)$ and an independent noisy image $y_j$. If the noise likelihood is known, the network loss can be defined as $\mathcal{L} = -\log p(y_j|f_\theta(y_i))$. Because $y_i$ and $y_j$ are i.i.d. samples, the network cannot predict the random component of $y_j$ starting from $y_i$. Instead, it will predict the deterministic common component, i.e. the clean signal $x$. This way, a neural network can be trained for image denoising by only seeing noisy examples.

Depending on the way of splitting the dataset into two independent sets, one can class self-supervised despeckling approaches into three categories:

- **Multi-image self-supervised approaches** exploit images acquired at different dates (sufficiently spaced-apart so that the speckle is temporally decorrelated). Thus, one obtains several samples representing the same underlying scene, provided that the changes can be compensated (Dalsasso et al., 2021b).

- In single-image self-supervised approaches, one can rely on specific architectures excluding the central pixel from the network receptive field (so-called blind-spot networks (Laine et al., 2019; Lee and Jeong, 2022)) and keep it apart to supervise network training. The specificity required by such a receptive field limits the choice of architecture and might reduce the network’s performance. Moreover, to verify the independence of the hidden pixel with the neighboring ones, it requires the speckle to be spatially independent: such a hypothesis is however not verified in practice, as data providers often deliver spatially-correlated SAR products, and whitening the speckle requires a tedious and touchy pre-processing, especially if sensors parameters are unknown (Lapini et al., 2013).

- **Most recently**, MERLIN approach (Dalsasso et al., 2021a) has demonstrated that any architecture can be trained on single SAR images by exploiting the complex nature of SLC SAR data. MERLIN exploits the phase information to split the intensity image into two independent sub-images, namely the real and imaginary parts, creating ideal conditions for self-supervised learning based on
noise2noise (Lehtinen et al., 2018).

Unlike the first two families of techniques, MERLIN does not make any hypothesis on temporal coherence (as it is trained on single dates), nor on speckle spatial correlation (as the network implicitly learns it). For these reasons, MERLIN would fit like a glove in an SSL framework for SAR imaging.

**Semantic Segmentation and building footprint** Semantic segmentation has been a popular topic in computer vision for many years. In the last decade, Convolutional Neural Networks (CNNs) have been shown to be highly effective in learning discriminative features and achieving high accuracy in various image segmentation tasks. U-shaped CNN architectures such as Unet (Ronneberger et al., 2015), UpperNet (Xiao et al., 2018) or DeepLab (Chen et al., 2018) and their variants set new standards in computer vision. Recently, transformer architectures tend to show better performances in semantic segmentation at the cost of strong parameters overhead (Liu et al., 2021; Xie et al., 2021; Themyr et al., 2023).

In the domain of remote sensing, land cover is a challenging task due to the strong contrast dynamics, the lack and/or quality of annotation, and the large size of remote sensing images. Hybrid CNN-Transformers architectures tend to show the best performances for multi-spectral image segmentation (Wang et al., 2022a; Scheibenreif et al., 2022). Concerning SAR data, the lack of large high-resolution annotated datasets, the precence of speckle and geometrical distortions strongly limit the capacity of very large models. Moreover, due to the different sensing techniques large models learned on optical images can not be directly transferred for this task.

**Semi and Self Supervised Learning** Semi-supervised learning aims to make the most of the available data in cases where only a small portion of the dataset is labeled. We can roughly classify the semi-supervised learning methods into three families: self-training with pseudo labels (Xie et al., 2020b), generative models (Ehsan Abbasnejad et al., 2017; Dai et al., 2017), and consistency regularization (Miyato et al., 2018; Tarvainen and Valpola, 2017; Laine and Aila, 2017; Valpola, 2015). Self-training involves iteratively extending the labeled dataset with high-confidence predictions from the model. Generative models learn the underlying distribution of the data and then use it to generate unseen samples. Finally, consistency regularization techniques involve enforcing the model’s output to be consistent across different perturbations of the input such as additive noise, contrast jittering, or random flip (Xie et al., 2020a; Cubuk et al., 2020). This enforces a smoothness assumption on the input space. Seminal works include Ladder Networks (Valpola, 2015) or Pi-model (Laine and Aila, 2017) where consistency is enforced between features maps or predictions from clean and noisy input. Adversarial training in semi-supervised learning aims to ensure smooth prediction on a ball around the data (Miyato et al., 2018).

Recent approaches in SSL rely on the same strategy where a pretext task is designed and solving it requires learning useful image representations. These techniques achieve state-of-the-art performance over approaches that learn representations from unsupervised data only. Visual pretext tasks include grayscale image colorization (Larsson et al., 2017), image patches localization and orientation prediction (Komodakis and Gidaris, 2018), jigsaw puzzles (Noroozi and Favaro, 2016), or inpainting (Pathak et al., 2016). Similarly to earlier semi-supervised approaches, contrastive methods encourage the model to learn representations invariant to strong augmentations (Chen et al., 2020a; Grill et al., 2020; Chen and He, 2021). Due to the intrinsic lack of annotations in remote sensing data, semi-supervised and self-supervised approaches appear as valuable strategies to train efficient classification and segmentation models. SSL approaches for SAR image classification and target recognition include contrastive strategies as well as patch rotation prediction (Ren et al., 2021; Xu et al., 2021; Zhang et al., 2019).

However, the design of pretext tasks for SAR images is critical as they need to preserve the physical sense and the geometry of SAR data. (Zhang and Ma, 2022) shows that some augmentations can have a negative effect on certain downstream tasks, as they can lead to undesired invariances suppressing discriminative features or to the loss of fine-grained information. For instance, geometric distortions appearing in SAR images such
as shadowing and layover depend on sensor’s orientation and are not rotation invariant.

3. Method

We propose to reduce the need for annotations by coupling a despeckling task (self-supervised) with a segmentation task (supervised) into a unique label-efficient framework: MERLIN-Seg. While segmentation can only rely on labeled data, self-supervised despeckling can benefit from the large amount of raw data available to learn a semantic representation of the data, without the need of augmenting them. The joint training of the two tasks reduces the labeling requirements for the supervised task. In the following, the despeckling task will be first addressed. Then, we will discuss how despeckling and segmentation can simultaneously be trained into a unique semi-supervised framework.

3.1. Self-supervised despeckling: MERLIN

We propose to use in this framework the self-supervised MERLIN speckle reduction strategy (Dalsasso et al., 2021a). Not only it achieves good restoration performances, but it is also very flexible as it is agnostic to the network architecture. Moreover, by training the network on Single-Look Complex (SLC) SAR images, it learns sensor-specific parameters (e.g., speckle spatial correlation) by its own.

Goodman’s model of speckle (Goodman, 2007) describes the complex speckle as a circular Gaussian distributed random variable \( s \). The signal \( z = a + jb \) measured over an area with reflectivity \( R \) is defined as \( z = s \sqrt{R} \). Thus, the distribution of \( z \) is given by:

\[
p_Z(z) = \frac{1}{\pi R} \exp\left(-\frac{|z|^2}{R}\right) = \frac{1}{\pi R} \exp\left(-\frac{a^2 + b^2}{R}\right).
\]  

(1)

Based on the complex speckle model, in (Dalsasso et al., 2021a) it has been shown that a complex SAR image \( z = a + jb \) can be decomposed into two i.i.d. components \( a \) and \( b \sim N(0, R/2) \) (with \( R \) the image of the reflectivity, i.e., the variance of the \( k \)-th pixel is \( R_k/2 \)), each one containing half of the information of an intensity image \( I = a^2 + b^2 \). This implies that a neural network can be trained in a self-supervised manner for speckle reduction as follows: the network \( f_a(\cdot) \) takes as input one component (e.g., the real part \( a \)) and it evaluates the quality of the restored reflectivity image \( \tilde{R} = f_a(a) \) with respect to the other component (e.g., the imaginary part \( b \)). This is achieved by maximizing the likelihood of \( b \) with respect to the network output \( \tilde{R} \):

\[
\mathcal{L}_{\text{MERLIN}}(\tilde{R}, b) = \sum_k -\log p(b_k|\tilde{R}_k)
= \sum_k \frac{1}{2} \log \left(\frac{b_k^2}{\tilde{R}_k}\right),
\]  

(2)

where index \( k \) indicates the \( k \)-th pixel of the estimated reflectivity image \( \tilde{R} \) or of the imaginary part \( b \). In practice, because of their independence, during training the real and the imaginary parts are permuted at each iteration.

At inference time, the trained network is applied separately on the real and imaginary part: the two intermediate estimations \( f_a(a) \) and \( f_b(b) \) are finally averaged to obtain the final reflectivity estimation.

3.2. MERLIN-Seg: joint despeckling and segmentation

In a classical segmentation framework through deep learning, a network is trained to produce, from the input image scene, a segmentation map \( \tilde{l} \) that has to be as close as possible to the groundtruth map \( l \). To optimize network weights for a binary segmentation problem, one can use the weighted Binary Cross-Entropy (BCE) defined as:

\[
\mathcal{L}_{\text{BCE}}(\tilde{l}, l) = \sum_k p_l \log(\tilde{l}_k) + (1 - l_k) \log(1 - \tilde{l}_k)
\]  

(3)

with \( p = \#\text{negative samples}/\#\text{positive samples} \) being a weight assigned to positive examples, \( l_k \) being the class label and \( \tilde{l}_k \) the output class label of pixel \( k \).

To implement MERLIN-Seg, we propose a generalization of MERLIN as follows. The SAR image \( z = s \sqrt{R} \) can be seen as augmentation of its reflectivity \( R \) by the complex speckle \( s \). The noisy nature of SAR images is leveraged within MERLIN-Seg to extract semantic features through self-supervised despeckling with MERLIN, without the need of augmenting SAR.
data. According to (Baranchuk et al., 2022), intermediate feature maps of a U-Net model for image denoising encode semantic representations of the input image. Thus, we upsample all feature maps from the upscaling branch of our U-Net model employed in MERLIN and stack a 3 convolutional layers to extract the segmentation map. For the despeckling task, MERLIN processes separately squared real and imaginary parts $a^2$ and $b^2$, characterized by a poorer signal-to-noise ratio (SNR) than the intensity image $I = a^2 + b^2$ by a factor of $\sqrt{2}$. To exploit all the available information for the segmentation task, we compute the feature maps from both real and imaginary parts with a sharing-weights network and we do an average before feeding them to the 3-layers segmentation head. We choose to have a lightweight segmentation head to limit the number of segmentation-specific parameters, and maximize the parameters shared between the tasks. Such a choice is beneficial to facilitate the tuning of the segmentation head when only few labels are available for the downstream task.

The proposed two-heads architecture is trained end-to-end to minimize the following loss function:

$$L_{\text{MERLIN-Seg}} = \lambda L_{\text{MERLIN}}(\tilde{r}, \tilde{b}) + (1 - \lambda)L_{\text{BCE}}(\tilde{l}, l)$$

with $\lambda \in [0, 1]$ an hyperparameter to balance the weight of the two tasks during training. Setting it to 0 corresponds to plain MERLIN despeckling. When no labels are available, the segmentation loss $L_{\text{BCE}}(\tilde{l}, l)$ is set to 0. However, cross-task parameters are still tuned thanks to self-supervised despeckling: while learning to suppress speckle from raw SAR data, the network co-learns to segment on labeled SAR data.

4. Experiments

This section describes the set up of the experiments, the two considered datasets for building footprint segmentation and discusses the obtained results.

4.1. Experimental Settings

Datasets. We evaluated the proposed MERLIN-seg method on two different SAR land cover datasets. The first one is a Single Look Complex (SLC) high resolution simulation of
12600x10800 pixels at X-band provided by the Onera’s E-PRIME simulator\(^1\) (Fig.2, top image). The slant-range resolution is 0.71 m and azimuth resolution is 1 m. E-PRIME can generate raw data and realistic SAR images based on the modeling of the physical properties of scatterers present in the 3D scene. In our case, the scene is rendered from cadastral and landcover information, digital elevation models, and fine-grained field surveys of the city of Sainte-Marie in the south of France. All ground cover elements ranging from small bushes to whole buildings are modeled in 3D and ground surface is generated at an average resolution of 5.3cm. The full 3D database cover over 10 by 10 km area. The simulation process allows us to generate perfect labels at the image resolution including SAR effects such as layover or side lobes.

The second dataset is obtained from an SLC TerraSAR-X 6000x10000 SpotLight image acquired in July 2012 over the city of Paris\(^2\) and its surrounding areas. The slant-range resolution is 0.45 m and the azimuth resolution is 0.87 m (Fig.2, bottom image). The labels are extracted from the BDtopo database provided by the French National Geographic Institute (IGN). The BDtopo can be freely downloaded and consists of vector data describing the 2.5D buildings geometry: each georeferenced footprint is associated with the object’s height. We performed the projection of the labels from nadir to the true incidence angle from the TerraSAR-X sensor. The scene is composed of dense urban areas and its segmentation is challenging due to the multiple overlays and shadows projected by the buildings.

Implementation details. To set up our experiments, we build on the U-Net model as in (Dalsasso et al., 2021a). To generalize the network so that it performs simultaneous despeckling and segmentation, we define an extra 3-layers segmentation head composed of 64 kernels of size 3 × 3 followed by Leaky ReLU, 32 kernels of size 3 × 3 followed by Leaky ReLU and a final 1 × 1 sized kernel followed by sigmoid activation to extract the class labels for each image pixel. The final binary segmentation mask is obtained by applying a threshold set to 0.5.

The E-PRIME simulated dataset is split into 3 non-overlapping subsets as follows: network is fed with 11161 image patches with a stride of 64, the validation set is composed of 5 non-overlapping image patches, whereas the test is conducted on 70 non-overlapping image patches. All patches are of size 256 × 256. The U-Net backbone is composed of 5 downsampling layers (compressing the 256 × 256 image patch down to an 8 × 8 latent representation) and 5 upscaling layers. The TerraSAR-X dataset is split into 3 independent subsets as follows: 1189 patches with a stride of 128 for training, 3 non-overlapping patches for validation and 65 non-overlapping patches for testing. All patches are of 512 × 512 pixels. The U-Net backbone as one extra downsampling and upsampling layer w.r.t. the network described above. The choice of patch size and network depth is done to enlarge network receptive field.

For the segmentation baseline, to allow a fair comparison we consider the same U-Net model as in MERLIN and replace the last layer with a 1 × 1 convolutional filter followed by Sigmoid, in order to extract class labels (Fig.1). The input of this network is the intensity SAR image. The network is trained with \(\mathcal{L}_{BCE}\) loss function on labeled data only. Both the segmentation baseline and MERLIN-Seg are trained for 100 epochs with a learning rate of 0.001 and Adam optimizer. For each percentage of labeled data, network weight's are randomly initialized and training is done from scratch. We highlight that when only a limited number of annotated samples are available, the segmentation baseline only sees labeled patches in input. On the contrary, thanks to the self-supervised despeckling task, MERLIN-Seg always sees all the data. In the latter, batches are sampled randomly, i.e. we don’t impose any constraint on the number of labeled images composing a batch.

4.2. Results

The baseline for both experiments is a standard U-Net trained with binary cross-entropy on labeled data. Table 1 summarizes our results on the simulated dataset for different proportions of labeled data. On this dataset, MERLIN-Seg is on par with

---

\(^1\)http://emprise-em.fr

\(^2\)These data have been provided by the DLR in the framework of the project LAN1746.
the baseline when 100% of labels are available with 0.71 mIoU achieved by both methods. MERLIN-Seg performances are stable even when only 1% labels are used for training. In this case, MERLIN-Seg yields 0.63 mIoU and 0.75 F1 score while the fully supervised baseline performances collapse with only 0.20 mIoU and 0.30 F1 score. On real data, the same behavior can be observed and is summarized in table 2. In this configuration, MERLIN-Seg slightly outperforms the supervised baseline when all the labels are available showing 0.76 against 0.75 mIoU for the baseline. The model trained with the MERLIN-Seg strategy is particularly stable as the performance drops are limited to 0.01 both for mIoU and F1 score. The fully supervised model shows a drop up to 0.12 mIoU and 0.08 F1 score and is outperformed by MERLIN-Seg with about 0.12 mIoU and 0.1 F1 score. Quantitative evaluation is confirmed by visual inspection of the results (Fig.3).

While the role of despeckling is to learn semantically meaningful features for downstream segmentation and we are not interested in its performances, it is worth to point out that high quality restoration is achieved on both EMPRISE and TerraSAR-X images. No residual speckle fluctuations seem to be left in the restored image, which is finely restored. We cannot provide a quantitative evaluation on SAR despeckling as the groundtruth is not available.

4.3. Ablation study

In this part, we analyze different aspect of MERLIN-Seg.
Table 2: Results with $\lambda = 0.5$ on TerraSAR-X dataset

<table>
<thead>
<tr>
<th></th>
<th>100%</th>
<th>20%</th>
<th>5%</th>
<th>1%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mIoU</td>
<td>F1</td>
<td>Acc</td>
<td>mIoU</td>
</tr>
<tr>
<td><strong>Baseline</strong></td>
<td>0.748</td>
<td>0.832</td>
<td>0.874</td>
<td>0.746</td>
</tr>
<tr>
<td><strong>MERLIN-Seg</strong></td>
<td>0.765</td>
<td>0.844</td>
<td>0.886</td>
<td>0.760</td>
</tr>
</tbody>
</table>

**Features aggregation** MERLIN processes separately the real and the imaginary parts for the despeckling task, due to their independence. A naive extension of MERLIN would consist in providing to the segmentation head only the feature maps estimated from the real part (resp. the imaginary part) and keep the imaginary part (resp. the real part) to supervise the despeckling task. In this configuration, the segmentation task only rely on a partial information and the performances are even below the baseline for all metrics on pure segmentation ($\lambda = 0$): see table 3, second row. To incorporate the information of both real and imaginary parts for downstream segmentation, two strategies are considered. The feature maps are computed for both real and imaginary parts. Then, they can be aggregated either by concatenated them, or by doing features average. While real and imaginary parts remain separated for despeckling, they can be both exploited for segmentation. The two aggregation strategies have proved to be effective as they outperform the baseline. We choose to keep the average as it shows to be slightly more efficient and it requires less training parameters.

Table 3: Results on EMPRISE dataset when setting $\lambda = 0$. MERLIN-Seg with features mean shows higher performances than the other strategies on building footprint segmentation.

<table>
<thead>
<tr>
<th></th>
<th>IoU</th>
<th>F1</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.710</td>
<td>0.803</td>
<td>0.974</td>
</tr>
<tr>
<td><strong>MERLIN-Seg</strong> (w/o aggregation)</td>
<td>0.671</td>
<td>0.750</td>
<td>0.970</td>
</tr>
<tr>
<td><strong>MERLIN-Seg</strong> (w/ concatenation)</td>
<td><strong>0.713</strong></td>
<td><strong>0.805</strong></td>
<td><strong>0.975</strong></td>
</tr>
<tr>
<td><strong>MERLIN-Seg</strong> (w/ mean)</td>
<td><strong>0.713</strong></td>
<td><strong>0.806</strong></td>
<td><strong>0.975</strong></td>
</tr>
</tbody>
</table>

**The impact of $\lambda$** The value of the $\lambda$ parameter directly impact the relative importance of the denoising and segmentation tasks. This tuning would depend on the proportion of annotated data but one would expect the performances to be sufficiently stable with respect to $\lambda$ to avoid painful hand-tuning. Table 4 shows the performances of the segmentation task for different values of $\lambda$ in the context where only 1% of labels are available. The performances achieved by MERLIN-Seg are quite stable for $\lambda$ values samples from 0.1 to 0.9 and the best regime is obtained for $\lambda = 0.5$ which gives a practical rull of thumb.

It is fundamental to observe the particular case of $\lambda = 0$. A drastic drop of performances occurs if self-supervised despeckling is not applied. This result strongly supports our thesis: given the same network architecture, the joint learning of segmentation and self-supervised despeckling ($\lambda \neq 0$) is highly beneficial w.r.t. a fully supervised training on segmentation only ($\lambda = 0$).

Table 4: Results on EMPRISE building footprint dataset when varying $\lambda$. The label percentage is fixed to 1% for all experiments.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>IoU</th>
<th>F1</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda = 0$</td>
<td>0.263</td>
<td>0.382</td>
<td>0.899</td>
</tr>
<tr>
<td>$\lambda = 0.1$</td>
<td>0.600</td>
<td>0.715</td>
<td>0.959</td>
</tr>
<tr>
<td>$\lambda = 0.2$</td>
<td>0.589</td>
<td>0.704</td>
<td>0.960</td>
</tr>
<tr>
<td>$\lambda = 0.5$</td>
<td><strong>0.634</strong></td>
<td><strong>0.745</strong></td>
<td><strong>0.965</strong></td>
</tr>
<tr>
<td>$\lambda = 0.7$</td>
<td>0.628</td>
<td>0.744</td>
<td>0.964</td>
</tr>
<tr>
<td>$\lambda = 0.9$</td>
<td>0.607</td>
<td>0.727</td>
<td>0.960</td>
</tr>
</tbody>
</table>

**End-to-end vs. Fine-tuning** We compare the proposed joint learning strategy with a fully SSL pre-training approach: once the network is pre-trained on self-supervised despeckling, its
weights are frozen and the feature maps are fed to the segmentation head to be fine-tuned on labeled data. We refer to this strategy as sequential approach. In this training configuration, we employ the same network as in MERLIN-Seg. The quantitative evaluation reported in Table 5 allows to say while self-supervised pre-training learns semantically meaningful features, showing good segmentation performances of building footprint segmentation when only 1% of the EMPIRE dataset is annotated, the sequential approach is outperformed by MERLIN-Seg. From this experiments we conclude that despeckling is a relevant task for SSL pre-training, although an end-to-end semi-supervised learning is still preferable.

Table 5: Results of sequential approach (fine-tuning) versus MERLIN-Seg (end-to-end) on EMPIRE building footprint dataset. The label percentage is fixed to 1% and for MERLIN-Seg we set $\lambda = 0.5$.

<table>
<thead>
<tr>
<th></th>
<th>IoU</th>
<th>F1</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>0.422</td>
<td>0.556</td>
<td>0.930</td>
</tr>
<tr>
<td>MERLIN-Seg</td>
<td>0.634</td>
<td>0.745</td>
<td>0.965</td>
</tr>
</tbody>
</table>

5. Conclusion

We have shown that self-supervised despeckling extract semantically meaningful features of SAR images for downstream segmentation task. The segmentation task benefits from despeckling, especially when few annotated samples are available: the features learned on despeckling guide segmentation, making the learning of the segmentation task easier.

In contrast to classical contrastive learning frameworks, our approach does not require to apply any augmentation to the data. Indeed, the speckle is seen as an intrinsic augmentation of SAR images and our model is trained to recover the speckle-free image in a self-supervised fashion.

While our experiments have been conducted on a U-shaped model, MERLIN-Seg is model agnostic and it can thus profit from the most performing networks for image segmentation. Moreover, its performances can be improved by resorting to additional input information, may it be multi-modal (e.g. multispectral images (Bergamasco et al., 2023)) and/or multi-temporal (e.g. SAR time series (Meraoumia et al., 2023)): the network would extract richer semantic information from the extra input to better despeckle SAR images, thus facilitating the downstream task.

6. Acknowledgements

This project has been funded by ANR (the French National Research Agency) and DGA (Direction Générale de l’Armement) under ASTRAL project ANR-21-ASTR-0011. We would like to thanks french DGA for supporting the EMPIRE project and Onera’s partner Scalian-DS for its major implication in the development of the simulation software.

References


