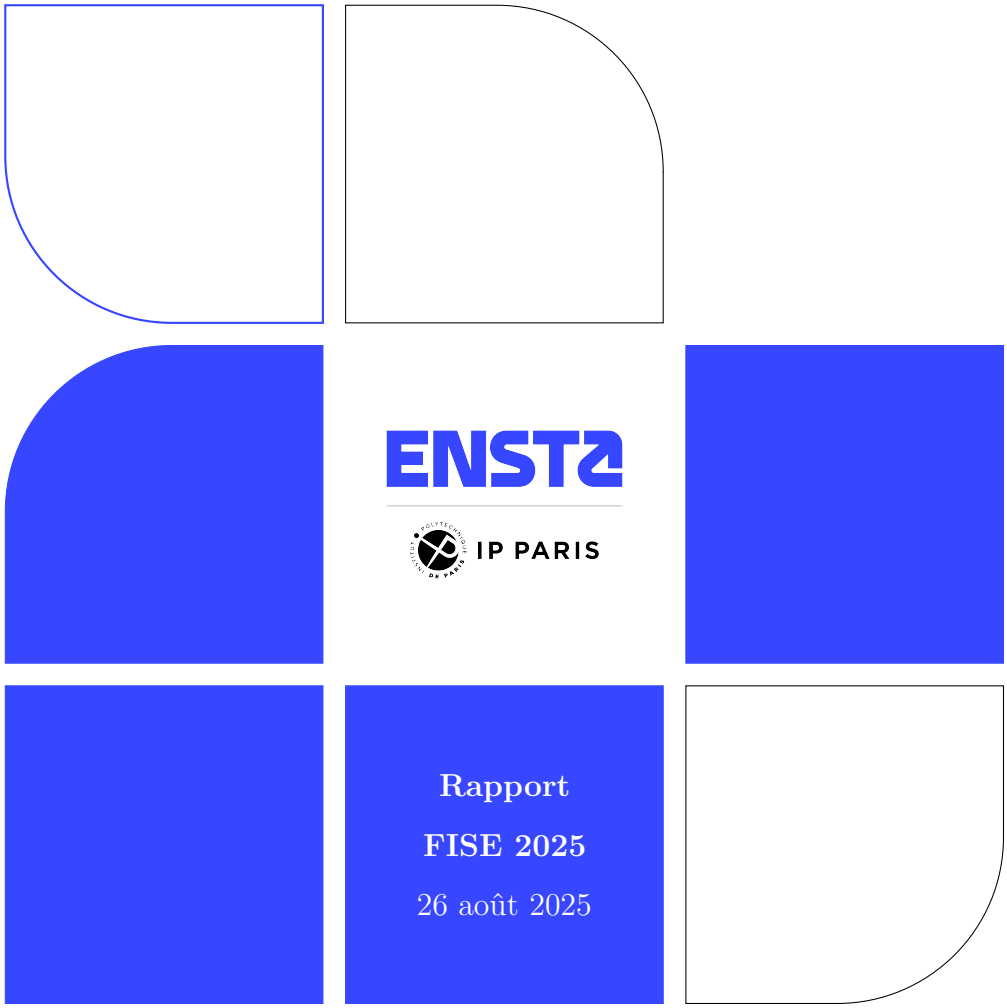


Self-supervised learning for semantic segmentation of polarimetric SAR imagery

Théo THUILLIER
theo.thuillier@ensta.fr

Supervised by : Fabrice Comblet (ENSTA), Ronny Hänsch (DLR)

This report is not confidential

Résumé

La segmentation sémantique de l'imagerie Radar à Synthèse d'Ouverture Polarimétrique (PolSAR) est cruciale pour la surveillance environnementale, mais la précision de la segmentation est souvent limitée par la rareté des données annotées. L'acquisition de vérité terrain fiables pour les données PolSAR est intrinsèquement difficile, nécessitant souvent des campagnes de terrain complexes et exigeantes sur le plan logistique. Cette étude explore l'apprentissage auto-supervisé (SSL) pour pallier cette limitation en utilisant le jeu de données de référence Pol-InSAR-Island [15]. Nous proposons un cadre méthodologique qui emploie un auto-encodeur masqué (MAE) pour apprendre des représentations de caractéristiques robustes à partir de données PolSAR non étiquetées, lesquelles sont ensuite affinées au sein d'une architecture U-Net pour la segmentation sémantique. Nous avons mené une étude d'ablation complète pour comparer le modèle pré-entraîné par SSL à un modèle de référence entièrement supervisé. Cette analyse a évalué systématiquement l'impact des différentes représentations de données, stratégies de normalisation et techniques d'augmentation de données sur les performances du modèle. Les résultats démontrent un gain de performance substantiel grâce au pré-entraînement SSL, augmentant l'Intersection sur Union moyenne (IoU) de 21,71 % (modèle de référence supervisé) à 36,93 %. De plus, le pré-entraînement a amélioré la stabilité de l'entraînement, réduisant de moitié le coefficient de variation (CV) entre les exécutions, de 1,22 à 0,66. Notre analyse a confirmé qu'une représentation des données en log-ratio étendue, combinée à une stratégie de standardisation tronquée et de normalisation par écrêtage, a fourni les meilleures performances. Bien que les techniques d'augmentation de données comme CutMix aient offert des améliorations modérées, la contribution du pré-entraînement SSL s'est avérée nettement plus significative, en particulier pour la segmentation des classes présentant des frontières complexes et hétérogènes. Ces résultats établissent l'apprentissage auto-supervisé comme une stratégie très efficace pour la segmentation sémantique PolSAR, démontrant que des caractéristiques puissantes et transférables, apprises à partir de données non étiquetées, peuvent permettre une classification de haute précision même avec un nombre très limité d'échantillons étiquetés.

Abstract

Semantic segmentation of Polarimetric Synthetic Aperture Radar (PolSAR) imagery is crucial for environmental monitoring, but the segmentation accuracy is often limited by the scarcity of annotated data. Acquiring reliable ground-truth labels for PolSAR is inherently challenging, often requiring complex and logistically demanding field campaigns. This study investigates self-supervised learning (SSL) to mitigate this limitation using the Pol-InSAR-Island benchmark dataset [15]. We propose a framework that employs a Masked Autoencoder (MAE) to learn robust feature representations from unlabeled PolSAR data, which are subsequently fine-tuned within a U-Net architecture for semantic segmentation. We conducted a comprehensive ablation study to compare the SSL-pretrained model against a fully supervised baseline. This analysis systematically evaluated how different data representations, normalization strategies, and data augmentation techniques affect model performance. The results demonstrate a substantial performance gain from SSL pretraining, boosting the mean Intersection over Union (IoU) from 21.71% (supervised baseline) to 36.93%. Furthermore, the pretraining enhanced training stability, halving the coefficient of variation (CV) across runs from 1.22 to 0.66. Our analysis confirmed that an extended log-ratio data representation combined with a trimmed standardization and clipping normalization strategy yielded the best performance. While data augmentation techniques like CutMix offered moderate improvements, the contribution from SSL pretraining was markedly more impactful, especially for segmenting classes with complex, heterogeneous boundaries. These findings establish SSL as a highly effective strategy for PolSAR semantic segmentation, demonstrating that powerful, transferable features learned from unlabeled data can enable high-accuracy classification even with a severely limited number of labeled samples.

Keywords

Deep Learning, self-supervised learning, SAR polarimetry, semantic segmentation, data representation, masked autoencoder.

Acknowledgments

I would like to express my sincere gratitude to all those who have supported and contributed to the completion of this report. My deepest thanks go to my supervisor, Ronny Hänsch, for his guidance and mentorship throughout my internship, as well as for providing me with the opportunity to investigate this field. I am also grateful for the experience and knowledge gained during my time at the organization. I wish to thank my colleagues from the HR-STE group for their valuable feedback and insightful discussions, and for helping me in use the GPU on the server. My gratitude also goes to professor, Ali Khenchaf for introducing me to the DLR, and to Fabrice Comblet, my school tutor, for his advice and guidance throughout my academic journey. I am especially thankful to Alberto Moreira for considering my application. Finally, I extend my heartfelt appreciation to my family for their unwavering support and encouragement throughout this project.

Contents

1 Introduction . . . 8
1.1 Host institution . . . 8
1.2 Context . . . 8
1.3 Objectives . . . 9
1.4 Outlines . . . 9
2 Background Concepts . . . 10
2.1 Synthetic Aperture Radar (SAR) . . . 10
2.2 Polarimetric SAR (PolSAR) . . . 13
2.3 Semantic Segmentation . . . 13
2.4 Impact of input representation for image processing . . . 14
2.5 Data augmentation for image processing . . . 16
2.6 Self-supervised Learning (SSL) . . . 17
3 Materials . . . 19
3.1 Pol-InSAR-Island Dataset . . . 19
3.2 Dataset Processing . . . 20
4 Deep learning-based Methodology . . . 24
4.1 Deep Learning model used . . . 24
4.2 Training methodology . . . 26
4.3 Data augmentation implemented . . . 27
4.4 Metrics for performance evaluation . . . 27
5 Results . . . 29
5.1 Impact of the Masking Rate . . . 29
5.2 Impact of the Representation on the Model Performance . . . 30
5.3 Impact of the Normalization on the Performance . . . 31
5.4 Impact of Data Augmentation on the Model Performance . . . 32
6 Conclusion . . . 35

Bibliography **36**

List of Tables

1 SAR Bands, Applications, and Example 10
2 Class Distribution 20
3 Acronyms defining the network training process. 29
4 Average IoU (mean ± std) obtained for each representation, aggregated over all normalization strategies. 31
5 Average IoU (mean ± std) obtained for each normalization method, aggregated over all representations. 32
6 Performance metrics by class for simple data augmentation (Mean IoU ± Std). Summary statistics are reported at the bottom. 33
7 Performance metrics by class for targeted data augmentation and references (Mean IoU ± Std). Summary statistics are reported at the bottom. 33

List of Figures

1 Illustration of the SAR imaging geometry. r_0 stands for the shortest approach distance, Θ_a for the azimuth beamdwidth and v for the sensor velocity. Adapted from [30]. . . . 11
2 Summary of basic Synthetic Aperture Radar (SAR) processing step. Adapted from [30]. 12
3 Illustration of image segmentation. Adapted from [41]. 13
4 Schematic illustration of the Masked Autoencoder principle, adapted from [14]. 18
5 Schematic representation of the F-SAR flight paths used for data acquisition. Adapted from [15] 19
6 Dataset file structure and decomposition of coherency matrices. Adapted from [15]. . . 20
7 Histogram of the ground truth distribution in the training set 21
8 Original ground-truth segmentation map (left), and train (middle) / test (right) partitions based on the "maze" split. 21
9 Histogram of the data distribution among the samples showing the first value of the coherency matrix (T11). The right plot excludes the top 5% of values. 23
10 U-Net architecture implemented to perform semantic segmentation. The schema was made from an adaptation of the code [19] 25
11 Schema showing the weights transfer from the Masked Autoencoder (MAE) to the U-Net model. 25
12 Masked Autoencoder architecture with similar encoder as the U-Net model implemented to perform semantic segmentation. The schema was made from an adaptation of the code [19]. 26
13 Masking strategy for SSL: 32×32 squares are randomly placed on a 256×256 patch with a masking rate of 40%. 27
14 Visual comparison of the masking rate impact on the reconstruction task. 30
15 Average IoU as a function of the masking rate. Error bars indicate one standard deviation across 10 independent simulations. Results are shown for the log-diagonal (min–max 5%-trimmed) and extended log-ratio (standardized 5%-trimmed) representations. . . . 31
16 Comparison of segmentation maps obtained with different training strategies. SSL provides the most accurate and stable results across all land cover classes. CutMix improves performance for homogeneous classes but produces over-smoothed boundaries in complex ecological transitions. Random crop demonstrates balanced segmentation performance with preserved spatial detail. Gaussian noise severely degrades performance by introducing artifacts and misclassifications, while horizontal and vertical flips, and channel dropout yield moderate improvements with variable class-dependent performance. 34

Glossary

CE Cross Entropy. 27

DLR German Aerospace Center. 19

FP1 Flight Path 1. 19, 21
FP2 Flight Path 2. 19, 21
FSL Fully Supervised Learning. 17, 24

IoU Intersection over Union. 35

MAE Masked Autoencoder. 18, 24, 26, 29
MSE Mean Square Error. 17, 26

PCA Principal Component Analysis. 17
Pol-InSAR Polarimetric Interferometric SAR. 19

RCM Range Cell Migration. 12

SAR Synthetic Aperture Radar. 6, 10, 12, 17
SSL Self-supervised Learning. 17, 24, 26, 29, 35

1 Introduction

1.1 Host institution

This report is the outcome of an internship conducted at the German Aerospace Center (DLR), Microwaves and Radar Institute, in Oberpfaffenhofen, Germany. The DLR, established in 1969, serves as the national research center for aerospace, energy, and transportation, and is responsible for implementing the German space program.

The internship was carried out within the SAR Technologies Department, led by Dr. Andreas Reigber, mainly focusing on the development of the F-SAR sensor, an advanced airborne Synthetic Aperture Radar (SAR) system featuring full polarimetry and multi-frequency capabilities. Technical supervision for the internship was provided by Dr. Ronny Hänsch, who leads the machine learning team within the Signal Processing Group of the SAR Technology Department.

1.2 Context

Climate change and environmental pressures have highlighted the need for reliable Earth observation systems capable of monitoring diverse landscapes and processes. Optical remote sensing has traditionally been an important tool in this context, yet its applicability is often limited by cloud coverage, illumination conditions, or adverse weather. Synthetic Aperture Radar (SAR), on the other hand, provides day-and-night, all-weather imaging capabilities and can operate across different frequency bands, offering a powerful and flexible means of observing the Earth's surface. Depending on the acquisition parameters, SAR is widely used for tasks such as vegetation monitoring, ice and snow assessment, or coastal zone mapping.

The airborne F-SAR sensor developed at DLR enables high-resolution, fully polarimetric acquisitions across several frequency bands, which can even be operated simultaneously. These characteristics allow F-SAR to support a broad range of applications, from forestry and agriculture to hydrology and cryosphere research. However, each acquisition campaign is conducted under very different conditions, depending on geographic location and scientific objectives: one mission may cover the German North Sea coast, another tropical rainforest, and another Arctic permafrost. As a result, the data distribution varies strongly between campaigns. Thus, training a machine learning model to generalize from one campaign's data to another is neither possible nor reasonable. Developing a model that can be trained by using data of one campaign only, without the need to generalize to other campaigns, is fully sufficient if it only performs well for this one campaign it was trained on.

A central task in SAR data analysis is semantic segmentation, i.e. the classification of each pixel into a meaningful land cover class. This problem is particularly relevant for polarimetric SAR (PolSAR) and interferometric SAR (InSAR) data, as they provide rich information about scattering mechanisms and surface structure. Semantic segmentation of SAR imagery is a challenging task as the data are complex-valued, highly variable across frequencies and environments. Recent advances in deep learning have shown strong potential for improving segmentation performance [53], but they rely on large, annotated datasets. In the SAR domain, labeled data are scarce, since expert knowledge is required for manual annotation.

To address this limitation, pretraining strategies such as self-supervised learning (SSL) have emerged as a promising solution. The key idea is to leverage large amounts of unlabeled SAR data to learn general feature representations that capture structural and statistical properties of the sensor measurements. Once pretrained, these models can be fine-tuned on a small number of labeled samples from the same campaign, thereby reducing annotation requirements while maintaining high segmentation accuracy. Recent works in remote sensing confirm the benefit of SSL for limited-label scenarios, showing improved stability and accuracy across multiple tasks [45].

1.3 Objectives

The main objective of this internship is to investigate the potential of SSL for improving the semantic segmentation of PolSAR imagery in a campaign-specific setting. In particular, this work addresses three research questions:

- To what extent can masked autoencoder (MAE) pretraining improve network performance on a segmentation task?
- How does the choice of data representation and normalization impact the final performance of the model?
- How do different data augmentation strategies affect model accuracy and stability?

1.4 Outlines

This report is organized as follows: Section 1 gives the context of the work done and introduces the objectives. Section 2 presents the fundamental principles of PolSAR, semantic segmentation, and image preprocessing, along with an introduction to self-supervised learning. The Pol-InSAR-Island benchmark dataset and the processing methods implemented are described in Section 3. In Section 4, we present the deep learning approach used. Section 5 discusses the results obtained and display some visual comparison. Finally, the findings of this study and the perspectives are summarized in Section 6.

2 Background Concepts

This section presents the fundamental concepts that form the basis of this study. It begins by introducing the principles of Synthetic Aperture Radar (SAR) in Section 2.1 and its extension, Polarimetric SAR (PolSAR), in Section 2.2. Following this, Section 2.3 outlines the primary task of semantic segmentation. The subsequent sections delve into crucial preprocessing considerations, discussing the impact of input data representation and data augmentation techniques in Sections 2.4 and 2.5. Finally, the section concludes by presenting the self-supervised learning (SSL) paradigm in Section 2.6, the key methodology investigated in this work.To evaluate the impact of the input representation, trainings were run for every representation–normalization combination described in Section 4. For each representation, we calculated the mean and standard deviation of the IoU across all normalizations. The results are summarized in Table 4.

2.1 Synthetic Aperture Radar (SAR)

The SAR is an active remote sensing system mounted on a moving device, e.g., aircraft or spacecraft. It illuminates a target surface with microwaves and retrieves the reflected signal (backscattered). This process enables the generation of high-resolution images that are unaffected by daylight, cloud cover, or weather conditions [47]. During each cycle of emission and reception, the system measures the characteristics of the backscattered pulse, such as time delay, amplitude, and phase, in order to produce an image. The moving capacity of the system enables it to combine multiple received signals to synthetically create an aperture much larger than the physical antenna [30].

SAR systems have the capacity to adapt the wavelength to the application and overcome problems that arise when using only optical sensors. This characteristic makes them useful for Earth observation. The common bands used are the X band [2.4–3.8 cm], the C band [3.8–7.5 cm], the S band [7.5–15 cm], the L band [15–30 cm], and the P band [30–100 cm]. Band selection depends on the application, as it involves a trade-off between penetration and resolution. The lower the wavelength, the higher the resolution, but the penetration is reduced. Table 1 summarizes the result and shows some examples of missions using each band.

Band	Applications	Missions
X [2.4-3.8] cm	High resolution (urban monitoring, ice/snow,)	TerraSAR-X, COSMO-SkyMed
C [3.8-7.5] cm	Agriculture, ocean, maritime navigation	Sentinel-1, RADARSAT-2
S [7.5-15] cm	Agriculture, Atmosphere	HJ-1C, NISAR
L [15-30] cm	Forestry, soil moisture, geology	ALOS-2, SAOCOM
P [30-100] cm	Biomass estimation, subsurface imaging	BIOMASS (ESA)

Table 1 – SAR Bands, Applications, and Example

The previously described system can be illustrated by its geometry, schematized in Figure 1. It consists of a platform moving in the *azimuth* along track direction, and where the *slant range* is the perpendicular direction to the radar flight path. The covered radar scene is delimited by the *swath width*, which defines the extent of the ground range. SAR systems emit a chirp, a waveform that is linearly frequency modulated, resulting in the current instant frequency $f_i(t) = f_c + k_r \cdot t$ where f_c is the carrier frequency and k_r is the chirp rate. This leads to a bandwidth of $B_r = k_r \cdot \tau$ where τ is the duration of the pulse. The system then receives the echo, which is digitized into a two-dimensional data matrix. The range dimension is referred to as fast time, while the azimuth dimension is referred to as slow time. The distance between the target at the coordinates $(x_0, 0, \Delta h)$, and the radar moving at a constant velocity v is computed with the Pythagorean theorem.

To illustrate SAR geometry, Figure 1 from [30], shows a moving platform in the *azimuth* / along-track direction, and where the *slant range* is the perpendicular direction to the radar flight path. The covered radar scene is delimited by the *swath width*, which defines the extent of the ground range, and

the duration of the data take defines the azimuth extent. SAR systems emit a chirp, a waveform that is linearly frequency modulated, resulting in the current instant frequency $f_i(t) = f_c + k_r \cdot t$ where f_c is the carrier frequency and k_r is the chirp rate. This leads to a bandwidth of $B_r = k_r \cdot \tau$ where τ is the duration of the pulse. The system then receives the echo, a two-dimensional data matrix which is referred to in range as *fast time* and in azimuth as *slow time*. The distance between the target at the coordinates $(x_0, 0, \Delta h)$, and the radar moving at a constant velocity v is computed with the Pythagorean theorem as:

$$r(t) = \sqrt{r_0^2 + (vt)^2}, \quad r_0 = \sqrt{(H-h)^2 + x_0^2}, \tag{1}$$

where $r_0 = r(t=0)$. The product of the illumination time (t) by the velocity (v) is shorter than r_0, which satisfies the condition $\frac{v \cdot t}{r_0} \ll 1$ and enables us to make the approximation:

$$r(t) \approx r_0 + \frac{(vt)^2}{2r_0}. \tag{2}$$

Finally, the phase variation is linked to the azimuth by the relation:

$$\varphi(t) = \frac{-4\pi r(t)}{\lambda}, \tag{3}$$

where λ is the wavelength.

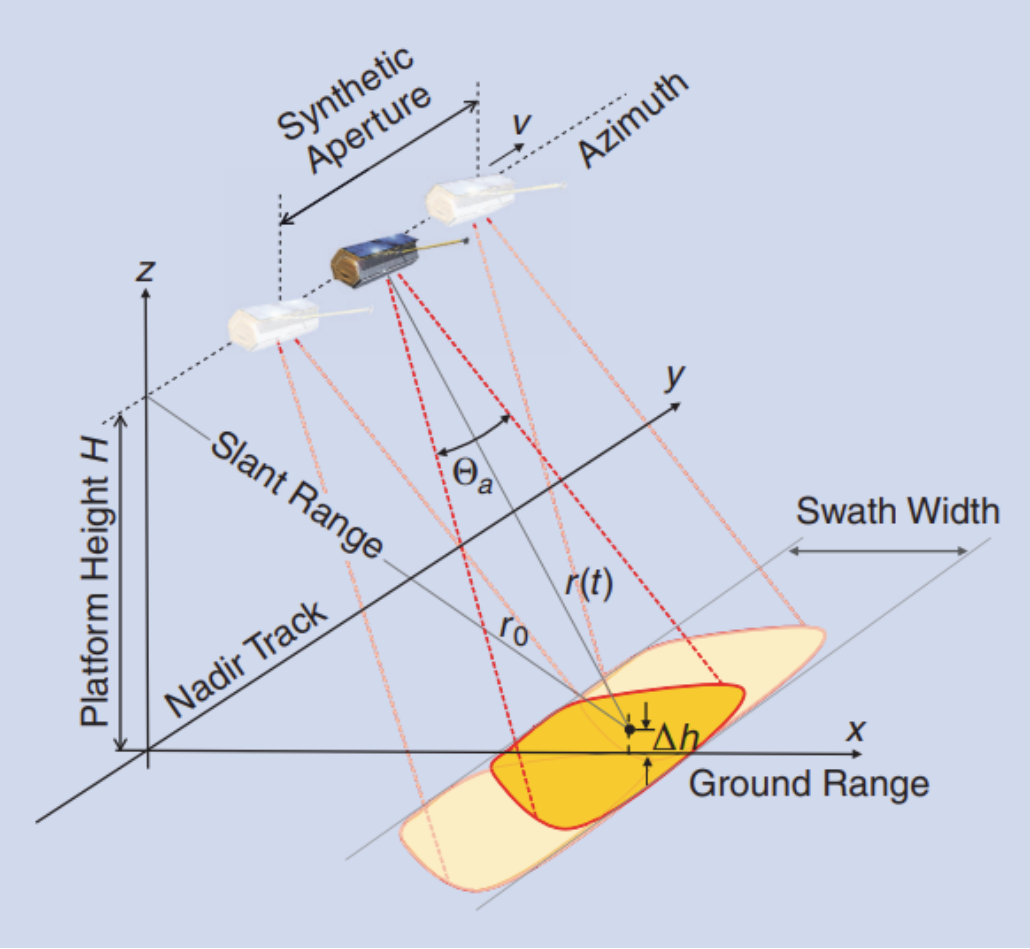


Figure 1 – Illustration of the SAR imaging geometry. r_0 stands for the shortest approach distance, Θ_a for the azimuth beamdwidth and v for the sensor velocity. Adapted from [30].

The raw signal cannot be used directly and requires several signal processing steps to create a high-resolution image. The basic SAR processing *range compression* and *azimuth compression* are summarized in Figure 2. Range compression uses matched filtering in the frequency domain by correlating the received signal with the complex conjugate of the transmitted chirp spectrum, resulting in a slant-range resolution of:

$$\delta_r = \frac{c_0}{2B_r}, \tag{4}$$

where c_0 is the speed of light and B_r the bandwidth of the transmitted pulse. Azimuth compression exploits Doppler frequency variations induced by platform motion to synthetically enlarge the antenna aperture by a length of:

$$L_{\text{sa}} = \frac{r_0}{\lambda} d_a, \tag{5}$$

where d_a is the real antenna length, leading to an azimuth resolution of:

$$\delta_a = \frac{d_a}{2}. \tag{6}$$

The effective illumination time, defined as:

$$T_{\text{ill}} \approx \frac{r_0 \lambda}{v d_a}, \tag{7}$$

increases for shorter antennas, thus improving azimuth resolution.

The SAR image output is represented with intensity values, where each pixel represents the reflectivity of the ground at that location. For accurate interpretation, two additional processing steps are essential: *calibration*, to ensure that every point has an intensity representative of its reflectivity, and *geocoding* to guarantee that every pixel is associated with a position on the ground [30]. Various other factors must be considered, including Range Cell Migration (RCM) and speckle [30].

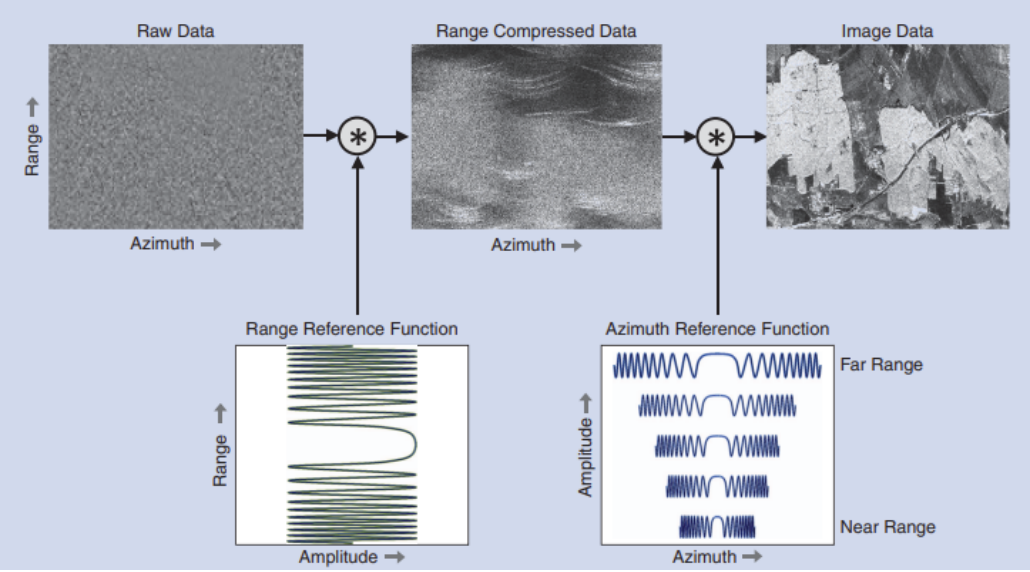


Figure 2 – Summary of basic SAR processing step. Adapted from [30].

2.2 Polarimetric SAR (PolSAR)

Polarimetric Synthetic Aperture Radar (PolSAR) is a remote sensing technique that analyzes the polarization properties of scattered electromagnetic waves. The technique is based on the **complex scattering matrix S**. This matrix describes how a target modifies the polarization of an incident electromagnetic wave. The reflection on the target transforms the incoming plane wave $\vec{E}^i$ into the scattered wave $\vec{E}^r$. This transformation is formalized by the equation:

$$\vec{E}^r = \frac{\exp(-ikr)}{r}[\mathbf{S}]\vec{E}^{i*}, \tag{8}$$

where k is the wavenumber, and r is the range to the target. The common polarizations used are horizontal (H) and vertical (V), which means that matrix 8 becomes:

$$\begin{bmatrix} E_H^r \\ E_V^r \end{bmatrix} = \frac{\exp(-ikr)}{r} \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \begin{bmatrix} E_H^i \\ E_V^i \end{bmatrix}^*. \tag{9}$$

The scattering matrix describes the scattering behavior of each pixel. In the case of a monostatic SAR, where the same antenna is used for transmission and reception, the reciprocity theorem holds $S_{HV} = S_{VH}$. To characterize the scattering process, the coherency matrix [T] (or covariance matrix [C]) is computed thanks to the scattering vector, either in its lexicographic form :

$$\vec{k}_{\text{lex}} = \begin{bmatrix} S_{HH} \\ S_{HV} \\ S_{VV} \end{bmatrix}, \tag{10}$$

or in the Pauli form:

$$\vec{k}_{\text{pauli}} = \frac{1}{\sqrt{2}} \begin{bmatrix} S_{HH} + S_{VV} \\ S_{HH} - S_{VV} \\ 2S_{HV} \end{bmatrix}. \tag{11}$$

The diagonal elements of the matrix 12, T_{11}, T_{22}, T_{33} represent the power in each scattering mechanism (single-bounce, double-bounce, and volume), while the off-diagonal elements encode the correlation between channels. This matrix is particularly useful for discriminating targets that may reflect the same intensity but yield different scattering behavior (vegetation and man-made structures). coherency matrix [T] is then defined as the product of the scattering vector by its Hermitian transpose [30] such as:

$$\mathrm{T}_{3\times3} = \left\langle \vec{k}_{\text{pauli}} \vec{k}_{\text{pauli}}^{\dagger} \right\rangle = \begin{bmatrix} \langle |S_{HH} + S_{VV}|^2 \rangle & \langle (S_{HH} + S_{VV})(S_{HH}^* - S_{VV}^*) \rangle & 2\langle (S_{HH} + S_{VV})S_{HV}^* \rangle \\ \langle (S_{HH} - S_{VV})(S_{HH}^* + S_{VV}^*) \rangle & \langle |S_{HH} - S_{VV}|^2 \rangle & 2\langle (S_{HH} - S_{VV})S_{HV}^* \rangle \\ 2\langle S_{HV}(S_{HH}^* + S_{VV}^*) \rangle & 2\langle S_{HV}(S_{HH}^* - S_{VV}^*) \rangle & 4\langle |S_{HV}|^2 \rangle \end{bmatrix} \tag{12}$$

2.3 Semantic Segmentation

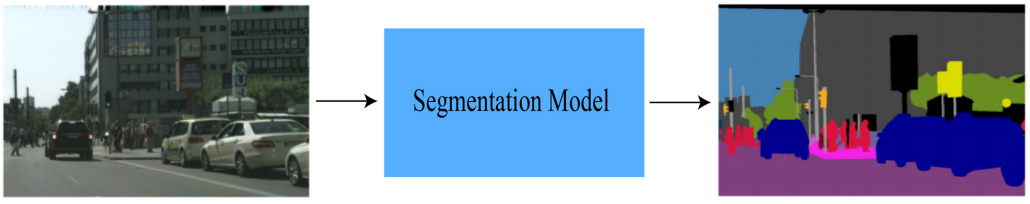


Figure 3 – Illustration of image segmentation. Adapted from [41].

Image segmentation is a subfield of computer vision that consists of dividing an image into multiple coherent regions, thereby transforming raw pixels into more interpretable units. Classification assigns a single label to an entire image, while detection localizes objects using bounding boxes. In contrast, segmentation operates at the pixel level, and each pixel in the image is assigned a semantic label. An illustration of this principle is shown in Figure 3.

Before the development of deep learning–based methods, early approaches to semantic segmentation generally relied on handcrafted features combined with classical machine learning algorithms. Feature descriptors were designed to capture local texture, intensity statistics, or edge information, and classifiers such as support vector machines or random forests were trained to predict pixel labels from these features. While these pipelines achieved useful results, they required substantial manual feature engineering and often struggled to generalize across datasets or imaging conditions.

The rise of deep learning methods has profoundly changed the field of computer vision. Deep neural networks consist of multiple interconnected layers that automatically extract hierarchical features from data. These models excel at complex pattern recognition tasks by optimizing the weights of the connections to minimize the error between the ground Truth and the prediction [11]. A breakthrough moment was the introduction of AlexNet [23], the first deep convolutional neural network (CNN) to achieve state-of-the-art results in large-scale image recognition. CNNs apply convolutional kernels and pooling operations to extract spatial hierarchies, effectively automating the feature extraction process that classical segmentation methods required manually.

Based on the CCN, architectures for semantic segmentation such as DeepLabv3+ [3] and U-Net [46] became state-of-the-art on optical images. DeepLabv3+ integrates dilated convolutions and multiscale feature aggregation to capture both local and global context. U-Net uses an encoder–decoder structure with skip connections that preserve fine-grained spatial details.

Already used for optical images, the described CNN architectures have been adapted to SAR imagery. For example, a U-Net-based framework has been used to segment rivers and land cover in Sentinel-1 imagery [32], while fully convolutional networks (FCNs) such as FCN-ResNet50-32s have been fine-tuned on TerraSAR-X patches for building extraction [49]. Transfer learning from optical-trained FCN and U-Net models has also been applied to high-resolution airborne PolSAR datasets, achieving competitive results in land cover mapping [1].

However, semantic segmentation of SAR images remains challenging due to the scarcity of labeled data. Annotating SAR imagery is both expensive and time-consuming, as it requires a human expert. Unlike optical images, which can benefit from synthetic datasets generated using graphics engines, realistic synthetic SAR data is far more difficult to produce, resulting in a scarcity of labeled datasets. It is therefore sometimes necessary to aggregate the data from several campaigns, which is no easy task [2].

Recent state-of-the-art methods have therefore adapted CNN architectures to better handle these constraints. Inception-based encoder–decoder networks with multiscale skip connections [29] have been proposed to capture features at different resolutions, while multiscale attention-based FCNs (MANet) integrate attention mechanisms to emphasize relevant scattering structures [50]. More advanced designs, such as the Multi-Path Residual Network (MP-ResNet) [6] have been introduced to overcome the limited expressiveness of shallow CNNs, improving the extraction of high-level semantic features without overfitting. Other architectures like HR-SAR-Net [43] demonstrate that carefully designed residual connections and shallow networks can still perform effectively when overfitting is a risk.

2.4 Impact of input representation for image processing

Deep learning models for image processing performance depend not only on the network architecture but also on the data representation. In SAR imagery, where the input data are noisy, high-dimensional, and complex-valued, the representation choice is not easy. Choosing a representation is challenging because it affects feature extraction and convergence speed. A clever preprocessing pipeline can

mitigate classical image processing issues such as varying illumination, inconsistent contrast, and speckle, enabling models to generalize more effectively to unseen data.

A common preprocessing technique used is scaling transformations, making the input data numerically stable. A basic transformation, *normalization*, consists of shifting every pixel intensity into a fixed interval normally $[0, 1]$ or $[-1, 1]$. This min-max transformation is defined by :

$$I_{\text{norm}} = \frac{I - I_{\text{min}}}{I_{\text{max}} - I_{\text{min}}}, \tag{13}$$

where I_{min} represent the minimum and I_{max} the maximum pixel intensities in the dataset. This rescaling of the data enables each characteristic to contribute equally during the learning process, preventing the domination of features of high value and slowing the learning process [13]. Another common transformation is the **standardization**, also known as the *z-score normalization*. This transformation centers the data by subtracting the average and rescales it with a unitary variance, such as:

$$I_{\text{stand}} = \frac{I - \mu}{\sigma}, \tag{14}$$

where μ is the mean and σ is the standard deviation of the pixel values. Unlike the min-max normalization, the standardization is less sensitive to high pixel values because it's using the mean and the standard deviation of the data. However, outliers will still be present in the data.

As described in Sections 2.1 and 2.2, SAR images are inherently complex because they encode the scattering properties of the observed targets. Feeding raw high-dimensional polarimetric data directly into a deep learning model is generally not feasible. This is due to the so-called *curse of dimensionality*: as the number of input dimensions grows, the volume of the feature space increases exponentially, making the available training samples sparse relative to the space they must cover. In practice, this sparsity hinders the model's ability to learn generalizable patterns and greatly increases the risk of overfitting, especially when annotated data are scarce.

A first challenge is **Speckle**, generating a granular interference that reduces the accuracy of pixel-level tasks like classification by decreasing spatial consistency [28]. Traditional techniques try to reduce its influence through multi-look processing, averaging multiple acquisitions at the cost of reducing the spatial resolution [27], or use specific filters such as Lee filter. Recent approaches use deep learning models such as autoencoders or U-Net architectures, trained to remove noise patterns from data while preserving structural details [8].

A second challenge is the data **multi-polarization**. Each pixel in PolSAR imagery is linked with a coherency matrix, resulting in multiple polarization channels. To process such data, transformation techniques are used to turn complex-valued information into a real-valued representation easier to handle for a neural network. Standard methods include [26, 4]:

- **Pauli decomposition**: separates scattering into odd-bounce, even-bounce, and volume components.
- **Freeman-Durden decomposition**: split the scattering mechanisms as surface, double-bounce, and volume scattering.
- **H-α decomposition**: relies on entropy and average scattering angle, and is particularly effective for land cover classification.

The importance of representation has been demonstrated by han et al. [12]. A feature selection algorithm combining fast filtering with an SVM classifier was used to select optimal features from multiple polarimetric decompositions, resulting in a higher Overall Accuracy (OA) than the CNN-based approach from [52], which was not using a tailored feature representation.

2.5 Data augmentation for image processing

Data Augmentation (DA) is a fundamental technique used in deep learning to expand the training dataset by generating new samples from existing ones. The diversity of the training set then increases without the need for expensive new acquisitions, improving the model's ability to generalize and reducing the risk of overfitting [38].

Common data augmentation techniques are the geometrical one, applying transformation such as rotation, flipping, translation to help the model become invariant to the rotation and position of the object in the image. This can be interpreted as observing the target from a different viewpoints. Photometric transformations, altering the color and the intensity in the image, are employed on optical image to simulate variation in lighting conditions and colors distortions [48]. More advanced techniques are based on occluding information, such as random erasing, cutout, and channel dropout. These techniques hide parts of the image or specific channels, making the model robust when such information is missing. Adding Gaussian or multiplicative noise to the image to simulate sensor imperfections, helping the model to focus on essential features rather than irrelevant details [31].

These techniques, developed for optical imagery, can be used on SAR images but are not as effective. However, the intrinsic characteristics of these data have resulted in the development of domain-specific augmentation methods. Huang et al. [17] highlighted that generative AI models offer a powerful solution for the augmentation of SAR images, addressing both the *quantity* and *quality* issues.

Increasing Data Quantity

In many SAR Applications, the number of training samples are limited to a few ranges of viewing angles, restricting the variability of the training dataset. Standard geometric transformation (rotation, flipping), cannot fully capture the strong dependence of SAR back-scatter. Unlike optical imagery, SAR data encodes the scattering properties of objects, including backscatter intensity, multiple-bounce interactions, and aspect-angle dependence. To overcome these limitations, Song et al. [40] developed an advanced generative approaches to synthesize novel target perspectives from existing SAR acquisitions. Generative models, such as adversarial autoencoders (AAEs) and generative adversarial networks (GANs), have demonstrated the ability to produce physically plausible SAR images across a range of azimuths and elevations.

Another approach to generate new data is the Optical-to-SAR (O2S) translation technique. The principle is to use optical imagery to fill missing information in SAR datasets or to generate synthetic SAR samples when direct acquisition is unavailable. The challenge with this approach is that a single optical image corresponds to multiple SAR images depending on the SAR geometry. To moderate this ambiguity, an unsupervised domain adaptation framework based on progressive transfer learning using generative adversarial networks (GANs) was proposed in [37]. Due to the huge difference between the input data, the model gradually aligns optical and SAR domains at three complementary levels: pixel space, latent feature space, and prediction space, enabling a more reliable transfer. Using the principle, the temporal shifting GAN (TSGAN) [34] introduces both temporal and multimodal information. The model takes as input an optical image at the desired timestamp, a SAR image acquired at a different time but with the same viewing geometry, and a change map derived from optical imagery between the two timestamps. A siamese encoder architecture is incorporated in both the generator and discriminator to enhance feature consistency. A change-weighted loss function is used, preventing overfitting on the input SAR data. TSGAN reduces the GAN hallucination phenomenon and achieves higher Structural Similarity Index Measure (SSIM) and Peak Signal to Noise Ratio PSNR compared to traditional translation methods by explicitly focusing on unchanged regions [17].

SAR image composition represents another strategy to increase the quantity. The principle is to synthesize new samples by seamlessly combining SAR patches of targets, SAR images of complex backgrounds, and other images logically [17]. This strategy has been used mainly to detect tasks in cluttered environments such as maritime environments, using a style-embedded augmentation network

to seamlessly integrate ship target slices into sea clutter [42]. Similarly, Kuang et al. [24] proposed a collaborative sample enhancement framework based on Pix2Pix background synthesis, which enables the creation of flexible, diverse, and high-quality samples that preserve both target fidelity and background realism.

Improving Data Quality

Besides expanding the dataset in terms of quantity, data augmentation tries to enhance the intrinsic quality of SAR images. A major limitation in SAR is the presence of speckle, which is multiplicative and spatially correlated, reducing interpretability and degrading the performance of pixel-level tasks. Several data-driven approaches have been implemented, Cycle-GAN-based methods consider despeckling as an image-to-image translation task, avoiding the need for a ground truth image [25]. Diffusion-based frameworks such as the SAR-DDPM [33] introduce a noise predictor conditioned on speckled input, while R-DDPM [16] incorporates overlapping region sampling during the inverse diffusion process to better preserve structures. Both approaches have been successful in removing speckle from the image without introducing artifacts, representing a step forward compared to traditional filters.

Another possible limitation of SAR datasets is the lack of complete polarimetric information. Full quad-pol acquisitions are not always acquired due to sensor limitations or acquisition costs. Recent strategies aim at reconstructing the missing polarimetric channels from partial measurements using deep learning. Song et al. [39] reconstruct full-pol covariance matrices from single-polarization images. In the same way, Deng et al. [5] generated pseudo quad-pol data from dual-pol acquisitions in order to improve urban damage assessment. Zhang et al. [51] designed a complex-valued dual-branch CNN to reconstruct pseudo quad-pol matrices from compact polarimetric data.

2.6 Self-supervised Learning (SSL)

Self-supervised Learning (SSL) is a machine learning paradigm in which models are trained on a *pretext task* tailored to exploit the inherent structure of the data. The central advantage of SSL is that it removes the dependency on manual annotations, allowing models to leverage massive unlabeled data. In the field of SAR, where annotated datasets are scarce, this approach is particularly relevant. The representation learned through SSL can be fine-tuned on specific tasks such as classification, segmentation, detection, and often achieves equivalent or better results in comparison to Fully Supervised Learning (FSL) [21].

SSL methods can be categorized into three families: autoassociative learning, contrastive learning, and non-contrastive learning.

Autoassociative learning, introduced in 1991 by Mark Kramer [22], is a technique based on the principle of reconstructing missing or altered parts of the inputs. This approach can be seen as an extension of the Principal Component Analysis (PCA) to nonlinear transformation. The model's objective during training is to reconstruct the original input from a corrupted version while minimizing a reconstruction loss, often Mean Square Error (MSE). The alteration techniques employed include partial occlusion, small geometrical transformations, or patch masking. Autoassociative SSL has been successfully applied in domains such as medical imaging and remote sensing, where it produces semantically meaningful and noise-robust representations [14, 7].

Contrastive learning, relies on the principle of similarity discrimination. Multiple augmented views of the same instance through techniques such as cropping, noise injection, or color distortion are generated. An augmented image and the original form a *positive pairs*, while views from different instances are treated as *negative pairs*. The model learn to maximize the similarity in the embedding space for the positive pairs and minimize it for the negative ones. Loss functions such as InfoNCE or Triplet Loss [35, 36] are commonly used. This results in embeddings where semantically similar inputs are clustered together and dissimilar ones are scattered. A strength of contrastive learning is that its performance scales with strong and diverse data augmentation techniques.

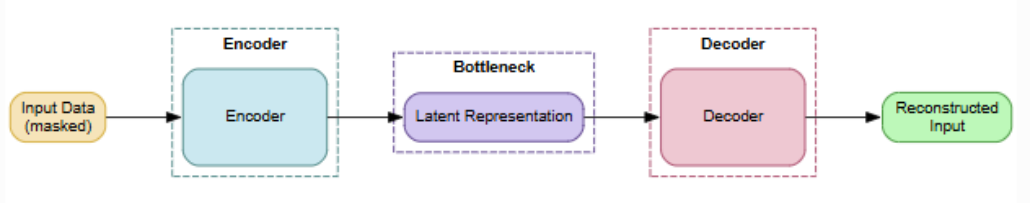


Figure 4 – Schematic illustration of the Masked Autoencoder principle, adapted from [14].

Non-contrastive learning, unlike contrastive approaches, removes the constraint for negative samples. Instead, it encourages the model to produce different augmented views of the same input to produce similar embeddings, while using internal mechanisms to prevent representational collapse. An example of application of this method is the framework Bootstrap Your Own Latent (BYOL), proposed by Grill et al. [10]. They used two networks, an online and a target network learning from one another. An augmented view of an input is passed through the online network, which is trained to predict the representation produced by the target network on a different view of the same input. The target network's weights are updated as a moving average of the online network, ensuring training stability.

Masked Autoencoder (MAE) for Pretraining

MAE, introduced by He et al. [14], representes an application of the autoassociative approach to images. The key principle is to randomly mask a large portion of image patches (e.g., 75%) and train the model to reconstruct the missing pixels from the remaining, visible ones. Figure 4 illustrates the principle. The original architecture is asymmetrical: a lightweight encoder processes only the unmasked patches, extracting compact feature representations, while a more powerful decoder reconstructs the original image, including the masked regions. This design enables the model's encoder to focus on learning meaningful, transferable representations rather than trivial low-level statistics.

In the SAR image field, MAE has attracted significant attention as it offers the possibility to leverage huge unlabeled datasets However, considering the SAR-specific challenges such as speckle, geometric distortions, and complex-valued polarimetric channels, adaptations of the MAE framework can be useful to achieve better performance. For instance, Wang et al. [44] proposed the Feature-Guided Masked Autoencoder (FG-MAE), where the model reconstructs higher-level features instead of raw pixels. They use the Histogram of Oriented Gradients (HOG) as reconstruction targets to be less sensitive to speckle. Experimental results show that FG-MAE significantly improves the performance of downstream SAR tasks and scales effectively to larger datasets.

3 Materials

This section details the dataset and preprocessing steps implemented. We first introduce the Pol-InSAR-Island benchmark dataset [15], detailing its acquisition characteristics and the provided land cover annotations. Subsequently, we describe the data processing pipeline, including patch extraction, the creation of multiple input representations from the PolSAR coherency matrix, and the different normalization strategies employed in our experiments.

3.1 Pol-InSAR-Island Dataset

The dataset employed in this study is the *Pol-InSAR-Island* benchmark, specifically designed for multi-frequency Polarimetric Interferometric SAR (Pol-InSAR)-based land cover classification [15]. It addresses the scarcity of publicly available, comprehensively annotated datasets in this domain. To ensure reproducibility and facilitate fair comparisons across methods, the dataset includes a predefined train–test split.

The data were acquired over the East Frisian island of Baltrum, Germany, using the German Aerospace Center (DLR) airborne F-SAR system. Two frequency bands are provided: **S-band** and **L-band**. Leveraging multiple frequencies enhances classification performance by providing complementary scattering signatures, which improve discrimination between classes that are otherwise spectrally similar at a single frequency.

To cover the full extent of the island, two separate flight passes (Flight Path 1 (FP1) and Flight Path 2 (FP2)) were conducted. The resulting imagery dimensions are 3616 × 2502 pixels for FP1 and 3616 × 2540 pixels for FP2. Data are delivered as geocoded 6 × 6 coherency matrices sampled on a 1 m × 1 m grid. The flight paths are schematized in Figure 5.

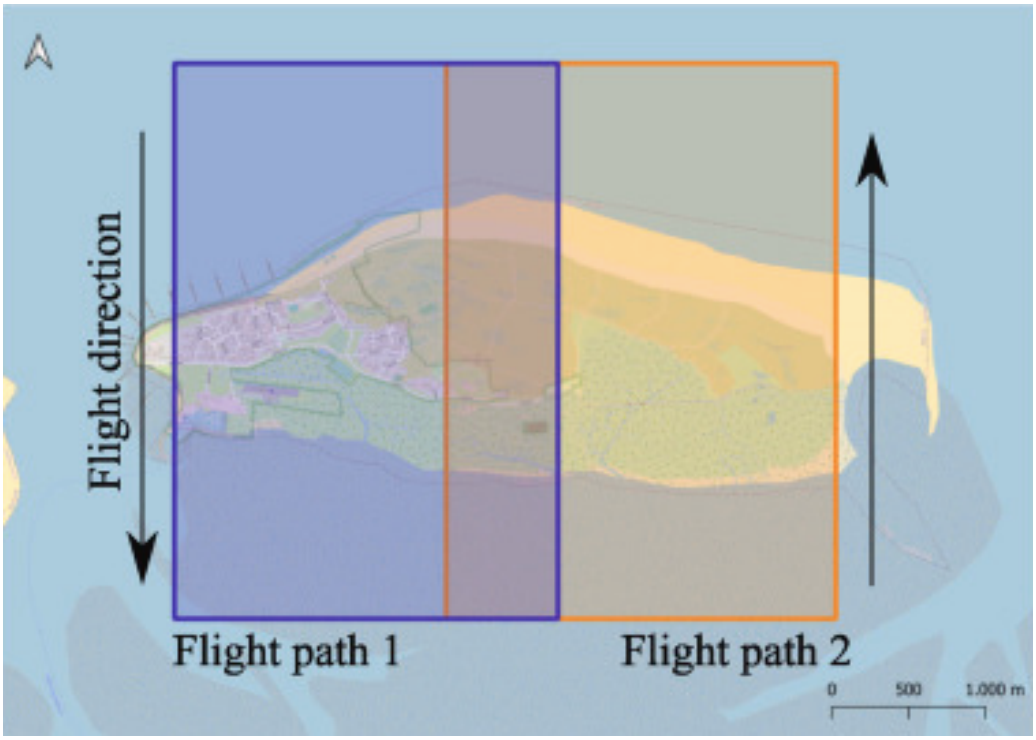


Figure 5 – Schematic representation of the F-SAR flight paths used for data acquisition. Adapted from [15]

```
Data structure:
.
├── data                        # Multi-frequency Pol-InSAR data
│   ├── FP1                  # Flight path 1
│   │   ├── L                   # L-band (frequency)
│   │   │   ├── T6              # Geocoded Pol-InSAR data components of the 6 x 6 coherency matrix
│   │   │   ├── incidence.bin # Incidence angle [rad]
│   │   │   ├── kz_*.bin  # Vertical wavenumber for vv, hv, vh and vv polarization [rad/m]
│   │   │   └── pauli.bmp     # Pauli RGB image of the master scene
│   │   └── S                   # S-band (frequency)
│   │       └── ...
│   └── FP2                  # Flight path 2
│       └── ...
└── label                       # Land cover labels
    ├── FP1                  # Flight path 1
    │   ├── label_train.bin    # Geocoded training and test label files (*.bin + *.hdr)
    │   ├── label_test.bin
    │   ├── ...
    └── FP2           # Flight path 2
        └── ...
```

Figure 6 – Dataset file structure and decomposition of coherency matrices. Adapted from [15].

3.2 Dataset Processing

The dataset is distributed in ENVI format, with flat-binary raster files (.bin) accompanied by ASCII header files (*.hdr). The real and imaginary components of the diagonal elements and upper triangle of the 6×6 coherency matrix are stored in separate files (`T11.bin`, `T12_real.bin`, `T12_imag.bin`). Figure 6 illustrates the file structure. The full 6×6 coherency matrix can be understood as two distinct 3×3 coherency matrices T_1 and T_2.

As this work focuses on PolSAR imagery, only T_1 was considered for feature extraction and further analysis. Since neural networks typically require real-valued inputs, the complex-valued matrix was converted into a real-valued representation by separating real and imaginary parts of the off-diagonal terms while keeping diagonal terms unchanged. This preserves the Hermitian structure while producing a consistent set of real-valued channels usable by the network.

Each pixel is annotated with one of 12 land cover classes: 0 – Unassigned, 1 – Tidal flat, 2 – Water, 3 – Coastal shrub, 4 – Dense high vegetation, 5 – White dune, 6 – Peat bog, 7 – Grey dune, 8 – Couch grass, 9 – Upper salt marsh, 10 – Lower saltmarsh, 11 – Sand, and 12 – Settlement.

As mentioned already, the train-test split of the dataset has already been performed using a 'maze'-based spatial partitioning method. This ensures that the test set contains only spatially distinct, unseen areas while preserving similar class distributions across sets. Figure 8 shows the original segmentation map alongside the derived train and test partitions. Table 3.2 summarizes the class distributions, while Figure 7 highlights class imbalance in the training set, with several underrepresented categories (e.g., peat bog, couch grass).

Table 2 – Class Distribution

Category	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Class 10	Class 11	Class 12
Train	6.78%	21.48%	5.39%	3.45%	3.66%	0.93%	13.92%	1.66%	12.05%	9.33%	9.27%	12.07%
Test	8.73%	18.67%	4.61%	2.49%	2.85%	1.21%	14.35%	1.92%	12.46%	8.26%	10.98%	13.47%

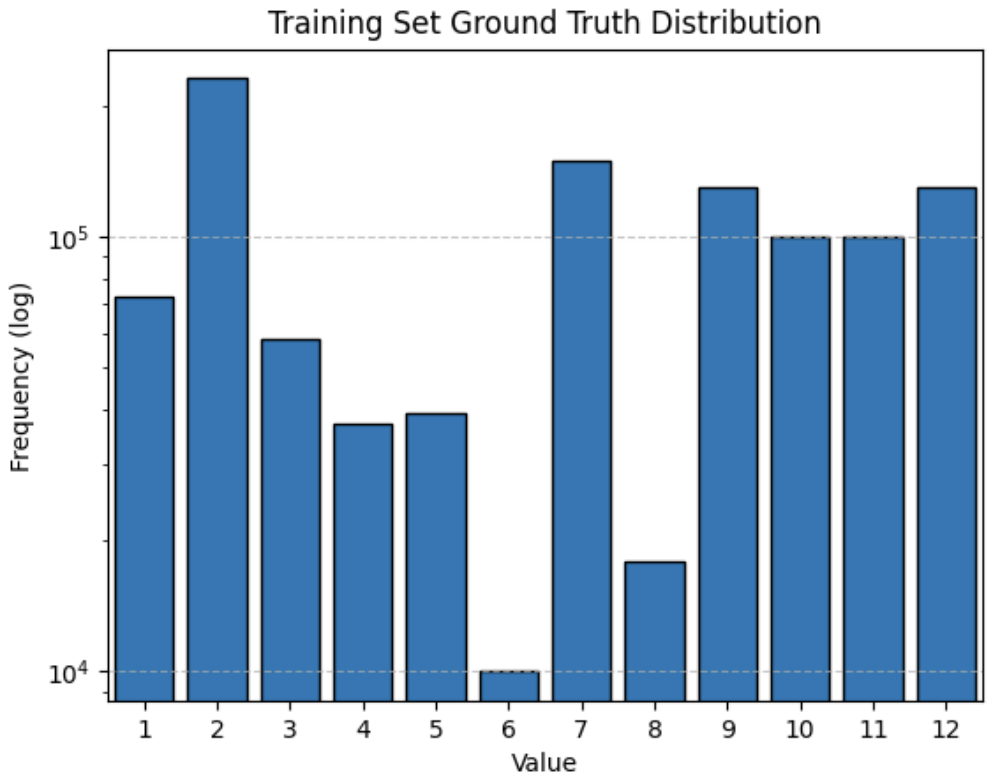


Figure 7 – Histogram of the ground truth distribution in the training set

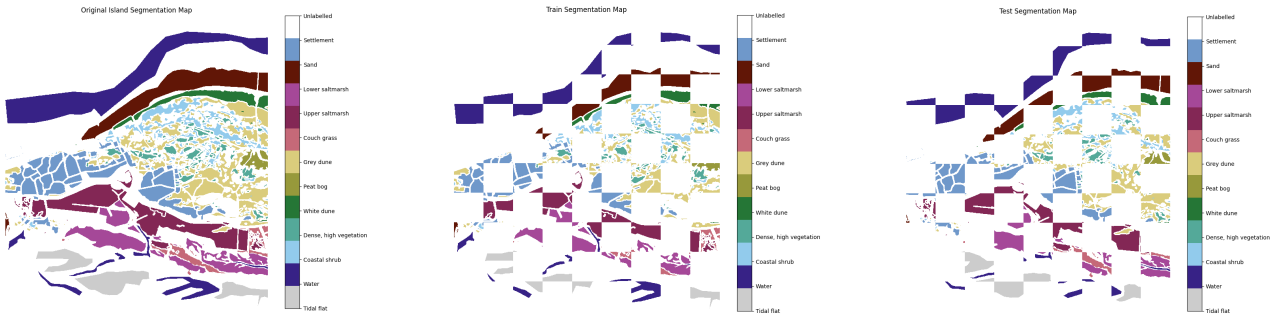


Figure 8 – Original ground-truth segmentation map (left), and train (middle) / test (right) partitions based on the "maze" split.

The original large images were subdivided into non-overlapping patches of 256×256 pixels, which is the patch size used in the original dataset definition. For FP1 (3616×2502 pixels), after excluding 382 unusable top pixels, this yields:

$$\left\lfloor \frac{3616 - 382}{256} \right\rfloor \times \left\lfloor \frac{2502}{256} \right\rfloor = 12 \times 9 = 108 \tag{15}$$

patches, evenly split between the train and test sets. Applying the same procedure to FP2 (3616×2540 pixels) also results in 108 patches, giving a total of 216 patches.

To investigate the influence of input representation on the model's performance, several transformations of the coherency matrix were implemented. The general idea was to assess the framework's performance as fast as possible by implementing simple and fast transformations. Additionally, the extended log-ratio representation helps preserve as much of the scattering mechanism as possible while helping the model generalize more easily. Here is a list of the implemented transformations:

- **Grayscale representation:** Only the T_{11} coefficient is preserved and replicated across three channels, producing a grayscale-like input while discarding cross-polarization terms :

$$\mathbf{T} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \quad \rightarrow \quad \mathbf{X} = \begin{bmatrix} T_{11} \\ T_{11} \\ T_{11} \end{bmatrix} \tag{16}$$

- **Diagonal-only representation:** The three diagonal terms are retained, providing information on scattering power in co- and cross-polarized channels:

$$\mathbf{T} \rightarrow \mathbf{X} = \begin{bmatrix} T_{11} \\ T_{22} \\ T_{33} \end{bmatrix} \tag{17}$$

- **Log-diagonal representation:** To reduce the dynamic range of diagonal values, a logarithmic transformation is applied:

$$\mathbf{T} \rightarrow \mathbf{X} = \begin{bmatrix} \log(T_{11}+1) \\ \log(T_{22}+1) \\ \log(T_{33}+1) \end{bmatrix} \tag{18}$$

- **Real–Imag decomposition:** All diagonal terms are preserved, and each complex off-diagonal element is split into real and imaginary parts. This results in a compact 9-channel real-valued representation:

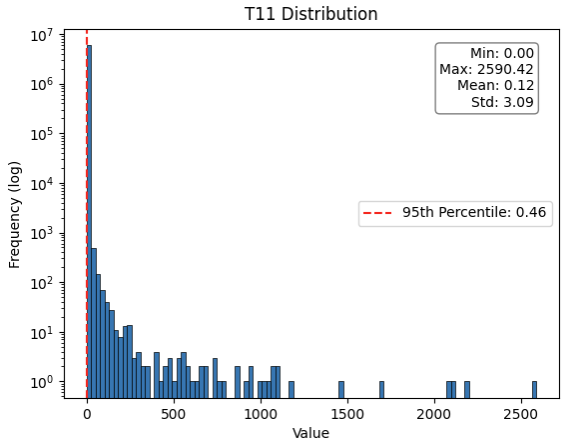
$$\mathbf{T} \rightarrow \mathbf{X} = \begin{bmatrix} T_{11} \\ Re(T_{12}) \\ Re(T_{13}) \\ Im(T_{12}) \\ (T_{22}) \\ Re(T_{23}) \\ Im(T_{13}) \\ Im(T_{23}) \\ T_{33}) \end{bmatrix}, \tag{19}$$

- **Extended log-ratio representation:** A richer representation is obtained by combining logarithms of diagonal and magnitude terms with normalized cross terms, capturing both intensity and relative phase information:

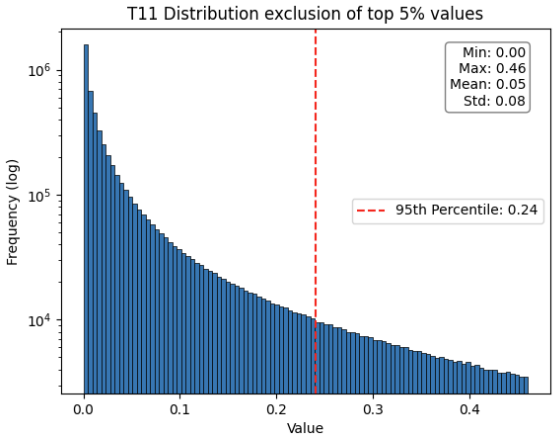
$$\mathbf{T} \rightarrow \mathbf{X} = \begin{bmatrix} \log(T_{11}) \\ \log(M(T_{12})) \\ \log(M(T_{13})) \\ \log(T_{22}) \\ \log(M(T_{23})) \\ \log(T_{33}) \\ Re(T_{12})/M_{12} \\ Im(T_{12})/M_{12} \\ Re(T_{13})/M_{13} \\ Im(T_{13})/M_{31} \\ Re(T_{23})/M_{23} \\ Im(T_{23})/M_{32} \end{bmatrix}, \tag{20}$$

  where $M(T_{ij})$ is the magnitude of the complex element T_{ij}.

In addition to representation choices, normalization of the transformed channels is crucial to stabilize training. Two strategies were implemented:



Channel T_{11} histogram



Channel T_{11} histogram with exclusion of the top 5% values

Figure 9 – Histogram of the data distribution among the samples showing the first value of the coherency matrix (T11). The right plot excludes the top 5% of values.

- **Min–max scaling**, applied either globally or after excluding the top 5% of values. In the latter case, two approaches were tested:
  1. excluding the extreme values when computing $I_{\min}$ and $I_{\max}$, but leaving the data unchanged;
  2. excluding the extremes and additionally cropping all pixel values above the 95th percentile P_{95}. The clipping operation can be written as:

$$I' = \min(I, P_{95}), \tag{21}$$

     where I is the original pixel intensity and I' is the clipped value.
- **Standardization**, centering each channel to zero mean and unit variance. Similar to min–max, both global and 5%-trimmed versions were tested, with and without the optional clipping step of Equation 21.

The decision to exclude the top 5 % of values comes from the distribution; a lot of outlier values are present across the different channels, making the generalization harder for the model. The histograms Figure 9 show that the majority of the values are scattered around 0, mean=0.12, and the 95th percentile is equal to 0.46, when the max value is more than 2500. This small fraction of pixels would have dominated during the scaling.

4 Deep learning-based Methodology

This section presents the deep learning framework developed for this study. We first describe the U-Net and Masked Autoencoder (MAE) architectures, along with the weight-transfer strategy used to link them. Next, we detail the two-stage training methodology, which involves a self-supervised pretraining phase and a subsequent supervised fine-tuning stage. Finally, we outline the data augmentation techniques applied during training and the evaluation metrics used to assess the final segmentation performance.

4.1 Deep Learning model used

To perform the data semantic segmentation, a U-Net architecture was used. We chose this architecture as it is a widely used and trusted framework for image segmentation. It was primarily designed to improve medical image analysis, making tumor detection in MRI scans easier. The U-Net architecture has since been successfully applied to other fields. The architecture consists mainly of two parallel branches: an encoder progressively reducing the spatial dimensions of the feature maps, and a decoder restoring them to their original resolution. This process of reduction and expansion compresses the information into a latent space, where the representation is more compact and optimal for the task. By combining low-level features from the contracting path with high-level features from the expansive path, the network is able to segment images at multiple scales. Skip connections, from the encoder to the decoder, transfer spatial details, preserving fine-grained information and improving segmentation accuracy. Figure 10 illustrates the model implemented with three input channels. The implementation used is different from the original, because a batch normalization was added to speed up the computation [18]. The number of input channels was varied in different experiments depending on the input data representation used.

The encoder is composed of blocks built with two consecutive convolutional layers with 3×3 pixel kernel filters, followed by a batch normalization and a Rectified Linear Unit (ReLU) activation function. These double convolution blocks progressively increase the number of filters while capturing increasingly abstract features. Between each encoder block, in order to downsample the feature maps, a 2×2 max pooling layer is applied, reducing the spatial resolution and expanding the receptive field. After several double blocks, the bottleneck contains the largest number of filters, enabling the extraction of the most abstract representation of the input data.

The decoder uses an upsampling layer to upscale the feature maps. This operation can be performed using bilinear upsampling or transposed convolutions. The upsampling operation chosen for our implementation is a transposed convolution. At each stage, the upsampled features are concatenated with the corresponding feature maps from the encoder through skip connections and refined by another double convolutional block. A final 1×1 convolutional layer and a softmax function are applied to assign a probability to each pixel, enabling semantic segmentation of pixels based on the highest predicted probability.

The current U-Net architecture mentioned was developed to learn through FSL; however, we want to use a MAE that can be trained through SSL and then transfer its encoder weights to the U-Net encoder. The schema Figure 11 illustrates this idea. But, in order to be able to transfer the weight from one encoder to another, both should have the same structure. This constraint leads us to design the following MAE Figure 12. The encoders are the same, and the decoding part is composed of upsampling layers, transpose convolution, and double convolutional blocks. For the final layer, no activation function was added to preserve the wide range dynamic of the coherency matrix.

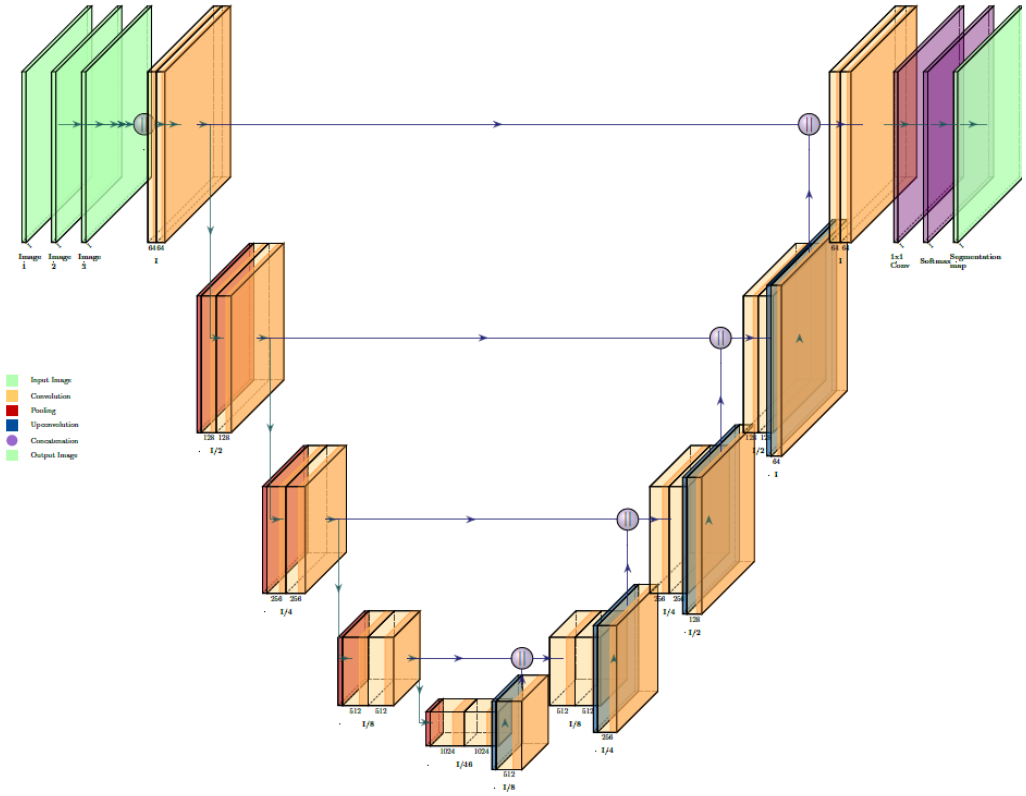


Figure 10 – U-Net architecture implemented to perform semantic segmentation. The schema was made from an adaptation of the code [19]

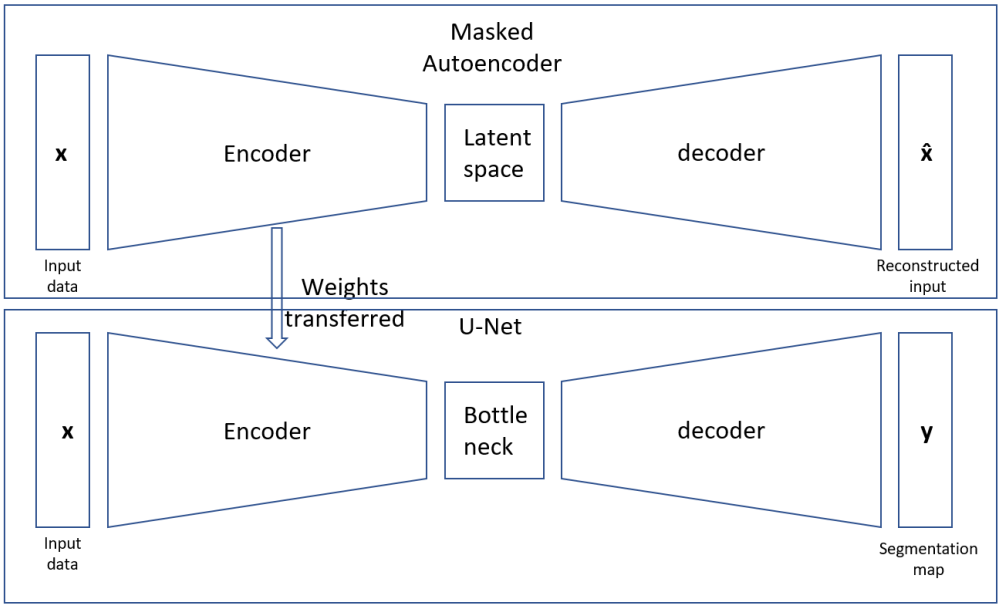


Figure 11 – Schema showing the weights transfer from the Masked Autoencoder (MAE) to the U-Net model.

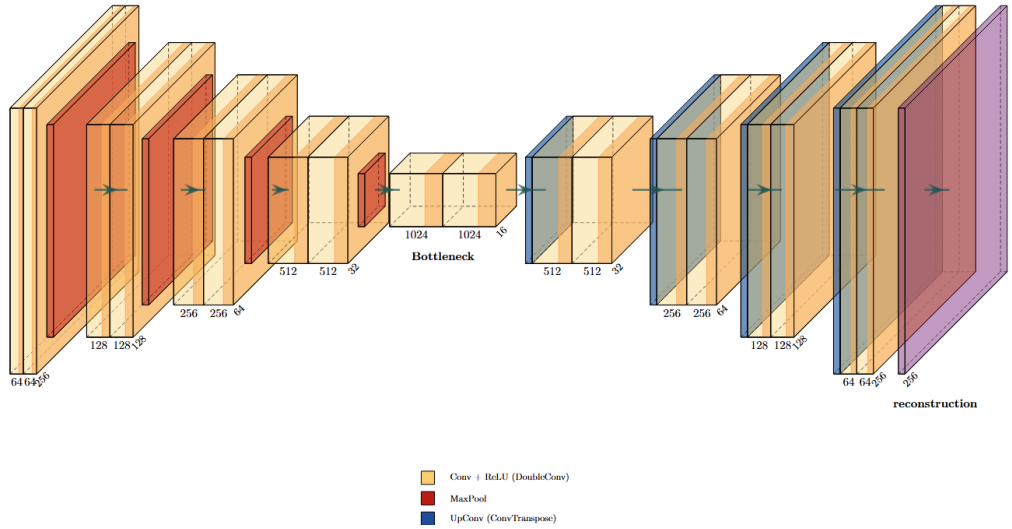


Figure 12 – Masked Autoencoder architecture with similar encoder as the U-Net model implemented to perform semantic segmentation. The schema was made from an adaptation of the code [19].

4.2 Training methodology

In the previous sections, the processing steps applied to the dataset and the architecture of the different models have been described. The training methodology is now detailed in this subsection.

Self-supervised Learning (SSL)

For the pretraining stage, we opted for a MAE. The MAE framework was chosen because it provides a simple and effective SSL task, whereas alternative SSL methods for SAR often involve complex designs that are difficult to implement. The choice of the loss plays a crucial role in the model's ability to learn from the dataset, as it determines the type of backpropagation performed during training. In case of regression, reconstruction of the input data, the MSE, or L^2-norm error, is used and defined as :

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{22}$$

where y_i is the reference value and $\hat{y}_i$ the model prediction, and N is the number of samples.

Masking was performed by randomly placing a fixed number of non-overlapping square masks of size 32×32 on 256×256 patches. The mask applied is different for each channel. Masked pixels were set to 0, and reconstruction loss was computed only over the masked regions. Figure 13 shows an example of the masking process applied with a masking rate of 40%. This forces the model to understand the context, structure, and relationships of the surrounding data points rather than memorizing trivial patterns. Empirical tests confirmed that restricting the loss to masked pixels yielded superior performance compared to calculating it over the entire image.

The model was optimized using the Adam optimizer with an initial learning rate of 1×10^{-4}. A cosine annealing scheduler was employed to gradually decrease the learning rate to a minimum of 1×10^{-6} over the course of training. Since no labels were required for SSL pretraining, the dataset was split into training (70%), validation (15%), and test (15%) sets independently of the predefined maze split. Based on empirical evaluation, the batch size was set to 24, and the model was trained for 150 epochs.

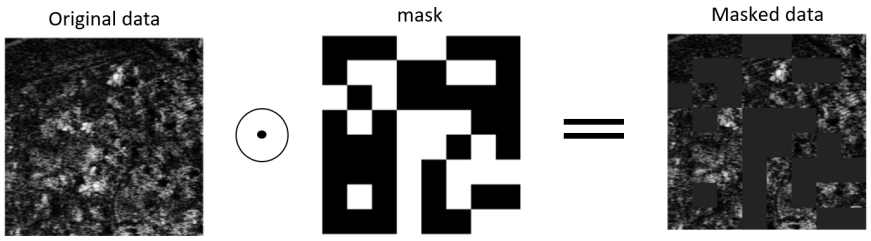


Figure 13 – Masking strategy for SSL: 32×32 squares are randomly placed on a 256×256 patch with a masking rate of 40%.

Fully Supervised Learning (FSL)

For the semantic segmentation task, we utilized the Cross Entropy (CE) (Cross-Entropy) loss function, which is the standard choice for multi-class, pixel-level classification problems. It is defined as:

$$\text{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij}), \tag{23}$$

where N is the number of pixels, C is the number of classes, y_{ij} is the one-hot encoded ground-truth label (1 if pixel i belongs to class j, 0 otherwise), and p_{ij} is the model's predicted probability.

The models were fine-tuned for a maximum of 300 epochs, with an early stopping patience of 10 epochs. The learning rate was initialized to 1×10^{-3} and coupled with a cosine annealing scheduler that reduced it to a minimum of 1×10^{-5}. The optimizer used was Adam, with a batch size of 24. For this stage, we followed the dataset's predefined maze partition: 40% for training, 10% for validation, and 50% for testing.

4.3 Data augmentation implemented

Given the limited number of training samples (54 patches for training/validation and 54 for testing), data augmentation was applied to improve generalization and mitigate the class imbalance during the segmentation training. The following strategies were implemented:

- **Geometric transformations:** horizontal flip, vertical flip, and random 90° rotations.
- **Noise injection:** additive and multiplicative Gaussian noise to mimic acquisition variability.
- **Random cropping and resizing:** crops covering 60–100% of the image were randomly sampled and then rescaled to the original size via bilinear interpolation.
- **Channel drop:** randomly zeroing out one or more input channels to enforce robustness to missing information.
- **CutMix:** patch replacement with 128×128 or 64×64 regions cut from other training samples.

4.4 Metrics for performance evaluation

To assess segmentation quality, we utilized the Intersection over Union (IoU) metric, also known as the Jaccard index [20], which effectively accounts for both false positives and false negatives. For a predicted mask A and reference data mask B, IoU is calculated as:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}. \tag{24}$$

From this, we can compute the mean Intersection over Union to provide a comprehensive performance measure across all classes. This metric calculates the average of the IoU scores for each class, offering a single, robust score that summarizes the overall segmentation quality. By averaging across classes, mIoU gives a more balanced assessment of the model's ability to segment both common and rare categories. It is defined as:

$$\text{mIoU} = \frac{1}{C} \sum_{c=1}^{C} \text{IoU}_c, \tag{25}$$

where C is the total number of classes.

This approach ensures a fair comparison with the baseline results from the original dataset publication [15].

Table 3 – Acronyms defining the network training process.

GS	Grayscale representation
3T	Diagonal-only representation
log(3T)	Log-diagonal representation
9T	Real–Imag decomposition
12T	Extended log-ratio representation
raw	Means that no normalization was used.
Mm	Min–max scaling
stand	Standardization
5%	Means that the normalization was computed while excluding the top 5% of values.
5%C	Means that the normalization was computed while excluding the top 5% of values and a cropping was applied on the data.

5 Results

In this section, we evaluate the impact of different input representations, normalization strategies, and data augmentation techniques using the training methodology described in Section 4.2. Furthermore, we compare the performance of the proposed SSL-based framework against a baseline model trained from scratch without pretraining. Table 3 summarizes the acronyms used throughout this section to refer to the various representations and normalization strategies.

All experiments were conducted on NVIDIA A100-SXM4-40GB GPUs. Due to the high variance typically observed in deep learning training, each configuration was repeated 10 times using repeated random splits: a train–validation–test split for the reconstruction task and a train–validation split for the segmentation task. This approach provides a more robust estimate of variability. For models requiring a pretraining step, the entire pretraining–finetuning pipeline was repeated 10 times; after each pretraining run, the resulting MAE encoder was used once to initialize a U-Net and then discarded.

The following subsections present the impact of the different preprocessing steps, followed by a comparison between the pretrained and non-pretrained models.

5.1 Impact of the Masking Rate

The masking ratio is a key hyper-parameter for MAE. To set an initial value, we qualitatively examined reconstructions produced under a range of masking ratios. Figure 14 displays results obtained with the extended log-ratio representation and full-dataset standardization at several masking ratios. A 40% mask preserves fine structural detail while still compelling the model to infer missing regions; lower ratios lead to overly trivial reconstructions. Consequently, we adopted a 40% masking ratio for all subsequent experiments.

A quantitative evaluation was then conducted to assess the impact of this choice. The full training pipeline (including pre-training of the MAE, transfer of encoder weights, and downstream fine-tuning) was repeated ten times. For each run, the average Intersection over Union (IoU) and the associated standard deviation were computed. To reduce variance, the analysis focused on two configurations that had previously shown the best trade-off between mean performance and variability:

- Log-diagonal representation with min–max 5% trimmed normalization
- Extended log-ratio representation with standardized 5% trimmed normalization

As illustrated in Figure 15, quantifying the optimal masking rate is not straightforward due to the high variance across runs. Nevertheless, the results indicate that a masking rate between 40% and 45%

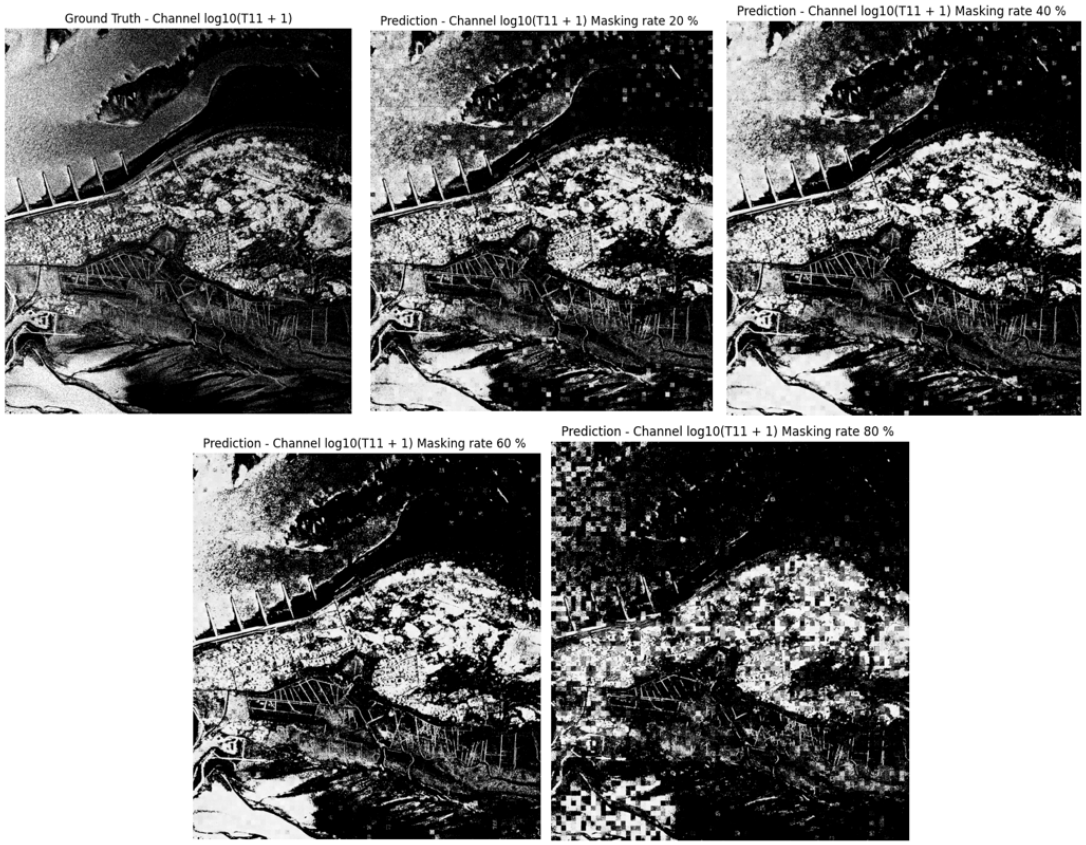


Figure 14 – Visual comparison of the masking rate impact on the reconstruction task.

provides consistently above-average performance for both tested configurations. If the constraint of maintaining a high masking rate is relaxed, masking rates between 20% and 25% also yielded reasonably good results.

5.2 Impact of the Representation on the Model Performance

To evaluate the impact of the input representation, trainings were run for every representation–normalization combination described in Section 4. For each representation, we calculated the mean and standard deviation of the IoU across all normalizations. The results are summarized in Table 4.

The weakest performance was obtained with the grayscale representation, with mean IoU values of 11.78% (FSL) and 15.49% (SSL). This confirms that retaining only a single diagonal term discards most of the polarimetric information, making the representation poorly suited for segmentation tasks.

At the other end of the spectrum, the extended log-ratio representation achieved the best results in both FSL and SSL, with 23.79% and 32.19% respectively. This representation preserves both diagonal and normalized cross terms while applying logarithmic scaling, improving information content and numerical stability. The second-best performance was obtained with the log-diagonal representation, reaching 21.91% (FSL) and 30.51% (SSL). Despite discarding off-diagonal terms, it outperformed the real–imag decomposition (19.29% FSL, 23.46% SSL), suggesting that appropriate rescaling can be more beneficial than retaining raw correlation terms.

Across all representations, SSL consistently improved performance compared to FSL. The largest relative gains were observed for log-diagonal (+39%) and extended log-ratio (+35%), while grayscale still benefited with a +31% improvement.

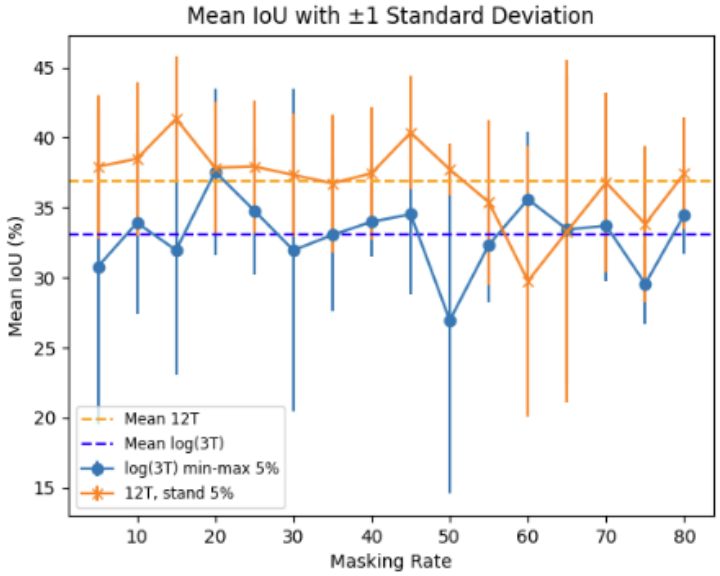


Figure 15 – Average IoU as a function of the masking rate. Error bars indicate one standard deviation across 10 independent simulations. Results are shown for the log-diagonal (min–max 5%-trimmed) and extended log-ratio (standardized 5%-trimmed) representations.

Finally, SSL also reduced variability. For example, the log-diagonal standard deviation decreased from 9.15% (FSL) to 4.25% (SSL, -54%), and grayscale dropped from 10.35% to 5.26% (-49%). This demonstrates that SSL not only increases mean performance but also stabilizes results across runs.

		12T	9T	log(3T)	3T	GS
FSL	mean	23.785	19.288	21.906	17.22	11.782
	std	9.15	7.7516	9.071	8.09	10.35
SSL	mean	32.192	23.455	30.505	22.892	15.494
	std	4.253	5.099	5.39	4.9558	5.258

Table 4 – Average IoU (mean ± std) obtained for each representation, aggregated over all normalization strategies.

5.3 Impact of the Normalization on the Performance

To evaluate the impact of normalization, the results were grouped by normalization technique, and the mean and standard deviation of the IoU were computed across all representations. These results are reported in Table 5.

The weakest normalization strategy was min–max scaling on all data. In FSL, it achieved only 3.50% ± 4.68%, compared to 13.28% ± 10.34% with raw data. SSL also suffered under full min–max, reaching 19.60% ± 4.80% compared to 22.85% ± 5.55% for raw data. This highlights how outliers strongly distort min–max scaling.

The best performance was obtained with min–max 5%-trimmed plus clipping (35.10% ± 6.07%), closely followed by standardization 5%-trimmed plus clipping (34.55% ± 5.58%). While both strategies reached similar average values, the latter was slightly more stable due to its lower variability.

Intermediate results were observed for standardization without clipping (23.03% FSL, 27.39% SSL) and min–max 5%-trimmed (22.27% FSL, 26.45% SSL). These show that trimming outliers consistently improves performance, even without clipping.

Across all normalization strategies, SSL improved results relative to FSL. Gains were most pronounced for weaker normalizations, such as min–max (+16.1%) and raw input (+9.6%). For the best strategies, the gain was smaller but still positive, e.g., +5.0% for standardization 5%-trimmed plus clipping. SSL also reduced variability: raw normalization decreased from 10.34% to 5.55% (-46%), and standardization 5%-trimmed plus clipping from 12.42% to 5.58% (-55%).

		raw	Mm	Mm 5%	Mm 5%C	stand	stand 5%	stand 5%C
FSL	mean	13.278	3.503	22.266	19.090	23.030	24.676	29.553
	std	10.338	4.678	10.575	15.578	10.572	7.980	12.417
SSL	mean	22.846	19.600	26.446	35.097	27.385	21.260	34.550
	std	5.550	4.804	6.487	6.070	5.726	7.675	5.575

Table 5 – Average IoU (mean ± std) obtained for each normalization method, aggregated over all representations.

5.4 Impact of Data Augmentation on the Model Performance

To evaluate the role of data augmentation (DA), we conducted experiments using the extended log-ratio representation with standardized normalization. Six augmentation methods were tested: horizontal flip, vertical flip, single-channel dropout, random crop with resizing, CutMix with 128×128 patches, and Gaussian noise injection $\mathcal{N}(0, 0.05^2)$. In addition, the second flight path (FP2) was used as a form of directional augmentation. Results without augmentation (FSL and SSL) served as reference. Tables 6 and 7 summarize the results.

The weakest augmentation was Gaussian noise injection, which reduced mean IoU to 15.71% with severe instability (CV of 1.38). This degradation was especially strong for water segmentation, which dropped from 69.24% (SSL) to 35.30%.

The best augmentation method was CutMix with 128×128 patches, which achieved 34.87% mean IoU. This nearly matched SSL performance (36.93%) while operating within the FSL framework. CutMix was particularly effective for homogeneous classes such as settlements (86.80% ± 4.08%) but underperformed on complex classes like lower salt marsh (31.80% compared to 71.80% in SSL).

Intermediate results were obtained with random crop (28.24%, +6.53 percentage points over FSL baseline), which shows balanced performance across different land cover types while maintaining reasonable spatial coherence. Horizontal and vertical flips gave smaller gains, with vertical flip (26.56%) outperforming horizontal flip (22.61%). Single-channel dropout and FP2 augmentation produced marginal improvements (22.22% and 23.70%), with mixed results across classes.

As in the previous sections, SSL provided the strongest overall improvement, boosting mean IoU from 21.71% (FSL) to 36.93% (+15.22 points). It also reduced variability, with CV decreasing from 1.22 to 0.66. The effect was consistent across most classes, including large gains for water, grey dunes, and settlements. However, rare classes such as peat bog and couch grass remained poorly segmented, with IoU values below 1.1% regardless of augmentation.

Table 6 – Performance metrics by class for simple data augmentation (Mean IoU ± Std). Summary statistics are reported at the bottom.

Class	Horizontal Flip	Vertical Flip	CutMix 128	Random Crop
1. Tidal flat	7.82 ± 13.03	15.28 ± 27.28	46.68 ± 33.16	41.59 ± 30.52
2. Water	49.05 ± 20.47	62.08 ± 9.76	73.22 ± 8.66	66.04 ± 23.76
3. Coastal shrub	17.74 ± 13.02	25.75 ± 12.29	25.04 ± 18.18	20.79 ± 13.38
4. Dense vegetation	18.64 ± 12.60	12.40 ± 9.84	30.47 ± 7.35	19.31 ± 9.72
5. White dune	1.10 ± 1.70	0.35 ± 0.43	0.52 ± 0.78	0.55 ± 1.26
6. Peat bog	0.17 ± 0.31	0.14 ± 0.24	0.36 ± 0.56	0.28 ± 0.37
7. Grey dune	36.10 ± 20.50	33.43 ± 21.89	54.76 ± 9.74	33.04 ± 18.87
8. Couch grass	0.10 ± 0.20	0.30 ± 0.68	0.10 ± 0.22	0.02 ± 0.00
9. Upper salt marsh	41.00 ± 28.33	49.43 ± 16.46	49.48 ± 19.65	44.33 ± 22.28
10. Lower sal marsh	23.70 ± 22.55	41.70 ± 17.79	31.80 ± 27.17	25.92 ± 20.10
11. Sand	7.63 ± 10.39	7.36 ± 5.91	19.15 ± 15.02	13.03 ± 16.48
12. Settlement	68.23 ± 31.32	70.46 ± 19.48	86.80 ± 4.08	73.96 ± 25.28
Mean of means	22.61	26.56	34.87	28.24
Mean of std	16.23	11.84	12.05	15.17
Std of means	21.86	24.79	28.39	24.67
Mean CV	0.91	0.78	0.78	0.78

Table 7 – Performance metrics by class for targeted data augmentation and references (Mean IoU ± Std). Summary statistics are reported at the bottom.

Class	SSL	FSL	Channel Drop	Gaussian Noise	FP2
1. Tidal flat	27.47 ± 23.77	19.57 ± 27.29	14.10 ± 20.40	13.26 ± 25.79	9.29 ± 22.15
2. Water	69.24 ± 7.23	51.19 ± 26.75	45.52 ± 30.15	35.30 ± 35.91	56.07 ± 21.20
3. Coastal shrub	30.32 ± 14.63	13.60 ± 13.03	13.82 ± 15.62	14.18 ± 11.41	26.81 ± 12.40
4. Dense vegetation	21.12 ± 12.83	16.30 ± 13.89	19.58 ± 13.43	8.00 ± 9.84	8.82 ± 8.83
5. White dune	1.47 ± 3.87	0.46 ± 0.93	0.36 ± 0.90	0.41 ± 0.78	2.84 ± 4.26
6. Peat bog	0.08 ± 0.00	0.10 ± 0.17	0.99 ± 2.26	0.21 ± 0.36	0.59 ± 1.34
7. Grey dune	60.40 ± 8.53	29.20 ± 21.01	30.80 ± 25.10	20.06 ± 22.50	42.41 ± 16.81
8. Couch grass	0.11 ± 0.14	0.20 ± 0.60	0.39 ± 0.78	0.04 ± 0.10	0.81 ± 1.20
9. Upper salt marsh	56.58 ± 14.75	29.98 ± 26.63	39.95 ± 26.87	28.75 ± 29.37	19.07 ± 22.83
10. Lower salt marsh	71.80 ± 9.15	37.30 ± 26.71	30.54 ± 19.75	20.79 ± 22.51	21.76 ± 21.21
11. Sand	19.00 ± 24.97	15.74 ± 19.85	8.29 ± 11.30	13.49 ± 15.33	38.58 ± 21.37
12. Settlement	85.56 ± 5.25	46.83 ± 30.05	62.25 ± 37.99	34.02 ± 34.87	57.39 ± 24.04
Mean of means	36.93	21.71	22.22	15.71	23.70
Mean of std	10.43	17.24	17.05	17.40	14.80
Std of means	30.51	20.87	20.84	12.57	19.28
Mean CV	0.66	1.22	1.23	1.38	1.08

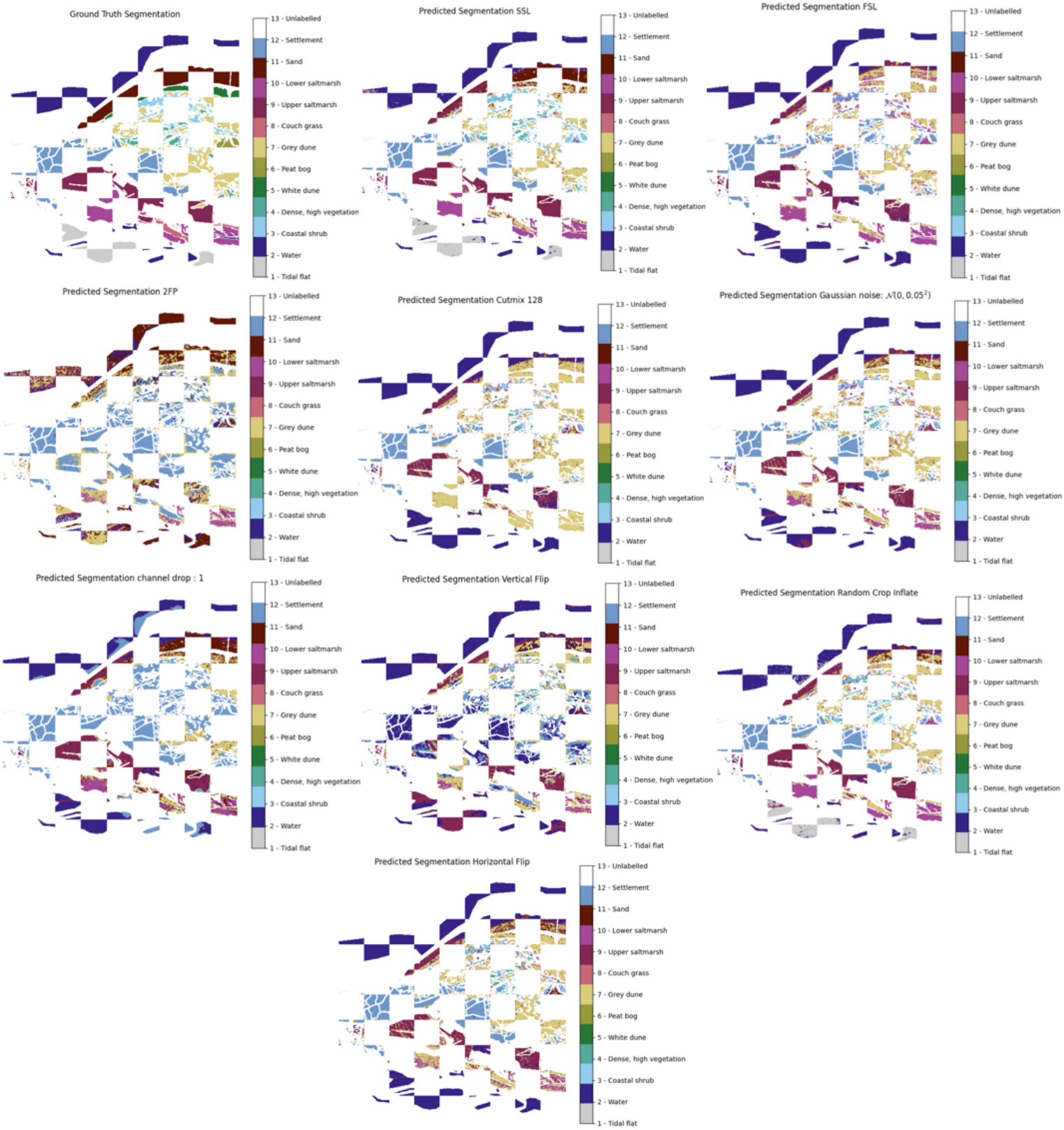


Figure 16 – Comparison of segmentation maps obtained with different training strategies. SSL provides the most accurate and stable results across all land cover classes. CutMix improves performance for homogeneous classes but produces over-smoothed boundaries in complex ecological transitions. Random crop demonstrates balanced segmentation performance with preserved spatial detail. Gaussian noise severely degrades performance by introducing artifacts and misclassifications, while horizontal and vertical flips, and channel dropout yield moderate improvements with variable class-dependent performance.

6 Conclusion

This study investigated the potential of SSL to improve the semantic segmentation of PolSAR imagery, particularly in scenarios with limited annotated data. The scarcity of labeled PolSAR datasets—resulting from the need for expert knowledge and the unique conditions of each acquisition campaign—poses a fundamental challenge to traditional supervised methods. To address this, we developed a MAE) framework that leverages unlabeled data from the Pol-InSAR-Island coastal monitoring dataset to learn meaningful feature representations. We demonstrated that transferring these learned features significantly improves the performance of downstream segmentation tasks.

Our quantitative evaluation demonstrated that SSL pretraining systematically outperformed the fully supervised baseline in all tested configurations. With an identical data processing pipeline (extended log-ratio representation and standardized normalization), the SSL framework achieved a mean Intersection over Union (IoU) of 36.93% compared to 21.71% for the baseline, a substantial improvement of +15.22%. Beyond accuracy, SSL also enhanced training stability, reducing the coefficient of variation across runs from 1.22 to 0.66. Our experiments confirmed that data representation and normalization strategies are critical, with the extended log-ratio representation and a 5%-trimmed standardization with clipping yielding optimal results. These findings show that SSL enables the extraction of robust, transferable features that are less sensitive to specific processing choices, thereby reducing the reliance on handcrafted pipelines.

Comparison with data augmentation techniques showed that CutMix was the most effective method (34.87% mean IoU), nearly matching the SSL results. However, SSL maintained superior performance for complex classes with heterogeneous boundaries, such as lower salt marsh (71.80% vs. 31.80% for CutMix). This demonstrates that pretraining provides more robust feature representations than data augmentation alone.

For future work, several limitations should be acknowledged. The quantitative evaluation of the masking rate was conducted after feature selection, which could introduce bias; a more systematic exploration of this hyperparameter is warranted. Furthermore, our best-performing model achieved a strong mean IoU of 50.2% using a single-frequency polarimetric approach. This result can be contextualized by the original benchmark study, which reached 67% mean IoU by leveraging a multi-modal dataset combining both S and L-band data with interferometric information [15]. This highlights a clear path forward: applying this SSL pretraining framework to the full multi-modal dataset may help bridge this performance gap.

Finally, future research could explore pretext tasks better suited to the unique challenges of PolSAR data. Given the presence of speckle, reconstructing raw pixel values may be suboptimal. A more powerful approach, inspired by recent work in SAR analysis that reconstructs abstract features instead of pixels [44], involves recovering physically meaningful representations. Therefore, designing a pretext task to recover polarimetric scattering mechanisms [9] is a particularly promising direction, as it would compel the model to learn features grounded in physical properties, making them inherently more robust to noise.

Bibliography

[1] Haixia Bi, Feng Xu, Zhiqiang Wei, Yibo Han, Yuanlong Cui, Yong Xue, and Zongben Xu. An active deep learning approach for minimally-supervised polsar image classification. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 3185–3188, 2019. 14

[2] Alexandre Becker Campos, Matthias H Braun, and Paola Rizzoli. A deep unsupervised learning approach for monitoring snow facies over ice sheets using tandem-x bistatic data. *IEEE Transactions on Geoscience and Remote Sensing*, 2025. 14

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 14

[4] S.R. Cloude and E. Pottier. An entropy based classification scheme for land applications of polarimetric sar. *IEEE Transactions on Geoscience and Remote Sensing*, 35(1):68–78, 1997. 15

[5] Jun-Wu Deng, Ming-Dian Li, and Si-Wei Chen. Urban damage-level estimation with reconstructed quad-pol sar data from dual-pol sar mode. *IEEE Geoscience and Remote Sensing Letters*, 21:1–5, 2024. 17

[6] Lei Ding, Kai Zheng, Dong Lin, Yuxing Chen, Bing Liu, Jiansheng Li, and Lorenzo Bruzzone. Mp-resnet: Multipath residual network for the semantic segmentation of high-resolution polsar images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. 14

[7] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners, 2022. 17

[8] Giulia Fracastoro, Enrico Magli, Giovanni Poggi, Giuseppe Scarpa, Diego Valsesia, and Luisa Verdoliva. Deep learning methods for synthetic aperture radar image despeckling: An overview of trends and perspectives. *IEEE Geoscience and Remote Sensing Magazine*, 9(2):29–51, 2021. 15

[9] A. Freeman and S. L. Durden. A Three-Component Scattering Model for Polarimetric SAR Data. *IEEE Transactions on Geoscience and Remote Sensing*, 36(3):963–973, 1998. 35

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. 18

[11] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018. 14

[12] Ping Han, Zetao Chen, Yishuang Wan, and Zheng Cheng. Polsar image classification based on optimal feature and convolution neural network. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 1735–1738, 2020. 15

[13] Trevor Hastie, Robert Tibshirani, Jerome Friedman, et al. The elements of statistical learning, 2009. 15

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 6, 17, 18

[15] Sylvia Hochstuhl, Niklas Pfeffer, Antje Thiele, Stefan Hinz, Joel Amao-Oliva, Rolf Scheiber, Andreas Reigber, and Holger Dirks. Pol-insar-island - a benchmark dataset for multi-frequency pol-insar data land cover classification. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, 10:100047, 2023. 2, 6, 19, 20, 28, 35

[16] Xuran Hu, Ziqiang Xu, Zhihan Chen, Zhengpeng Feng, Mingzhe Zhu, and LJubisa Stankovic. Sar despeckling via regional denoising diffusion probabilistic model, 2024. 17

[17] Zhongling Huang, Xidan Zhang, Zuqian Tang, Feng Xu, Mihai Datcu, and Junwei Han. Generative artificial intelligence meets synthetic aperture radar: A survey. *IEEE Geoscience and Remote Sensing Magazine*, 2024. 16

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 24

[19] Haris Iqbal. Plotneuralnet. https://github.com/HarisIqbal88/PlotNeuralNet, 2020. GitHub repository. 6, 25, 26

[20] Paul Jaccard. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bull Soc Vaudoise Sci Nat*, 37:241–272, 1901. 27

[21] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:4037–4058, 2019. 17

[22] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991. 17

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 14

[24] Yi Kuang, Fei Ma, Fangfang Li, Yingbing Liu, and Fan Zhang. Semantic-layout-guided image synthesis for high-quality synthetic-aperature radar detection sample generation. *Remote Sensing*, 15(24), 2023. 17

[25] Francesco Lattari, Vincenzo Santomarco, Riccardo Santambrogio, Alessio Rucci, and Matteo Matteucci. Cyclesar: Sar image despeckling as unpaired image-to-image translation. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2023. 17

[26] Jong-Sen Lee and Eric Pottier. *Polarimetric radar imaging: from basics to applications*. CRC press, 2017. 15

[27] David Long and Fawwaz Ulaby. *Microwave radar and radiometric remote sensing*. Artech, 2015. 15

[28] C. Lopez-Martinez and X. Fabregas. Polarimetric sar speckle noise model. *IEEE Transactions on Geoscience and Remote Sensing*, 41(10):2232–2242, 2003. 15

[29] Fariba Mohammadimanesh, Bahram Salehi, Masoud Mahdianpari, Eric Gill, and Matthieu Molinier. A new fully convolutional neural network for semantic segmentation of polarimetric sar imagery in complex land cover ecosystem. *ISPRS Journal of Photogrammetry and Remote Sensing*, 151:223–236, 2019. 14

[30] Alberto Moreira, Pau Prats-Iraola, Marwan Younis, Gerhard Krieger, Irena Hajnsek, and Konstantinos P. Papathanassiou. A tutorial on synthetic aperture radar. *IEEE Geoscience and Remote Sensing Magazine*, 1(1):6–43, 2013. 6, 10, 11, 12, 13

[31] Alhassan Mumuni and Fuseini Mumuni. Data augmentation: A comprehensive survey of modern approaches. *Array*, 16:100258, 2022. 16

[32] Manohara M.M. Pai, Vaibhav Mehrotra, Shreyas Aiyar, Ujjwal Verma, and Radhika M. Pai. Automatic segmentation of river and land in sar images: A deep learning approach. In *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 15–20, 2019. 14

[33] Malsha V. Perera, Nithin Gopalakrishnan Nair, Wele Gedara Chaminda Bandara, and Vishal M. Patel. Sar despeckling using a denoising diffusion probabilistic model. *IEEE Geoscience and Remote Sensing Letters*, 20:1–5, 2023. 17

[34] Moien Rangzan, Sara Attarchi, Richard Gloaguen, and Seyed Kazem Alavipanah. Tsgan: An optical-to-sar dual conditional gan for optical based sar temporal shifting, 2024. 16

[35] Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S. Zimmermann, and Wieland Brendel. Infonce: Identifying the gap between theory and practice, 2025. 17

[36] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 17

[37] Yu Shi, Lan Du, Yuchen Guo, and Yuang Du. Unsupervised domain adaptation based on progressive transfer for ship detection: From optical to sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–17, 2022. 16

[38] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019. 16

[39] Qian Song, Feng Xu, and Ya-Qiu Jin. Radar image colorization: Converting single-polarization to fully polarimetric using deep neural networks. *IEEE Access*, 6:1647–1661, 2018. 17

[40] Qian Song, Feng Xu, Xiao Xiang Zhu, and Ya-Qiu Jin. Learning to generate sar images with adversarial autoencoder. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2022. 16

[41] Farhana Sultana, Abu Sufian, and Paramartha Dutta. Evolution of image segmentation using deep convolutional neural network: A survey. *Knowledge-Based Systems*, 201-202:106062, 2020. 6, 13

[42] Kun Sun, Yi Liang, Xiaorui Ma, Yuanyuan Huai, and Mengdao Xing. Dsdet: A lightweight densely connected sparsely activated detector for ship target detection in high-resolution sar images. *Remote Sensing*, 13(14), 2021. 17

[43] Xiaying Wang, Lukas Cavigelli, Manuel Eggimann, Michele Magno, and Luca Benini. Hr-sar-net: A deep neural network for urban scene segmentation from high-resolution sar data. In *2020 IEEE Sensors Applications Symposium (SAS)*, pages 1–6, 2020. 14

[44] Yi Wang, Hugo Hernández Hernández, Conrad M Albrecht, and Xiao Xiang Zhu. Feature guided masked autoencoder for self-supervised learning in remote sensing, 2023. 18, 35

[45] Xinye Wanyan, Sachith Seneviratne, Shuchang Shen, and Michael Kirley. Extending global-local view alignment for self-supervised learning with remote sensing imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2443–2453, June 2024. 8

[46] Weihao Weng and Xin Zhu. Inet: Convolutional networks for biomedical image segmentation. *IEEE Access*, 9:16591–16603, 2021. 14

[47] Carl A. Wiley. Synthetic aperture radars. *IEEE Transactions on Aerospace and Electronic Systems*, AES-21(3):440–443, 1985. 10

[48] Suorong Yang, Weikang Xiao, Mengchen Zhang, Suhan Guo, Jian Zhao, and Furao Shen. Image data augmentation for deep learning: A survey, 2023. 16

[49] Wei Yao, Dimitrios Marmanis, and Mihai Datcu. Semantic segmentation using deep neural networks for sar and optical image pairs. 2017. 14

[50] Zhenyu Yue, Fei Gao, Qingxu Xiong, Jun Wang, Amir Hussain, and Huiyu Zhou. A novel attention fully convolutional network method for synthetic aperture radar image segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:4585–4598, 2020. 14

[51] Fan Zhang, Zhuoyue Cao, Deliang Xiang, Canbin Hu, Fei Ma, Qiang Yin, and Yongsheng Zhou. Pseudo quad-pol simulation from compact polarimetric sar data via a complex-valued dual-branch convolutional neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:901–918, 2022. 17

[52] Juanping Zhao, Weiwei Guo, Shiyong Cui, Zenghui Zhang, and Wenxian Yu. Convolutional neural network for sar image classification at patch level. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 945–948. IEEE, 2016. 15

[53] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, December 2017. 8