EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems
Research Article EAI.EU

Artificial Intelligence-Enabled Lightweight Flood Segmentation Model with Polarization Fusion and Attention

Premisha Premananthan[1], Brad D. E. McNiven[1], Muhammad Fahim[2], Quang Nhat Le[1,*], and Hang Le[3]

[1]Memorial University, St. John's, NL A1B3X5, Canada
[2]Queen's University Belfast, Belfast BT7 1NN, UK
[3]Duy Tan University, Da Nang, Vietnam

Abstract

Rapid and reliable flood maps are essential for emergency response and disaster management. However, many deep learning models for flood segmentation are computationally demanding, limiting real-time use and deployment on modest hardware. This paper presents *PFALNet*, a reduced-parameter segmentation network that integrates dual-polarization synthetic aperture radar image backscatter and a polarization-ratio channel. The model employs concurrent spatial and channel squeeze-and-excitation attention to enhance multi-scale feature fusion for flood delineation. Experiments on a public Sentinel-1 flood-mapping benchmark show that PFALNet achieves strong performance on the Florence test region (F1 = 90.53%, IoU = 82.70%, κ = 88.80%) while remaining lightweight (0.31M parameters, 5.88 GFLOPs) and enabling real-time inference (6.23 ms per 256 × 256 tile, ~ 160 tiles/s) on a single GPU. These results indicate that carefully designed lightweight synthetic aperture radar (SAR) segmentation models can deliver competitive performance with substantially reduced computational cost.

Received on 22 April 2026, accepted on 25 July 2026, published on 24 August 2026

Keywords: Attention mechanisms, flood segmentation, sentinel-1, synthetic aperture radar (SAR), UNet

doi: 10.4108/eetinis.133.12738

*Corresponding author. Email: qnle@mun.ca

1. Introduction

Floods are some of the most common and damaging natural disasters in the world [1–3]. They can cause loss of life, destroy roads, bridges, and buildings, disrupt daily activities, and create long-term economic problems. Recent major floods in South Asia, Europe, and North America have shown that even when accurate weather forecasts and early warning systems exist, their effectiveness can be limited. This highlights the need for fast and reliable flood maps. These maps help to plan evacuations, organize emergency services, and prioritize damage assessment. As a result, there is a strong need for automated flood-mapping systems that can work over large regions and provide near-real-time information for civil protection and humanitarian agencies.

Satellite remote sensing is a key part of these systems because it can regularly observe large areas of the Earth [1, 4, 5]. In particular, synthetic aperture radar (SAR) sensors such as Sentinel-1 are well-suited for flood monitoring because they operate independently of solar illumination and are largely insensitive to cloud cover and moderate precipitation. Sentinel-1 C-band SAR acquired in interferometric wide swath (IW) mode provides dual-polarization backscatter in vertical transmit vertical receive (VV) and vertical transmit horizontal receive (VH) with frequent revisit times, enabling monitoring of rapidly evolving flood events. At C-band, the backscatter response over water and land is governed by surface, volume, and double-bounce scattering mechanisms [2]. Permanent open water surfaces tend to appear dark due to specular reflection away from the sensor, while rough water

and flooded vegetation can produce enhanced volume or double-bounce returns, and urban areas exhibit complex scattering from buildings and infrastructure. Combining VV, VH, and a polarization-ratio feature computed directly from these two channels can therefore improve separability between permanent water, flooded vegetation, bare soil, and built-up areas.

Traditional SAR flood mapping often relied on thresholding and change detection, but performance can vary substantially across land-cover types and acquisition conditions [4, 5]. Recent work has therefore shifted toward data-driven segmentation using fully convolutional encoder-decoder architectures, where UNet-style designs are commonly adopted for dense prediction in remote sensing [6, 7]. Despite strong reported accuracy, many state-of-the-art flood segmentation models rely on heavy backbones with tens of millions of parameters and multi-giga floating-point operations (GFLOPs) inference cost, which can limit practical deployment in time-critical or resource-constrained settings.

A particularly influential benchmark for wide-area flood detection is the NASA Sentinel-1 flood detection dataset, originally released for the IEEE GRSS Emerging Techniques in Computational Intelligence (ETCI) 2021 Flood Detection Challenge. It comprises tiled Sentinel-1 IW scenes over four major flood events (Bangladesh, Nebraska, North Alabama, and Florence) with dual polarization and corresponding flood masks. Prior studies on this benchmark highlight that using all three inputs—VV, VH, and a polarization-ratio feature derived from VV and VH—can improve robustness across regions [2, 8].

Motivated by the need for efficient large-area mapping, this paper investigates whether a single lightweight UNet architecture, augmented with concurrent spatial and channel squeeze-excitation (scSE) attention, can achieve competitive performance on the NASA Sentinel-1 flood detection benchmark while drastically reducing parameters and computational cost compared to high-capacity UNet++ baselines [8, 9]. We focus on the standard three-channel NASA tiles (VV, VH, and a polarization ratio feature computed directly from these two channels) and adopt a leave-one-region-out (LORO) protocol to assess cross-region generalization.

This work contributes a lightweight SAR flood segmentation approach with the following elements. The proposed *PFALNet* is a lightweight segmentation network for Sentinel-1 SAR flood mapping on the NASA benchmark. The architecture uses a compact encoder-decoder backbone built from lightweight residual blocks and depthwise convolutions and integrates concurrent scSE modules at selected stages to provide efficient attention during multi-scale feature fusion. The model operates on three-channel Sentinel-1 tiles comprising VV, VH, and a polarization-ratio channel and is trained end-to-end using a composite loss that combines dice and focal terms to address the severe class imbalance between flooded and non-flooded pixels. Performance is evaluated on the NASA Sentinel-1 flood detection benchmark using a LORO protocol, reporting intersection over union (IoU), F1-score, and Cohen's κ together with model parameters, GFLOPs, and inference time. PFALNet is compared with published UNet and UNet++ results on the same dataset. This comparison is used to study the trade-off between performance and model complexity. The results show that the lightweight design remains competitive. At the same time, it uses far fewer parameters and much less computation than heavier UNet++ baselines.

The remainder of the paper is organized as follows. Section 2 reviews related work on SAR-based flood mapping, UNet-family architectures, lightweight segmentation networks, and attention mechanisms. Section 3 formulates the flood segmentation problem and summarizes the NASA benchmark dataset and evaluation metrics. Section 4 introduces the PFALNet architecture in detail. Section 5 describes the experimental setup. Section 6 presents quantitative and qualitative results. Section 7 discusses the findings and limitations. Section 8 concludes the paper and outlines directions for future research.

2. Related Work

2.1. SAR-Based Flood Mapping

Early SAR-based flood mapping pipelines were largely built around low-level backscatter analysis, thresholding, and change detection between pre- and post-event images [4, 5]. Typical approaches include global or adaptive thresholds on calibrated SAR intensity. Sometimes they also use digital elevation models and permanent-water masks. These extra inputs help reduce false alarms near riverbanks and in low-lying areas. These methods are attractive because they are conceptually simple and computationally efficient. Still, their performance can degrade under varying incidence angles, heterogeneous land cover, and complex scattering from vegetation or urban structures; consequently, parameters often need to be tuned for each new flood event and geographic region.

In recent years, researchers have increasingly used machine learning and deep learning for flood mapping. These methods learn patterns directly from labeled data instead of using hand-chosen thresholds. In near real time, deep models have also been used in practical flood-mapping workflows. For example, the authors in [3] trained UNet- and SegNet-based models and demonstrated quasi-operational flood mapping

with IoU up to about 88%. Beyond purely data-driven segmentation, several studies have coupled deep learning with hydrodynamic modeling or spatio-temporal filters to support near-real-time inundation prediction and mapping from Sentinel-1 imagery [10, 11]. Collectively, these works motivate the need for models that are not only accurate but also efficient and robust enough for frequent large-scale deployments.

2.2. UNet-Family Architectures for Flood Detection

UNet [6] introduced an encoder-decoder architecture with skip connections and has become a de facto standard for dense prediction in medical imaging and remote sensing. In the flood-mapping domain, the work in [12] trained a UNet on Sen1Floods11 and transferred it to a regional flood in Tabasco, Mexico, reporting an IoU of approximately 73%. Their case study shows that a relatively standard UNet, when trained on a global dataset, can produce operationally useful flood maps for specific events.

On the NASA Sentinel-1 benchmark, the authors in [2] trained UNet and feature pyramid networks (FPN) with EfficientNet-B7 encoders using three-channel inputs (VV, VH, and a polarization ratio feature computed directly from these two channels). Under a LORO protocol, they achieved a mean IoU (mIoU) of $\sim$ 75% on the held-out Florence region. In a follow-up study, the study in [8] explored nested UNet++ architectures with several backbones (ResNet34, InceptionV3, and EfficientNet-B7), systematically compared single-band and multi-band configurations, and performed a Shapley-value analysis of VV, VH, and ratio importance. They demonstrated that using VV, VH, and the polarization-ratio channel together consistently yields the best performance across NASA regions and generalizes reasonably well to additional flood events in Spain, India, and Vietnam.

Other recent work has further diversified UNet-style architectures for SAR floods. The authors in [13] proposed WVResU-Net, which integrates residual connections and vision multi-layer perceptrons (MLPs) into a UNet framework for dual-polarized Sentinel-1 imagery. On UrbanSARFloods and related datasets, WVResU-Net outperforms several standard convolutional neural network (CNN) and transformer baselines, suggesting that hybrid CNN-MLP designs can deliver strong performance in complex urban scenes. Beyond single-date inputs, the attentive dual stream siamese UNet (ADSS-UNet) for bi-temporal flood detection on Sen1Floods11 was introduced in [14]. ADSS-UNet processes pre- and post-flood VV/VH stacks through shared encoders and utilizes scSE modules in the decoder to fuse temporal information, resulting in clear gains in F1 and IoU compared to uni-temporal baselines.

Overall, these UNet-family models demonstrate that high-capacity encoder-decoder architectures can achieve excellent accuracy on global flood benchmarks. However, they typically employ heavy backbones (e.g., EfficientNet-B7, ResNet-50) with tens of millions of parameters and multi-GFLOP inference costs, and only a few works explicitly discuss runtime or hardware constraints.

Attention for boundary refinement: Several UNet variants enhance encoder-decoder feature fusion using attention or channel re-weighting, which can help suppress SAR clutter and sharpen flood boundaries. In this paper, a lightweight attention model is used to improve delineation while keeping the overall model efficient.

2.3. Flood Benchmark Datasets and Competitions

Several curated datasets and challenges have shaped recent progress in deep learning-based flood mapping. Sen1Floods11 [1] provides 512×512 Sentinel-1 tiles for 11 flood events with pixel-wise labels and has been widely used for training and evaluating UNet-type models [3, 14]. The NASA Sentinel-1 flood detection benchmark [2, 8], originally released for the IEEE GRSS ETCI 2021 Flood Detection Challenge, focuses on four large events (Bangladesh, Nebraska, North Alabama, and Florence) and offers thousands of 256×256 tiles with dual polarization and binary flood masks, supporting region-wise cross-validation protocols.

UrbanSARFloods [7] extends this landscape by providing 8,879 chips of size 512×512 over 18 flood events, together with intensity and coherence before and during inundation. The dataset covers both open floodplains and dense urban areas, and reveals that models trained primarily on open-water floods (e.g., Sen1Floods11) can struggle in urban settings [1]. These benchmarks support fair comparisons between different architectures and encourage systematic evaluations across regions and events. At the same time, they also expose the limitations of current models in terms of generalization, class imbalance, and sensitivity to land-cover heterogeneity.

3. Proposed Methodology

Let $x \in \mathbb{R}^{H\times W\times C}$ denote a multi-channel Sentinel-1 SAR tile, where H and W are the spatial dimensions and C denotes the number of input channels derived from the available polarizations. The goal is to predict a binary flood mask $y \in \{0,1\}^{H\times W}$, where $y_{ij} = 1$ indicates flooded pixels and $y_{ij} = 0$ indicates non-flooded pixels.

We learn a segmentation model $f_\theta(\cdot)$ parameterized by θ that maps an input tile to a per-pixel flood

probability map:

$$p = f_\theta(x), \qquad p \in [0,1]^{H\times W}. \tag{1}$$

A hard prediction $\hat{y}$ is obtained by thresholding the probabilities:

$$\hat{y}_{ij} = \begin{cases} 1, & p_{ij} \geq \tau, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where τ is a decision threshold used to convert predicted probabilities into binary flood masks. In all experiments, a fixed threshold of $\tau = 0.5$ is applied to the sigmoid outputs.

Given a training set $\mathcal{D} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$, model parameters are estimated by minimizing the empirical risk:

$$\theta^\star = \arg\min_\theta \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}\Big(f_\theta(x^{(n)}), y^{(n)}\Big), \tag{3}$$

where $\mathcal{L}(\cdot)$ is a segmentation loss function defined over pixel-wise predictions and labels. This formulation supports evaluation under cross-region splits, where the model is trained on a set of regions and tested on an unseen region to assess generalization.

Fig. 1 illustrates the overall architecture of *PFALNet* and all subsections in this section refer to the components highlighted in the figure.

3.1. Design Objectives

Our goal is to design a single UNet-style model that is small enough for potential real-time or embedded deployment, yet strong enough to compete with heavy UNet/UNet++ baselines on the NASA Sentinel-1 benchmark. Compared to encoder-decoder architectures with EfficientNet-B7 or similar high-capacity backbones [2, 8], the proposed network should reduce parameters and GFLOPs by one to two orders of magnitude while preserving most of the segmentation accuracy.

The design of PFALNet is guided by four principles. First, it preserves the classical encoder-decoder structure with skip connections, as in UNet [6], so that the network can combine fine local texture cues with broader spatial context. Second, it uses narrow feature channels and a moderate network depth, avoiding very wide encoders and deep cascades that typically dominate the computational cost in heavier models. Third, it improves efficiency by replacing part of the standard convolutions with depthwise or grouped operations, drawing on ideas from ENet [15], LinkNet [16], and MobileNetV3 [17]. Finally, it introduces attention through lightweight scSE blocks [9], applied only at strategically chosen stages where the improvement in feature refinement justifies the additional overhead.

Instead of designing multiple variants, we deliberately focus on a single compact architecture, PFALNet, and tune its hyperparameters (base channels, dropout, and weights) around the NASA benchmark. With a base width of 48 channels, two encoder downsampling stages, and deep supervision, the final configuration stays below 0.4 M parameters while still providing enough capacity for Sentinel-1 flood segmentation.

3.2. PFALNet Backbone and Attention

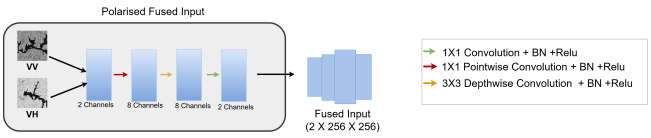

Figure 2. Polarized fused input sub-module.

Fig. 2 defines the polarized fused input submodule. The data pipeline first constructs a three-channel tensor:

$$\mathbf{x} = [\mathrm{VV}, \mathrm{VH}, R] \in \mathbb{R}^{3\times 256\times 256},$$

$$R = \frac{\mathrm{VV} + \mathrm{VH}}{|\mathrm{VV} - \mathrm{VH}| + \epsilon}, \tag{4}$$

where VV and VH are terrain-normalized Sentinel-1 backscatter channels. R is the polarization ratio, which is computed according to (4), clipped to a fixed range, and rescaled to $[0,1]$. ϵ is a small constant added for numerical stability. All three channels are normalized by a SAR-aware normalizer fitted on the training tiles. Inside the network, the tensor $\mathbf{x}$ is split as

$$\mathrm{VV} = \mathbf{x}_{0:1,:,:}, \quad \mathrm{VH} = \mathbf{x}_{1:2,:,:}, \quad R = \mathbf{x}_{2:3,:,:}.$$

A dedicated *polarization fusion* module then produces a single fused SAR channel from VV and VH. This module concatenates VV and VH into a two-channel map $[B, 2, 256, 256]$, projects it with a 1×1 pointwise convolution to 8 channels, applies a depthwise 3×3 convolution, and finally uses a 1×1 convolution with sigmoid activation to generate a spatial weight map $A \in [0,1]^{1\times 256\times 256}$. The fused polarization response is given by

$$F_{\mathrm{pol}} = A \odot \mathrm{VV} + (1 - A) \odot \mathrm{VH}. \tag{5}$$

Hence, the model can smoothly interpolate between VV- and VH-dominated behavior at each pixel. The ratio channel is kept separate, and the actual input to the UNet backbone becomes

$$\tilde{\mathbf{x}} = [F_{\mathrm{pol}}, R] \in \mathbb{R}^{2\times 256\times 256}.$$

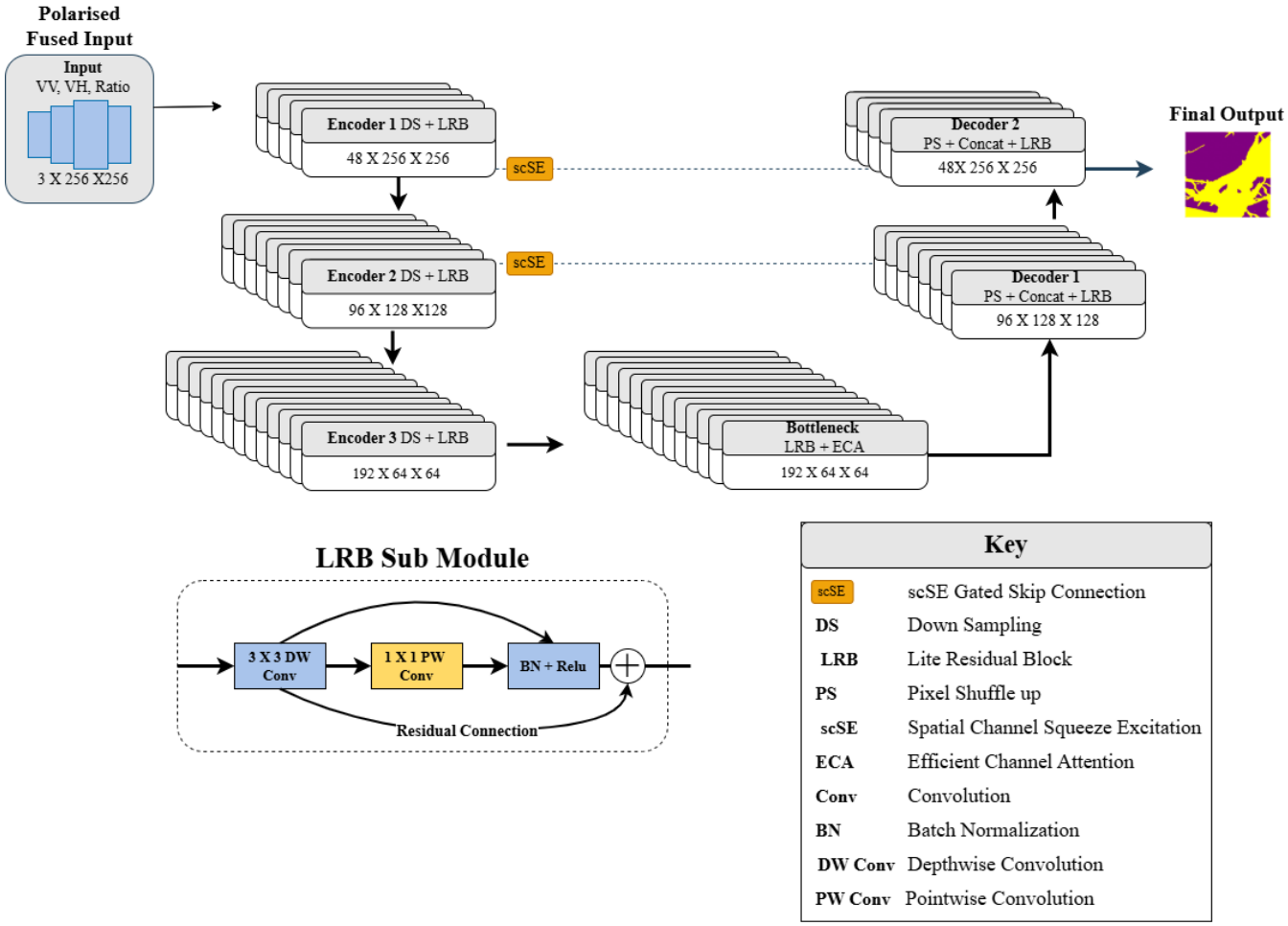

Figure 1. Overview of the proposed PFALNet architecture.

3.3. Encoder

Encoder 1 (e_0): The e_0 receives $\tilde{\mathbf{x}}$ and consists of a 3×3 convolution followed by batch normalization, rectified linear unit (ReLU), and a lite residual block (LRB). The e_0 maps the input tensor to a feature representation with 48 channels at a spatial resolution of 256×256, which is then passed to the encoder stages. The e_0 maps

$$[B, 2, 256, 256] \rightarrow [B, 48, 256, 256],$$

where the LRB internally expands the feature width from 48 to 96 channels and projects it back to 48 while preserving a residual connection and applying dropout. The e_0 output with spatial resolution 256×256 and 48 channels is stored as the first skip feature.

The second encoder stage (e_1) performs spatial downsampling and channel expansion. A stride-2 convolution reduces the resolution from 256×256 to 128×128 and expands the channel width from 48 to 96. For efficiency, the downsampling operator is implemented using depthwise separable convolution, followed by batch normalization and ReLU. A LRB then refines the features at 96 channels:

$$e_0 : [B, 48, 256, 256] \rightarrow e_1 : [B, 96, 128, 128].$$

An scSE block with 96 channels is applied on e_1 to produce a gated skip feature used later in the decoder.

The third encoder stage (e_2) repeats the pattern at a deeper level. The feature map is downsampled from 128×128 to 64×64 using a stride-2 convolution, and the channels are expanded from 96 to 192. For efficiency, the downsampling is implemented using depthwise-separable convolution, followed by batch normalization, ReLU, and a LRB at 192 channels:

$$e_1 : [B, 96, 128, 128] \rightarrow e_2 : [B, 192, 64, 64].$$

3.4. Bottleneck

Bottleneck with Channel Attention (b): The bottleneck consists of a LRB at 192 channels followed by an efficient channel attention (ECA) block. The LRB refines e_2 without changing its shape, and the ECA module applies channel-wise reweighting based on global

average-pooled statistics. Hence, the bottleneck output $b \in \mathbb{R}^{B\times192\times64\times64}$ is shown as

$$e_2 : [B, 192, 64, 64] \;\rightarrow\; b : [B, 192, 64, 64].$$

3.5. Decoder

Decoder and scSE-Gated Skips: The decoder mirrors the encoder but uses efficient PixelShuffle-based upsampling and scSE-gated skip connections.

Decoder 1 (d_1) receives as input the bottleneck feature map produced at the deepest encoder level. This bottleneck output is first upsampled by a factor of two using pixel shuffle, restoring the spatial resolution from 64×64 to 128×128. In parallel, the corresponding encoder feature map at the same spatial scale is passed through a 1×1 convolution and scSE attention to form the skip connection. The upsampled bottleneck features and the processed skip features are then concatenated along the channel dimension and refined using LRB. A 1×1 convolution first maps the 192 channels to $4 \times 96 = 384$, and a PixelShuffle layer rearranges these into 96 channels at twice the resolution, producing

$$[B, 192, 64, 64] \;\rightarrow\; [B, 96, 128, 128].$$

This upsampled tensor is added to the scSE-gated skip from e_1 (also $[B, 96, 128, 128]$), and the sum passes through a LRB at 96 channels:

$$b : [B, 192, 64, 64] \;\rightarrow\; d_1 : [B, 96, 128, 128].$$

This stage corresponds to "d_1: Pixel Shuffle up (192→96) + LRB (96×128×128)" in Fig. 1.

Decoder 2 (d_2) upsamples d_1 back to the original resolution. A 1×1 convolution maps 96 channels to $4 \times 48 = 192$, PixelShuffle upsamples from 128×128 to 256×256 and reduces channels to 48, and the result is added to the scSE-gated skip from e_0:

$$d_1 : [B, 96, 128, 128] \;\rightarrow\; d_2 : [B, 48, 256, 256].$$

An LRB at 48 channels then refines the fused features. In d_2, it first increases the spatial resolution back to 256×256 using pixel shuffle while reducing the channel width from 96 to 48. It then merges these upsampled decoder features with the second encoder skip feature, which is passed through scSE gating to suppress irrelevant responses and emphasize flood-relevant information. After concatenation, the LRB operates at 48 channels to smooth and refine the final full-resolution representation.

Deep-Supervision Heads: PFALNet employs three prediction heads for deep supervision. A 1×1 convolution applied to the final decoder feature d_2 produces full-resolution logits

$$\ell_{\text{final}} \in \mathbb{R}^{B\times1\times256\times256}.$$

Two auxiliary heads operate on intermediate features: one on d_1 at $[B, 96, 128, 128]$ and one on the bottleneck b at $[B, 192, 64, 64]$. Each head is a 1×1 convolution to a single channel; the corresponding logits are then upsampled with bilinear interpolation to 256×256 so that all three outputs share the same spatial size. During training, the composite focal and dice loss is averaged over the three heads, which stabilizes optimization and encourages correct predictions at multiple scales. During evaluation and for qualitative visualization, only the final full-resolution head is used to compute flood probabilities and binary masks.

scSE Attention on Skip Features: To enhance the skip features before they are merged into the decoder, we adopt scSE blocks [9] on the encoder features used in the skip connections. Given an input feature map $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$, the channel squeeze-excitation branch computes a global descriptor via average pooling and a small bottleneck MLP:

$$\mathbf{z}_c = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbf{F}_{:,i,j} \in \mathbb{R}^C, \tag{6}$$

$$\mathbf{s}_c = \sigma\Big(\mathbf{W}_2\,\delta(\mathbf{W}_1\mathbf{z}_c)\Big), \tag{7}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable matrices, $\delta(\cdot)$ is a ReLU, and $\sigma(\cdot)$ is a sigmoid. The channel-reweighted feature map is given by

$$\mathbf{F}_{\text{cSE}} = \mathbf{F} \odot \mathbf{s}_c, \tag{8}$$

with $\odot$ denoting channel-wise multiplication broadcast over H and W.

The spatial squeeze-excitation branch instead predicts a spatial mask via a 1×1 convolution:

$$\mathbf{s}_s = \sigma(\mathbf{W}_s * \mathbf{F}) \in (0, 1)^{1\times H\times W}, \tag{9}$$

$$\mathbf{F}_{\text{sSE}} = \mathbf{F} \odot \mathbf{s}_s, \tag{10}$$

where $\mathbf{W}_s$ is a 1×1 kernel. The final scSE output is obtained by an element-wise maximum:

$$\mathbf{F}_{\text{scSE}} = \max\left(\mathbf{F}_{\text{cSE}}, \mathbf{F}_{\text{sSE}}\right). \tag{11}$$

scSE is applied only to the deeper encoder layers and the bottleneck. This choice is based on how features behave in the network. Deep layers contain high-level information that better separates flooded and non-flooded areas. These deep features also have lower spatial resolution, so attention is cheaper to compute. Adding attention after every convolution block (as done in some attention UNet variants [18]) would not be efficient. For a lightweight model, it would increase the parameter count and inference time too much. Compared with other compact UNet-style models such as ENet, LinkNet, LSFUnet, DeepGDLeNet, and LMSC-UNet [15, 16, 19], PFALNet follows a different strategy.

It keeps the encoder depth shallow, introduces explicit polarization fusion tailored to Sentinel-1, concentrates its attention budget on a small number of strategically chosen layers, and adds deep supervision to regularize training of this very compact backbone.

3.6. Loss Function and Optimization

Flood pixels typically occupy only a small fraction of each NASA tile, especially in large rural or urban scenes where non-flooded land dominates. As noted in prior work on Sen1Floods11 and bi-temporal flood detection [1, 14], this severe class imbalance can lead to models that optimize overall accuracy while under-detecting floods. To counter this, we adopt a composite loss that combines soft dice loss and focal loss [20], closely aligned with the class-imbalance mitigation strategies commonly used in flood segmentation.

Let $\hat{p}_{ij} \in [0,1]$ and $y_{ij} \in \{0,1\}$ denote the predicted probability and ground-truth label at pixel (i,j). The soft dice coefficient is defined as

$$d = \frac{2\sum_{i,j} \hat{p}_{ij} y_{ij}}{\sum_{i,j} \hat{p}_{ij} + \sum_{i,j} y_{ij} + \epsilon}, \tag{12}$$

where ϵ is a small constant for numerical stability. The corresponding dice loss is

$$\mathcal{L}_d = 1 - d. \tag{13}$$

This term directly optimizes the overlap between predicted and reference flood masks, which correlates well with the IoU and F1 metrics used for evaluation.

The focal loss emphasizes hard-to-classify pixels by down-weighting easy examples [20]. The symmetric binary focal loss used during training is given by

$$\mathcal{L}_f = -\frac{1}{N} \sum_{i,j} \Big[y_{ij}(1-\hat{p}_{ij})^{\gamma} \log(\hat{p}_{ij}) + (1-y_{ij})\hat{p}_{ij}^{\gamma} \log(1-\hat{p}_{ij}) \Big], \tag{14}$$

where N is the number of pixels and $\gamma > 0$ denotes the focusing parameter. (14) represents the symmetric binary focal loss used in the implementation. Both flooded (positive) and non-flooded (negative) classes contribute to the loss through the first and second terms, respectively. The modulation factors reduce the contribution of easy pixels and assign greater weight to hard or misclassified samples, making the loss particularly effective under severe class imbalance.

The total loss is a convex combination of the dice and focal losses:

$$\mathcal{L} = \alpha \mathcal{L}_d + (1-\alpha)\mathcal{L}_f, \tag{15}$$

where $\alpha \in [0,1]$ controls the trade-off between region-level overlap and pixel-level reweighting. In all experiments, $\alpha = 0.4$ and $\gamma = 2.2$ are used.

Compared with using IoU-based or binary cross-entropy losses alone, the proposed composite loss encourages accurate alignment between predicted and reference flood extents through the dice term while simultaneously emphasizing under-represented flood pixels and difficult boundary regions through the focal term. This design is consistent with ADSS-UNet [14] and other flood-segmentation frameworks, where composite losses have been shown to improve optimization stability and foreground recall.

PFALNet is optimized using the Adam optimizer with an initial learning rate η_0 and weight decay λ. A ReduceLROnPlateau scheduler is applied with the validation selection metric $m\text{F1} + m\text{IoU}$ as the monitoring signal, so the learning rate is reduced only when improvements in joint mean F1 and mIoU saturate. This reflects the evaluation practice in our experiments and encourages models that balance foreground and background performance.

3.7. Model Complexity

Table 1 summarizes the parameter counts of PFALNet and representative high-capacity baselines. For NASA UNet++ with EfficientNet-B7 encoders, the parameter count is dominated by the backbone (around 66 M parameters), while ResNet-34-based configurations are still in the tens of millions. Attention U-Net variants with standard convolutions typically remain close to one million parameters. By contrast, PFALNet uses approximately 0.308 M parameters in the configuration evaluated in this paper, with an associated complexity of about 5.88 GFLOPs per 256×256 tile.

Table 1. Reported parameter counts for representative UNet-based baselines and the proposed model

Model	Parameters (M)
UNet++ (EfficientNet-B7 encoder) [8]	66.0
EfficientNet-B7 (backbone) [21]	66.0
ResNet-34 backbone [22]	21.8
Attention U-Net [18]	0.982
PFALNet (Our model)	**0.308**

This places PFALNet in a similar parameter regime to ENet-like designs [15], while being tailored to Sentinel-1 flood mapping and equipped with polarization fusion, scSE attention, and deep supervision. This compact architecture achieves competitive IoU and F1 on the NASA benchmark, demonstrating that careful architectural choices and loss design can substantially reduce complexity without sacrificing segmentation quality.

Table 2. Flood events in the NASA Sentinel-1 benchmark dataset and their spatial/temporal coverage

Area	Area (sq. km)	Start Date	End Date
Bangladesh	7,150	14.03.2017	12.07.2017
Nebraska	1,741	08.01.2017	22.12.2017
North Alabama	13,789	02.03.2019	27.12.2019
Florence	7,197	01.10.2018	05.10.2018

4. Experimental Setup

4.1. Data Splits and Preprocessing

The NASA Sentinel-1 benchmark described in Section III-B is used in this study. Table 2 summarizes the four flood events, including their spatial coverage and time windows.

A LORO protocol is adopted. All tiles from Florence are held out as an unseen test set. Tiles from Bangladesh, Nebraska, and North Alabama are used for training and validation. In total, the training set contains 32,421 tiles and the test set contains 10,400 tiles.

To avoid spatial and temporal leakage, the split inside the training regions is performed at the `region_date` (scene) level rather than at the tile level. All tiles that share the same `region_date` key are kept in the same subset. The unique `region_date` groups are shuffled using a fixed random seed. Approximately 80% of the groups are assigned to the training set and 20% to the validation set. The 80%–20% training-validation split was chosen to balance two goals: maximizing the amount of training data available for model learning while reserving a sufficiently large validation subset for reliable model selection. Since the split is performed at the region-date (scene) level, this ratio also helps maintain independence between training and validation samples, and reduces the risk of scene-level leakage.

A SAR-aware normalizer is fitted using only the VV and VH tiles from the training split. The raw PNG inputs are converted to floating-point arrays in the range $[0, 1]$ and mapped to an assumed dB interval (e.g., $[-25, 0]$ dB). The normalizer estimates global or robust statistics from the training pixels to define a clipping window in dB, after which values are linearly rescaled back to $[0, 1]$. At run time, the same train-fitted normalizer is applied to every training, validation, and test tile, ensuring consistent intensity scaling while preventing any information from the validation or test sets from influencing normalization parameters.

For each tile, the polarization ratio channel R is computed according to (4), clipped to a fixed range, and rescaled to $[0, 1]$ to match the intensity channels. The final network input thus consists of three channels: normalized VV, normalized VH, and the normalized polarization ratio R. The corresponding flood mask is resized using nearest-neighbor interpolation (if required) and thresholded at 0.5 to obtain a single-channel binary target tensor.

During training, we apply random horizontal and vertical flips and rotations by multiples of 90° to augment the data and improve generalization. No augmentation is applied during validation or testing. To address class imbalance at the tile level, tiles whose flooded area exceeds a minimal fraction are flagged as positive and used to construct a *WeightedRandomSampler* for the training loader. The sampler assigns larger sampling weights to flood-containing tiles so that each training epoch observes a more balanced mix of flooded and non-flooded examples. The validation and test sets are left untouched by re-weighting and are used solely for model selection and final performance reporting.

4.2. Training Configuration

All models are implemented in PyTorch and trained on a single NVIDIA RTX A6000 GPU using standard single-precision (FP32) arithmetic. The batch size is fixed at $B = 64$, and each model is optimized for up to 100 epochs, with the best checkpoint selected according to the validation selection metric $m\mathrm{F1} + m\mathrm{IoU}$. The optimizer is Adam with an initial learning rate of $\eta_0 \approx 3.85 \times 10^{-4}$ and weight decay $\lambda \approx 5.84 \times 10^{-5}$. For supervision, we employ a deep-supervision variant of a combined focal-dice loss, where the dice and focal components are mixed with a coefficient $\alpha_{\mathrm{mix}} = 0.4$, the focal-loss focusing parameter is set to $\gamma = 2.2$, and the focal-loss class weight is $\alpha_{\mathrm{focal}} = 1.0$. A fixed decision threshold of 0.5 on the sigmoid outputs is used to obtain binary flood masks, from which all reported pixel-wise metrics are computed.

4.3. Evaluation Metrics

We evaluate segmentation quality using standard pixel-wise metrics derived from the confusion matrix, including precision, recall, F1-score, and IoU for the flood class. We also report Cohen's κ and mean metrics across classes. To characterize efficiency, we report the number of parameters, model size in megabytes, multiply-accumulate operations (MACs), and the computational cost in GFLOPs per 256×256 tile.

Given the predicted mask $\hat{\mathbf{y}}$ and reference mask $\mathbf{y}$, each pixel is classified as true positive (TP), true negative (TN), false positive (FP), or false negative (FN). From these counts, we derive the standard evaluation metrics used in flood segmentation studies

on Sen1Floods11 and the NASA benchmark [1, 2, 8]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{16}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{17}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{18}$$

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}. \tag{19}$$

The IoU in (19) is defined for a single class (here, the flooded class) and is the primary metric used to compare different models on the NASA benchmark [2, 8]. In our binary flood vs. non-flood setting, we explicitly compute IoU and F1 for each class and then take their average. For the flooded (positive) class, the confusion counts (TP, FP, FN) in (16)-(19) yield

$$\text{IoU}_\text{f} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \qquad \text{F1}_\text{f} = \frac{2\,\text{TP}}{2\,\text{TP} + \text{FP} + \text{FN}}. \tag{20}$$

For the non-flood (negative) class, the roles of positives and negatives are reversed. Using $\text{TP}_\text{nf} = \text{TN}$, $\text{FP}_\text{nf} = \text{FN}$, and $\text{FN}_\text{nf} = \text{FP}$, where f = flood and nf = non-flood, we obtain

$$\begin{aligned} \text{IoU}_\text{nf} &= \frac{\text{TN}}{\text{TN} + \text{FN} + \text{FP}}, \\ \text{F1}_\text{nf} &= \frac{2\,\text{TN}}{2\,\text{TN} + \text{FN} + \text{FP}}. \end{aligned} \tag{21}$$

The mIoU and mean F1 (mF1) reported in this work are then defined as the macro average over the two classes:

$$\text{mIoU} = \frac{1}{2}\left(\text{IoU}_\text{f} + \text{IoU}_\text{nf}\right), \tag{22}$$

$$\text{mF1} = \frac{1}{2}\left(\text{F1}_\text{f} + \text{F1}_\text{nf}\right). \tag{23}$$

In other words, IoU_f and F1_f measure performance on the flooded class specifically, while mIoU and mF1 summarize performance across both flood and non-flood classes with equal weight. This macro-averaging is particularly useful under strong class imbalance, where overall accuracy can be dominated by the majority non-flood class, and a model that predicts "no flood" almost everywhere may still achieve a seemingly high score. Using only flood-class IoU/F1, on the other hand, can hide poor behaviour on the background class (e.g., large false-alarm regions), whereas mIoU and mF1 penalize errors on both classes symmetrically. For these reasons, recent NASA-based flood studies, as well as work on Sen1Floods11 and UrbanSARFloods, routinely report IoU/F1 for the flood class together with mIoU or related mean scores for model comparison [1, 2, 8, 13, 14].

To account for agreement beyond chance, we also report Cohen's κ coefficient [**?**]. Let p_o denote the observed agreement (i.e., overall accuracy) and p_e represent the expected agreement under random labeling, computed from the row and column marginals of the confusion matrix. Cohen's κ is given by

$$\kappa = \frac{p_o - p_e}{1 - p_e}. \tag{24}$$

High values of κ indicate that predictions agree with the reference labels substantially more often than would be expected by chance, which is particularly informative when the non-flood class dominates the pixel distribution.

Although the focus here is on binary segmentation, the same formalism naturally extends to multi-class scenarios in which separate labels are provided for permanent water, temporary floodwater, and land. In that case, one replaces the scalar masks with one-hot encodings and computes class-wise IoU, mIoU, and macro-averaged F1-scores. Several datasets, including Sen1Floods11, are distributed with multi-class water labels that are often collapsed to a binary decision for training and evaluation [1].

Beyond segmentation accuracy, a key objective of this work is to quantify the computational efficiency of the proposed lightweight architecture. To this end, we complement IoU, F1-score, and κ with the following complexity indicators for each model:

- Number of parameters: total count of learnable weights (reported in millions).
- Computational cost: number of GFLOPs required for a single forward pass on a 256×256 tile.
- Runtime: average inference time per tile and the corresponding throughput in frames per second (FPS), measured on a modern GPU under identical conditions.

These metrics allow us to directly compare PFALNet with heavy UNet/UNet++ baselines on the NASA benchmark dataset and to analyze the trade-off between segmentation quality and resource usage, which is critical for potential deployment on embedded or edge-computing platforms.

4.4. Baselines and Evaluation Protocol

Two configurations are considered in the evaluation. The first is *PFALNet* (present work), a lightweight UNet equipped with scSE attention, trained end-to-end on three NASA regions and evaluated on the held-out region under a LORO protocol. The second consists of published UNet/UNet++ baselines reported in [8] for the Florence region, including UNet variants with ResNet34, InceptionV3, and EfficientNet-B7 encoders, as well as UNet++ with an EfficientNet-B7 encoder.

Table 3. Flood-class performance comparison with previous UNet-based models on the test data from Florence

Performance metric	UNet ResNet34	UNet InceptionV3	UNet EfficientNet-B7	UNet++ EfficientNet-B7	PFALNet
Precision (%)	85.10	80.05	87.02	89.50	**91.05**
Recall (%)	88.25	87.80	88.90	89.10	**90.02**
F1 (%)	86.60	83.74	87.94	89.30	**90.53**
IoU (%)	73.10	71.70	75.06	75.76	**82.70**
Cohen's κ (%)	79.00	74.50	80.70	81.60	**88.80**

For PFALNet, we report metrics on the validation set (aggregated over Bangladesh, Nebraska, and North Alabama) and on the Florence test set. For the UNet/UNet++ baselines, we reproduce the metrics from [8], noting that their training configuration differs and that these serve as published reference values rather than direct reimplementations.

5. Results and Discussion

5.1. Quantitative Results on the Florence Region

Table 3 summarizes the performance of PFALNet on the Florence test region and compares it with reported UNet/UNet++ baselines from [8]. Despite using substantially fewer parameters than EfficientNet-B7-based configurations, PFALNet attains higher Florence test precision, recall, F1-score, IoU, and Cohen's κ, indicating that an appropriately designed lightweight architecture can achieve strong flood segmentation performance on this benchmark.

5.2. Complexity vs Performance

Table 4 summarizes the main implementation and training settings used for PFALNet. It includes the software and hardware environment, input size, optimization settings, loss-function parameters, and decision threshold used throughout the experiments. These settings were kept fixed during training to ensure a consistent and reproducible evaluation protocol. The table also provides a compact reference for the key hyperparameters used in the reported results.

Table 5 summarizes the computational complexity of PFALNet. Although UNet++ with EfficientNet-B7 delivers strong accuracy, it typically incurs much higher computational cost due to the heavy backbone [8]. In contrast, PFALNet achieves competitive performance for the same Florence test data with a small parameter count and low GFLOPs and provides real-time inference on a single GPU (e.g., about 6.23 ms per tile, corresponding to roughly 160 tiles/s).

Across the four regions, Table 5 confirms that the model capacity (Total_Params, MACs, and GFLOPs) is fixed, so any variation in performance reflects differences in data difficulty and domain shift among regions. Table 6 shows that Bangladesh and Nebraska yield lower precision and recall, with Best_Test mF1 values of about 72% and Best_Test mIoU ranging from about 56% to 63%, indicating these regions are more challenging for the model. In contrast, North Alabama and Florence achieve substantially higher scores, with Best_Test mF1 of about 92% and 95%, respectively, while Best_Test mIoU is about 87% in both cases.

When designing a generalized flood segmentation model, it is therefore important to report results on both easier and more challenging regions. A practical protocol is a LORO evaluation, where Florence is treated as an unseen test set, and the remaining regions are used for training and validation, directly measuring generalization to new geographic and environmental conditions. Taken together, the fixed compute footprint in Table 5 and the region-wise results in Table 6 indicate that the proposed lightweight architecture is computationally efficient and achieves strong accuracy in multiple regions, while highlighting challenging cases where robustness can be further improved.

5.3. Ablation: Effect of scSE Attention

To isolate the contribution of scSE attention, we perform an ablation study comparing two configurations of the same lightweight backbone: a version without scSE (*PFALNet, no scSE*) and the full PFALNet model. Both networks share the same encoder-decoder topology and training setup; the only difference is the presence or absence of scSE blocks.

Table 7 reports validation metrics for these two configurations. The backbone-only network attains moderate precision and recall, resulting in an F1-score of 62.7% and an IoU of 45.64%, with a κ value of 52.3%. This indicates that, while the lightweight backbone can detect some flooded pixels, it struggles to delineate inundation boundaries accurately and suffers from both missed detections and false alarms.

By contrast, adding scSE attention leads to a substantial improvement across all metrics. PFALNet achieves precision and recall of 91.05% and 90.02%, respectively, yielding an F1-score of 90.53% and an IoU

Table 4. Main implementation and training parameters used for PFALNet

Parameter	Value
Implementation	PyTorch
Hardware	Single NVIDIA RTX A6000 GPU
Numeric precision	FP32
Tile size	256×256
Batch size	64
Maximum epochs	100
Optimizer	Adam
Initial learning rate	3.85×10^{-4}
Weight decay	5.84×10^{-5}
Learning-rate scheduler	ReduceLROnPlateau
Validation criterion	$mF1 + mIoU$
Loss function	Composite focal–dice loss with deep supervision
Decision Threshold	$\tau = 0.5$
Dice–focal mixing coefficient	$\alpha = 0.4$
Focal-loss focusing parameter	$\gamma = 2.2$

Table 5. Model complexity (same architecture used across all regions)

Total parameters	Size (MB)	MACs (M)	GFLOPs
309,089	1.18	2939.69	5.88

of 82.7%. The κ value increases to 88.8%, reflecting very strong agreement with the reference masks despite class imbalance. In other words, scSE attention helps the network focus on flood-relevant channels and locations, recovering most flooded pixels while keeping spurious detections low, with only a modest increase in parameters and GFLOPs. Based on this ablation, PFALNet is presented as the final architecture in this study.

5.4. Qualitative Analysis

Fig. 3 illustrates qualitative examples of the proposed flood segmentation by PFALNet on Florence tiles. In urban areas with complex backscatter, PFALNet better delineates flooded streets and small water bodies, while reducing false positives on rooftops and bright scatterers. In rural floodplains, scSE helps preserve thin flooded channels and suppress isolated noise patches, leading to more spatially coherent flood masks. Each row corresponds to one Sentinel-1 SAR tile, and the three columns show, from left to right, the SAR input image, the corresponding ground-truth water mask, and the prediction produced by the lightweight UNet model. The SAR image is a three-channel composite derived from VV, VH, and a polarization-based ratio, where dark and homogeneous regions typically indicate open water with low backscatter, while brighter and more textured regions correspond to land surfaces such as vegetation, urban areas, and rough terrain. This raw SAR information is the only input provided to the network.

The middle column shows the ground-truth masks used for supervision. For the NASA/ETCI dataset, these masks are constructed by combining a permanent water product (large rivers, lakes, reservoirs, and coastal waters) with an event-specific flood water product, so that any pixel identified as either permanent or flooded water is labeled as water, and all remaining valid pixels are labeled as land. In the visualization, land is rendered in purple and water in yellow. The right column presents the model's predictions. Internally, the network outputs a probability map for each pixel representing the likelihood of belonging to the water class; a fixed threshold (e.g., $\tau = 0.5$) is then applied to obtain a binary mask, which is visualized using the same purple/yellow color scheme. Comparing the predicted masks with the ground truth reveals that the model successfully captures the extent of major water bodies and floodplains, with discrepancies along complex shorelines and narrow channels corresponding to false positives and false negatives. These pixel-wise agreements and disagreements form the basis for computing precision, recall, F1-score, and IoU reported in the results section in the paper, while the qualitative examples in Fig. 3 provide an intuitive visual confirmation of the model's ability to delineate both permanent and flood-induced water regions.

Table 6. Region-wise model performance evaluation for the PFALNet model

Region	Precision (%)	Recall (%)	Best Test mF1 (%)	Best Test mIoU (%)
Bangladesh	47.03	30.75	72.44	63.26
Nebraska	45.61	34.69	72.45	56.63
North Alabama	84.91	73.77	92.30	86.59
Florence	**91.05**	**90.02**	**94.82**	**86.93**

Table 7. Ablation study: backbone-only PFALNet vs. PFALNet with scSE attention on the Florence test dataset

Metric	PFALNet (no scSE)	PFALNet
Precision (%)	68.30	91.05
Recall (%)	57.90	90.02
F1 (%)	62.70	90.53
IoU (%)	45.64	82.70
Kappa (%)	52.30	88.80

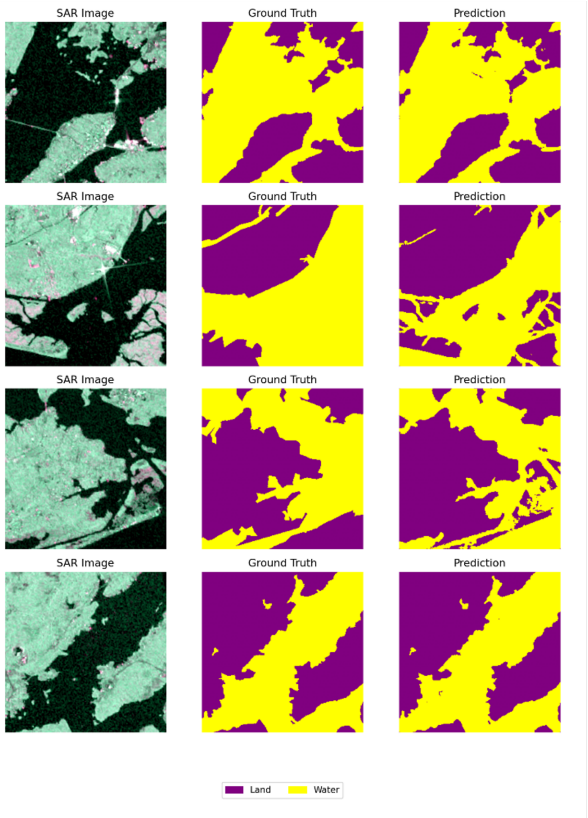

Figure 3. Qualitative examples on the Florence test region. From left to right: Sentinel-1 VV backscatter, ground-truth flood mask, and prediction from PFALNet

The experiments indicate that PFALNet achieves a favorable balance between segmentation performance and computational efficiency on the NASA Sentinel-1 benchmark. By combining a compact encoder-decoder backbone with scSE attention, the model achieves comparable IoU and F1 performance with substantially fewer parameters and GFLOPs than high-capacity UNet++ configurations, while maintaining real-time inference on a single GPU.

6. Conclusion

This paper presented *PFALNet*, an efficient model for flood segmentation from Sentinel-1 SAR imagery. Experiments on a publicly available NASA Sentinel-1 benchmark dataset demonstrate competitive flood delineation compared with heavier UNet-family baselines, while reducing computational cost and supporting fast inference. These findings suggest the proposed model is suitable for operational and large-area flood mapping where rapid updates are required. These results suggest that carefully designed lightweight architectures can overcome much of the gap to large backbones while being far more suitable for real-time and embedded deployments, such as rapid-response mapping chains or edge devices in emergency operation centers. Future work will extend the PFALNet model to multi-temporal inputs, incorporate ancillary data such as elevation and land-cover maps, and explore further compression techniques and cross-dataset evaluations to improve generalization to the most difficult flood scenarios.

References

[1] D. Bonafilia, B. Tellman, T. Anderson, and E. Issenberg, "Sen1Floods11: A georeferenced dataset to train and test deep learning flood algorithms for Sentinel-1," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Seattle, WA, Jun. 2020, pp. 210–211.

[2] B. Ghosh, S. Garg, M. Motagh, and S. Martinis, "Automatic flood detection from Sentinel-1 data using deep learning architectures," in *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. V-3-2022, Nice, France, Jun. 2022, pp. 201–208.

[3] V. Katiyar, N. Tamkuan, and M. Nagai, "Near-real-time flood mapping using off-the-shelf models with SAR imagery and deep learning," in *Proc. IEEE Int. Geosci.*

Remote Sens. Symp. (IGARSS), Brussels, Belgium, Jul. 2021, pp. 1234–1237.

[4] L. Giustarini, R. Hostache, P. Matgen, G. J.-P. Schumann, L. Pfister, and L. Hoffmann, "A change detection approach to flood mapping in urban areas using TerraSAR-X," *International Journal of Applied Earth Observation and Geoinformation*, vol. 21, pp. 52–65, Apr. 2013.

[5] S. Martinis and C. Rieke, "Backscatter analysis using multi-temporal and multi-frequency SAR data in the context of flood mapping at river saale, germany," *Remote Sensing*, vol. 7, no. 6, pp. 7732–7752, Jun. 2015.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, ser. Lecture Notes in Computer Science, vol. 9351. Munich, Germany: Springer, Oct. 2015, pp. 234–241.

[7] J. Zhao, Z. Xiong, and X. X. Zhu, "UrbanSARFloods: Sentinel-1 SLC-based benchmark dataset for urban and open-area flood mapping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Seattle, WA, USA, Jun. 2024, pp. 419–429.

[8] B. Ghosh, S. Garg, M. Motagh, and S. Martinis, "Automatic flood detection from Sentinel-1 data using a nested UNet model and a NASA benchmark dataset," *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 92, pp. 1–18, Mar. 2024.

[9] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel Squeeze & Excitation in fully convolutional networks," *arXiv preprint arXiv:1803.02579*, Mar. 2018.

[10] X. Wu, Z. Zhang, S. Xiong, W. Zhang, J. Tang, Z. Li, B. An, and R. Li, "A near-real-time flood detection method based on deep learning and SAR images," *Remote Sensing*, vol. 15, no. 8, p. 2046, Apr. 2023, published: 12 Apr. 2023.

[11] N. Mohamadiazar *et al.*, "Integrating deep learning, satellite image processing, and hydraulic modeling for near real-time flood mapping," *Journal of Hydrology*, vol. 635, p. 131025, Aug. 2024.

[12] F. Pech-May, R. Aquino-Santos, O. Álvarez-Cárdenas, J. Lozoya-Arandia, and G. Rios-Toledo, "Segmentation and visualization of flooded areas through Sentinel-1 images and U-Net," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 8996–9008, Apr. 2024.

[13] A. Jamali, S. K. Roy, L. H. Beni, B. Pradhan, J. Li, and P. Ghamisi, "Residual wave vision U-Net for flood mapping using dual polarization Sentinel-1 SAR imagery," *International Journal of Applied Earth Observation and Geoinformation*, vol. 127, p. 103662, Mar. 2024.

[14] R. Yadav, A. Nascetti, and Y. Ban, "Attentive dual stream siamese U-Net for flood detection on multi-temporal Sentinel-1 data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Kuala Lumpur, Malaysia, Jul. 2022, pp. 5222–5225.

[15] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, Jun. 2016.

[16] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, St. Petersburg, FL, USA, Dec. 2017, pp. 1–4.

[17] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, Korea, Oct. 2019, pp. 1314–1324.

[18] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention U-Net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, Apr. 2018.

[19] Z. Wang, X. Wang, and G. Li, "An extremely lightweight U-Net with soft fusion for flood detection using multi-source satellite images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Pasadena, CA, USA, Jul. 2023, pp. 6454–6457.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2980–2988.

[21] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, vol. 97. Long Beach, CA, USA: PMLR, Jun. 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[22] J. Karvonen, "U-net with ResNet-34 backbone for dual-polarized C-band baltic sea-ice SAR segmentation," *Annals of Glaciology*, vol. 65, p. e32, Nov. 2024.