



Generalization-Aware UAVSAR Flood Segmentation and FPGA Inference Feasibility

Alea Atkinson^{1*}, JJ Salter^{2*}, Joyce Zheng^{3*}, Imani Phillip⁴, Dr. Yuanwei Jin⁴, and Dr. Enyue Lu⁵

University of Florida¹, Johns Hopkins University², The University of Texas at Austin³, University of Maryland Eastern Shore⁴, Salisbury University⁵

*These authors contributed equally to this work.



Abstract

Flood mapping benefits from Synthetic Aperture Radar (SAR) imagery because it is largely unaffected by cloud cover and lighting conditions. Previous studies have applied deep learning to NASA's Uninhabited Aerial Vehicle SAR (UAVSAR) imagery for flood segmentation, but random tile-based splits may overestimate performance. This study evaluates U-Net flood segmentation using Leave One Flight Path Out (LOFPO) evaluation, withholding entire UAVSAR flight paths to assess generalization to unseen geography. LOFPO produced lower performance than random splits (mean Dice 0.6287 vs. 0.6974). We evaluate pretraining strategies to address this generalization gap. Weak SimSiam and MAE were the only self-supervised methods to modestly improve performance over random initialization, while weakly supervised pre-training consistently outperformed both baseline and self-supervised models, particularly with limited data and fine-tuning. An FPGA study showed that fixed-point arithmetic closely matched floating-point behavior in the first U-Net stage, although GPU inference remained faster at realistic tile sizes. These results demonstrate the importance of flight path based evaluation, the potential of automated labels for improving flood segmentation generalization, and the feasibility of fixed-point computation for future hardware acceleration.

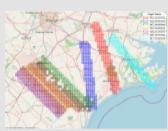
Introduction

Prior UAVSAR flood segmentation work demonstrates that deep learning models such as U-Net, Attention U-Net, and U-Net++ can segment flood extent from UAVSAR imagery with acceptable accuracy. However, much of the earlier evaluation relied on random train, validation, and test splits where tiles from similar locations could appear in both training and testing. Therefore, random tile splits may overestimate realistic deployment performance. In remote sensing literature, researchers investigate additional training methods to improve performance, including self-supervised learning (SSL) methods and weakly supervised pre-training using potentially inaccurate, noisy labels rather than verified manually annotated labels. Leveraging the abundance of unlabeled UAVSAR labels for generalization is relevant to flood mapping because it is time-sensitive. It may not be time efficient to manually annotate a large amount of UAVSAR images to create flood training maps, or to train traditional supervised segmentation models on an extended period. Therefore, on top of generalization to new flight paths in an ideal testing scenario, it is important to evaluate model performance in realistic deployment scenarios where reliably annotated data and training time are limited. The time sensitivity of flood mapping tasks motivates additional deployment considerations, including hardware inference feasibility. U-Net-style segmentation models rely heavily on convolution operations, which makes fixed-point FPGA analysis a natural way to study speed, resource use, and power tradeoffs for inference. In the literature we reviewed, we did not find prior work that synthesized a custom FPGA design for downstream UAVSAR flood segmentation using learned weights from a trained flood model. In this project, the FPGA work studies downstream FPGA feasibility for a trained UAVSAR flood-segmentation U-Net by evaluating fixed-point accuracy, VHDL verification, synthesis results, and architecture tradeoffs for one learned inference stage.

LOFPO Methodology

The model used for evaluation was a basic U-Net model, and the dataset for training was a Hurricane Florence UAVSAR flood segmentation dataset inherited from prior project work, containing 1067 valid UAVSAR and flood mask pairs. For each LOFPO fold, one flight path was held out as the test set, while the remaining flight paths were used to form training and validation sets. Any tile names overlapping with the held-out test path were removed from the training and validation pools.

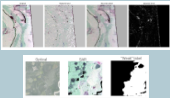
To compare LOFPO against traditional evaluation, we created random split baselines from the dataset. We evaluated model performance using Dice, intersection over union (IoU), precision and recall. We also evaluated the land-cover breakdown of all seven UAVSAR flight paths using the land cover data collected in previous project work. We then evaluated model performance across land cover types.



Pre-training Methodology

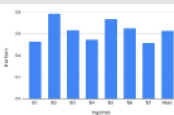
We processed 31 additional UAVSAR flight paths for pretraining using the same preprocessing workflow as the Florence dataset. For SSL, we tested methods that learn from unlabeled UAVSAR tiles, focusing on MAE and Weak SimSiam. MAE reconstructs masked image patches, while Weak SimSiam learns consistent features from SAR-safe augmented views of the same tile. For weakly supervised pre-training, we used automatically generated flood masks from OPERA/ARIA Milton products and change detection generated masks from prior project work. These masks were treated as weak labels because they were not expert-verified. We pretrained the same basic U-Net on 708 weakly labeled tiles for 20 epochs, saved the weights, and used them to initialize a new U-Net for Florence fine-tuning.

For both SSL and weakly supervised pre-training, we fine-tuned the initialized U-Net using the same Florence LOFPO splits as the baseline. We compared each pretrained model against a randomly initialized U-Net across matched training settings, including epoch budgets of 0, 1, 5, and 20 epochs; labeled-data fractions of 1%, 2%, 5%, 10%, 25%, 50%, 75%, and 100%; and three random seeds per setting.



LOFPO Results

We evaluated the basic U-Net model on each held-out flight path using global performance metrics and a threshold of 0.55, and performance varied substantially by held-out flight path. The highest metrics were on evaluation of the second flight path, with Dice 0.7852 and IoU 0.4844. The weakest performance was on flight path 7, with Dice 0.5432 and IoU 0.3461. Random split evaluation produced stronger results than strict flight-path evaluation. On the random split, our basic U-Net model achieved a Dice 0.6974, while LOFPO evaluation had an average dice of 0.6287.



Across land-cover analysis, trees had the highest Dice at 0.6746, followed by Rangeland at 0.5257. Bare Ground had low Dice at 0.1006. Water and Flooded Vegetation had Dice 0 and showed false-positive behavior.

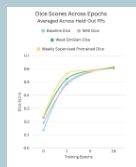
We also analyzed land cover per UAVSAR flight path to investigate possible reasons for performance variations. We found that fp7 was an outlier in water percentage, fp3 and fp6 had the highest rangeland percentages, and fp1, fp4, and fp5 had a high crop coverage.

Pre-training Results

To address the LOFPO generalization gap, we evaluated self-supervised and weakly supervised pre-training before U-Net fine-tuning. Among the SSL methods tested (MAE, denoising, contrastive learning, Weak SimSiam, and hybrid MAE + SimSiam) Weak SimSiam performed best, reaching 0.5960 mean Dice and 0.4250 mean IoU compared with 0.5832 Dice and 0.4133 IoU for random initialization. It achieved the highest Dice score on 5 of 7 flight paths, though gains were fold dependent. Overall, SSL helped only when the objective and augmentations were SAR-safe and well matched to the task.



We compared SSL initialization with weakly supervised pre-training using automatically generated flood labels. Across epoch-budget and data-fraction experiments, weakly supervised pre-training was the strongest overall strategy, especially when fine-tuning time or labeled data were limited. Weak SimSiam was still the strongest SSL method and improved label efficiency without generated masks, but the overall conclusion is that weak supervision from generated flood labels was the most reliable way to improve generalization under realistic constraints.



Hardware Methodology

Goal: Evaluate whether the first learned U-Net inference stage can be converted into fixed-point FPGA-style hardware without changing the trained model behavior.

Target stage: first Conv-BN-ReLU stage of trained U-Net; [32, 3, 3, 3] = 32 output channels, 3 input channels, 3x3 kernels

Workflow: Field BatchNorm into convolution weights and bias -> Convert stage to Q20 fixed-point arithmetic -> Generate Python golden outputs -> Verify VHDL against Python outputs in GHDL -> Synthesize architecture variants in Vivado on Artix-7 200T -> Compare resource use, timing, power, and GPU timing

Architecture variants: 32x direct computes all 32 output channels at once; 16-fold folded computes 16 channels at a time; 16-output stream is 16-output design plus tile/windowing input logic; 2-window*8-output computes 8 channels for 2 image windows at once; 2-lane shared reduces 2 multiplier lanes over many cycles

Hardware Results

Q20 fixed-point preserved the trained first U-Net stage closely.

Model-level impact: Largest Dice change across held-out paths: 0.00306 - Largest final mask change: 0.5423%

VHDL correctness: 512/512 golden-output checks passed; Max integer difference from Python Q20 checks = 32 output channels * 16 real-tile windows
Internal activation agreement = -68.064384 internal neural network values compared - Average error: 0.003189 - Pearson similarity: 0.999951 - Cosine similarity: 0.999986
The fixed-point VHDL version matched Python exactly on tested first-layer outputs and barely changed the final model map.

Hardware Results Continued

Architecture comparison for the first U-Net stage:
Cycles = effective cycles per 32-channel output
Energy/output = Vivado-estimated power * cycles at 100 MHz.

Design	Cycles	DSPs	LUTs	Power	Est. energy/output
32-output parallel	1	740/740	12,039	1.334 W	~13.3 nJ
16-output parallel	2	405/740	1,796	0.624 W	~12.5 nJ
16-output stream	2	405/740	30,660	0.940 W	~18.8 nJ
2-window*8-output	2	406/740	1,888	0.630 W	~12.6 nJ
2-lane shared	560	4740	2,531	0.176 W	~986 nJ

GPU comparison: 16-output streaming first-layer workload.

Patch	RTX time	FPGA time	RTX energy	FPGA energy
32x32	14.636 μ s	10.300 μ s	-1.700 μ J	-9.68 μ J
128x128	13.962 μ s	163.900 μ s	-1.622 μ J	-154 μ J
256x256	20.280 μ s	655.420 μ s	-2.355 μ J	-616 μ J

Energy uses ~116 W measured RTX 4080 board power and 0.940 W Vivado-estimated FPGA power.

Conclusion

Strict LOFPO evaluation revealed that UAVSAR flood segmentation performance drops when the model is tested on completely unseen flight paths, suggesting that random splits can overestimate real deployment performance. Performance varied substantially by held-out flight path, suggesting that geographic distance, land-cover composition, and UAVSAR texture differences all contribute to the generalization challenge. Pretraining helped reduce this gap, especially under realistic constraints where labeled data or fine-tuning time are limited. Weak supervision from automatically generated flood masks produced the strongest overall improvements, while SSL offered smaller and more conditional gains. Among SSL methods, Weak SimSiam was the most effective, suggesting that SAR-safe representation learning is more useful than reconstruction alone. Our results suggest that a practical rapid-response pipeline could pretrain models on unlabeled or cheaply labeled UAVSAR imagery, then fine-tune on a small amount of manually verified flood data. Future work should evaluate this approach across additional flood events with independent expert labels and expand the hardware feasibility study beyond the first U-Net layer.

Hardware study: Q20 fixed-point VHDL preserved the trained first U-Net stage; the 16-output FPGA estimate used lower energy than RTX 4080, but the FPGA was faster only on small patches.

Select References

K. Yang, C. A. P. Gonzalez, N. Lathauwer, S. Lee, Y. Shi, and S. Lu, "Change detection and cover classification of flooded regions in UAVSAR imagery using deep learning," in Proc. 2022 IEEE Int. Conf. Data Mining Workshops (ICDMW), Washington, DC, USA, Nov. 2022, pp. 4079-4084, doi: 10.1109/icdmw45920.2022.100494.

K. Yang, C. A. P. Gonzalez, N. Lathauwer, S. Lee, Y. Shi, and S. Lu, "Change detection and cover classification of flooded regions in UAVSAR imagery using deep learning," in Proc. 2022 IEEE Int. Conf. Data Mining Workshops (ICDMW), Washington, DC, USA, Nov. 2022, pp. 4079-4084, doi: 10.1109/icdmw45920.2022.100494.

Acknowledgements

We would like to express our sincere gratitude to Dr. Jin for his guidance and expertise in UAVSAR flood detection, and to Dr. Li for providing us with this opportunity. Special thanks to our fellow REU researchers, Ella, Caroline, Naisha, Juniper, Matthew, Jeffrey, and William for a spectacular summer in Salisbury. This work was funded by NSF CPE-2447041 under the Research Experiences for Undergraduates Program.