Comparing Image Segmentation Models for Flood Inundation Mapping

Sean Dixit
Department of Computer Science
Indiana University
Bloomington, US
sedixit@iu.edu

Aolong Li
Department of Mathematics
Indiana University
Bloomington, US
aolli@iu.edu

Vishakha Dikshit
Department of Computer Science
Indiana University
Bloomington, US
vidiksh@iu.edu

Yevhen Melnyk
Department of Mathematics
Indiana University
Bloomington, US
ymelnyk@iu.edu

***Abstract*— Flood inundation mapping plays a pivotal role in disaster management and mitigation efforts. Accurate identification and segmentation of flood pixels are crucial for assessing the extent of inundation and aiding timely response efforts. In this study, we present a comparative analysis of various segmentation models for flood pixel segmentation, focusing on their efficacy in flood inundation mapping tasks. The models we compare are: U-Net (2 approaches), LinkNet, IBM-NASA's Prithvi (2 approaches), and SegFormer.**

Keywords— image segmentation, satellite imagery, U-Net, Prithvi, Linknet, Segformer, encoder/decoder

I. Background and related works

Floods are among the most severe natural disasters, affecting more than 20% of the global population. In the realm of GeoAI, mapping flood inundation is treated as a semantic segmentation challenge, aiming to classify each pixel as either flooded or not flooded. The initial step in developing an effective model involves acquiring a dataset with precise annotations of flood areas. Saugat Adhikari, who initiated this project, established a framework for accelerating the annotation process through active learning. This method utilizes a backbone model to generate initial predictions, and an acquisition function recommends specific superpixels for human annotation and subsequent model retraining. Enhancing the efficiency of active learning may involve experimenting with various backbone models. Currently, popular methods like U-Net are employed, but newer models based on vision transformers, such as Segformer and IBM-NASA's Prithvi [4], have also been tested on the Sen1Floods11 dataset [5] and shown promising results [1].

Nevertheless, our objective is to develop and compare models that are aware of elevation, using high-resolution optical imagery from satellites and drones, as well as Digital Elevation Model (DEM) data, because elevation data is crucial in flood prediction. For instance, if a highly confident flooded pixel is surrounded by pixels at a lower elevation, there is a high likelihood that these neighboring areas are also flooded. Conversely, a dry pixel surrounded by higher elevation areas likely indicates dry surroundings. Therefore, we aim for our model to recognize and incorporate these elevation patterns into its predictions. The Sen1Floods11 dataset, previously evaluated, is distinct, containing six channels per pixel including RGB and various infrared data (narrow NIR, SWIR1, SWIR2). This necessitates a re-evaluation of model performance on our specific data format to ensure accuracy and effectiveness. We adapt well-known segmentation models to work on the 7-channel data and compare those results to the results we get on 6-channel data, along with comparing the results with a state of the art elevation-aware U-Net model, with hopes to outperform it.

II. Methods

A. Pre-processing Satellite Imagery

Our analysis utilizes high-resolution satellite imagery (1856x4104 pixels) of various regions, captured both before and after flood events, provided by Saugat Adhikari. We focus on five distinct regions for training and testing purposes. Each image pixel comprises seven channels, including RGB data from pre- and post-flood events, along with elevation information. To streamline machine learning processes, we pad these images to ensure that they can be subdivided into uniform patches of 128x128 pixels, facilitating consistent and efficient analysis. In terms of annotation, our approach differs from the Sen1Floods11 dataset, which categorizes labels into no water (class 0), water/flood (class 1), and no data/clouds (class 2). Instead, we employ a labeling system where Flood is designated as 1, Unknown as 0, and Dry as -1, providing a clear framework for our predictive modeling.

B. U-Net

The U-Net model was chosen for performance comparison due to its established reputation as a classic architecture for semantic segmentation, and served as the baseline model we tried to improve on. Moreover, it stands as one of the most widely adopted models in flood inundation mapping endeavors. The U-Net architecture is renowned for its effectiveness in capturing intricate spatial details, making it particularly suitable for tasks requiring precise delineation of flooded areas.

U-Net comprises two main components: an encoder and a decoder. The encoder section facilitates a contracting path,

wherein crucial image features are hierarchically extracted through convolutional layers. The spatial resolution progressively decreases through downsampling operations, thereby reducing the computational burden. In our experiments, the U-Net encoder includes four downsampling steps, executed through two convolutional layers for feature extraction and one max-pooling layer for downsampling. Our implementation also comprises many dropout layers, which helps reduce overfitting. The output from the encoder is then forwarded to the decoder component.

The decoder operates as an expanding path, aiming to restore the resolution of feature maps to match the original image dimensions. This process involves four upsampling modules, each comprising an interpolation upsampling layer followed by a convolutional layer, to reconstruct spatial details effectively.

A distinguishing characteristic of U-Net is the incorporation of skip connections. These connections establish links between decoder outputs and corresponding features generated during the encoding phase. This architectural design enhances segmentation accuracy by facilitating the integration of both local and global context information.

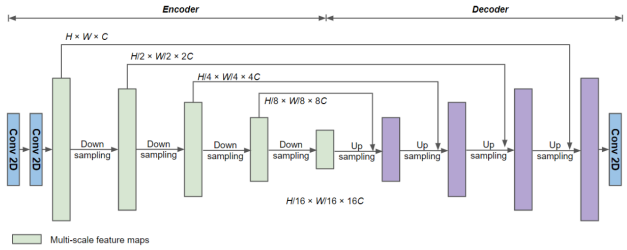

Fig. 1. The U-Net Architecture (adapted from [1]). H stands for height. W stands for width. C stands for channel.

In addition to the standard U-Net architecture, we employ a variant with an EfficientNet-B5 backbone. EfficientNet is known for its balance between accuracy and computational efficiency, making it suitable for resource-constrained environments and large-scale segmentation tasks. The variant with EfficientNet B5 as the backbone features five encoder and decoder layers, with each layer halving the spatial dimension. This model is employed via pytorch's segmentation_models library [2].

C. *LinkNet*

Linknet is a convolutional neural network architecture designed for semantic segmentation tasks. It features an encoder-decoder structure with skip connections, similar to U-Net, but with modifications aimed at improving efficiency and performance.

In Linknet, the encoder consists of a pre-trained convolutional neural network which extracts high-level features from input images. In our case, we use a pre-trained EfficientNet-B5 model as the backbone. These features are then passed through a series of convolutional and upsampling layers in the decoder to generate segmentation masks. Skip connections are incorporated between corresponding encoder and decoder layers to enable the fusion of low-level and high-level features, aiding in accurate segmentation. In our implementation utilizing PyTorch's segmentation_models library, the Linknet model is instantiated with a default configuration comprising four skip connections [2].

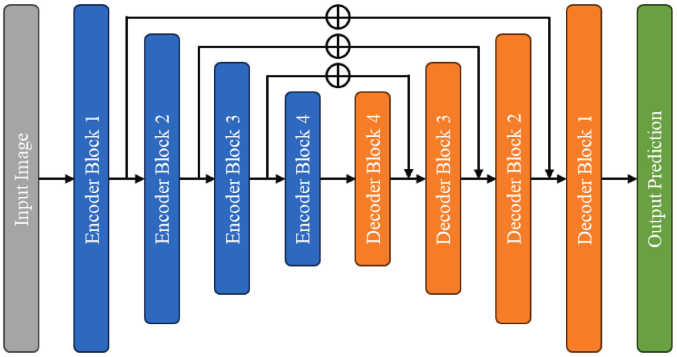

Fig. 2. The LinkNet Architecture (adapted from [3])

D. *IBM-NASA's Prithvi*

The Prithvi model, developed by IBM-NASA, is a novel approach to flood inundation mapping that uses a temporal Vision Transformer (ViT) architecture trained on 30-meter Harmonized Landsat Sentinel-2 (HLS) data encompassing the United States [1]. Unlike traditional ViT models that are trained on single time slice images, Prithvi is tailored to process time-series satellite imagery, incorporating an additional time dimension into its input.

The model's training strategy employs a Masked AutoEncoder (MAE), wherein the model learns to predict masked patches from the original image through self-supervised learning. This method ensures that both spatial, positional, and temporal information are captured within the resulting embeddings. The embeddings are then fed into the transformer encoder to capture inherent spatial and temporal dependencies.

To adapt the model for downstream tasks like image segmentation, a segmentation head is appended to the model. In the flood inundation mapping context, the integrated semantic segmentation pipeline comprises a pre-trained encoder module and a decoder responsible for pixel-wise image segmentation. The decoder deserializes the encoded vectors to create two-dimensional feature maps, which are then upsampled through four 2D transposed convolutional layers to improve segmentation precision. A final 2D convolutional layer is added to produce the ultimate predictions, classifying pixels into three categories: flood, non-flood, and unknown.

A distinguishing feature of the Prithvi model is its requirement for six input bands, including red, green, blue, narrow Near InfraRed (NIR), Short-Wave Infrared 1 (SWIR1), and Short-Wave Infrared 2 (SWIR2). SWIR1 and SWIR2 bands, with different wavelengths, are particularly valuable for discerning moisture content in soil and vegetation [1]. While the model is typically trained on datasets containing all six bands, adaptations are made when certain bands are unavailable. In our case, lacking the last three bands in our

dataset, we opt to train the model on our data rather than relying on pre-trained versions.

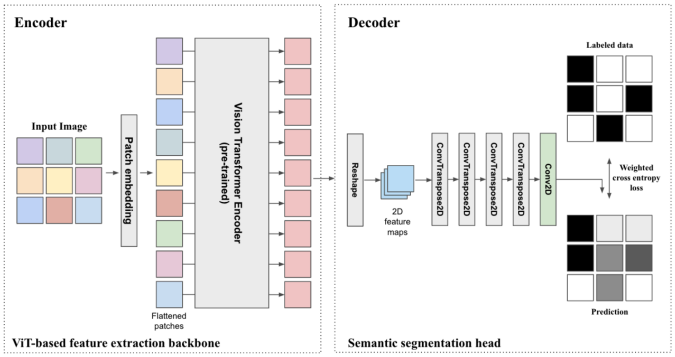

Fig. 3. The Prithvi Encoder-Decoder Architecture (adapted from [1])

We employ two models for comparison: the full Prithvi encoder-decoder model, and a model with Prithvi encoder as the backbone and Unet decoder. By replacing Unet's encoder with the Prithvi encoder, we preserve Unet's semantic segmentation capabilities while enhancing its temporal awareness, resulting in a hybrid model that should effectively capture both spatial and temporal features relevant to flood mapping. We modified Pytorch implementations of the Prithvi encoder and decoders along with the entire models to work for our 7 and 6 input channel data, and the original model implementations can be found here: https://github.com/isaaccorley/prithvi-pytorch.

E. SegFormer

Segformer adopts an encoder-decoder architecture, differing from U-Net by employing a transformer-based feature extraction backbone. Segformer introduces hierarchical feature representation in its encoder to generate multi-level features. These features retain both high-level features at low resolution and low-level features at high resolution, enabling comprehensive feature extraction across different scales. To enhance computational efficiency, Segformer incorporates Efficient Self-Attention in its transformer blocks, which efficiently scales down the number of input sequences, thereby reducing computational overhead. Additionally, it adopts Mix-FFN (Feed Forward Network) for data-driven positional encoding, using a 3-by-3 convolutional layer within the FFN to encode positional information. This adaptation mitigates performance degradation caused by positional encoding variations in test images with different resolutions. Segformer also partitions images into overlapping patches to maintain data continuity and local context, which are then merged before being processed by subsequent transformer blocks.

The Segformer decoder leverages the large effective receptive field of the transformer encoder to fuse multi-level features. It employs a lightweight multi-layer perceptron (MLP) module exclusively consisting of MLP components for generating segmentation masks, which substantially reduces computational demands.

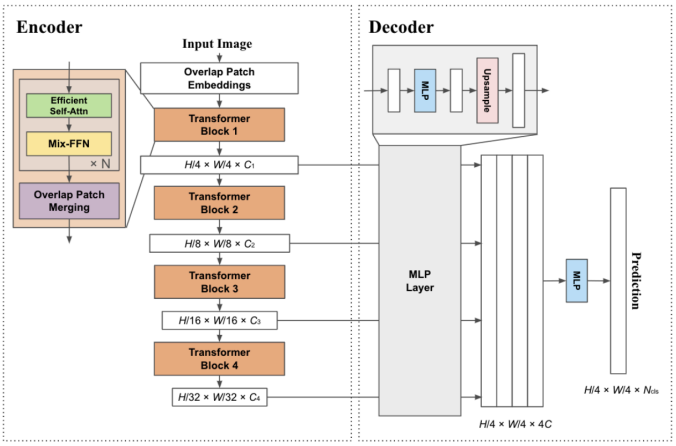

Fig. 4. The SegFormer Architecture (adapted from [1]). H stands for height, W stands for width, C stands for channel.

We modified the Pytorch implementation of SegFormer to make it work for our 6 input channel data. Despite our efforts, we encountered challenges when attempting to extend its functionality to handle 7 input channel data. The original implementation can be found here: https://github.com/lucidrains/segformer-pytorch.

F. Experiments

In our experimental setup, we aim to evaluate the performance of six different models for flood inundation mapping across various configurations. The models considered are Unet, Unet with EfficientNet backbone, Linknet, Prithvi, Unet with Prithvi encoder backbone, and Segformer. Each model is trained for 100 epochs using three different optimizers: SGD, Adam, and AdamW. Out of the 100 epochs, we choose the model that gives us the minimum loss. We observe that SGD consistently yields the best performance across all models and hence, is selected for further analysis.

For faster convergence, larger learning rates are employed for Unet and Linknet models due to their simpler architectures compared to Segformer and Prithvi. Detailed experimental settings are provided in Table 1.

Params	U-Net	U-Net (EfficientNet B5 Backbone)	LinkNet	Prithvi	Unet (Prithvi Encoder Backbone)	Segformer
Optimizer	SGD	SGD	SGD	SGD	SGD	SGD
Learning rate	2e-6	2e-6	2e-6	1e-7	1e-7	1e-7
Batch size	4	4	4	4	4	4
Loss function	Cross Entropy	Cross Entropy	Cross Entropy	Cross Entropy	Cross Entropy	Cross Entropy
Cropped image size	128x128	128x128	128x128	128x128	128x128	128x128

Table 1. Experimental settings for the 6 models used in the experiment.

For our experimentation, we test each model on two channel modes: one with six channels, representing pre-flood RGB and post-flood RGB values, and another with all seven channels, including elevation band data. We test the SegFormer on only the prior channel mode as it was unable to work on 7-channel data, as mentioned before. The dataset is

6-Channel Model	Avg. mIoU	Avg. IoU		Avg. mRec (%)	Avg. Recall (%)		Number of trainable parameters
		Flood	Non-flood		Flood	Non-flood	
U-Net	0.70	0.75	0.65	86.44	75.72	97.16	122K
U-Net (EfficientNet Backbone)	0.71	0.83	0.58	79.09	99.97	58.20	31M
LinkNet	0.60	0.78	0.41	70.55	99.96	41.13	29M
Prithvi	**0.88**	0.92	0.83	**91.95**	98.03	85.87	112M
U-Net (Prithvi Backbone)	0.84	0.90	0.78	89.55	98.33	80.77	115M
Segformer	0.83	0.89	0.76	88.82	96.82	80.82	20M

Table 2. Models trained and tested on 6 channel satellite imagery. Channels: RGB pre-flood (3) and RGB post-flood (3).

divided into 4 regions for training/validation and one region for testing.

The final results are presented using many metrics obtained from the experiments, including Intersection over Union (IOU), Mean Intersection over Union (mIOU), Recall (Rec), and Mean Recall (mRec). By focusing on recall, we emphasize the importance of correctly identifying flooded areas, which could be critical for disaster management and response efforts. The equations 1-4 provide their equations, with TP being True Positive, FP being False Positive, and FN being False Negative. The equations were taken from [1].

$$IoU = \frac{TP}{TP + FN + FP} \quad (1)$$

$$mIoU = \frac{IoU_{flood} + IoU_{nonflood}}{2} \quad (2)$$

$$Rec = \frac{TP}{TP+FN} \quad (3)$$

$$mRec = \frac{Rec_{flood} + Rec_{nonflood}}{2} \quad (4)$$

III. Results

A. Model Comparison on 6-Channel Data

We conducted experiments by training and testing each network model on 6-channels of satellite images first. Table 2 presents our results, illustrating the performance metrics for flood inundation mapping for 6-channel data across different models. We also plotted the binarized prediction segmentation maps, where flooded regions are marked by yellow pixels, non-flooded regions are marked by purple pixels, and unknown regions are marked by blue pixels.

From the results, it's evident that models utilizing the Prithvi architecture, including Prithvi itself and U-Net with Prithvi backbone, consistently outperform other models across various metrics. These models achieve higher mIoU and Recall scores, with the Prithvi model getting the highest mIoU at 0.88, along with the highest average mRec at 91.95%. Segformer also performed much better than the U-Net, U-Net with EfficientNet and LinkNet models, getting 12%+ more average mIoU than the U-Net w/ EfficientNet backbone model.

The U-Net and EfficientNet backbone U-Net models performed very similarly, with ~1% difference in average mIoU, though their average mRec do differ significantly, with the U-Net having about 7% higher average mRec. Comparing the two to the U-Net with Prithvi backbone, it is clear that the Prithvi backbone gives much better performance for 6-channel data.

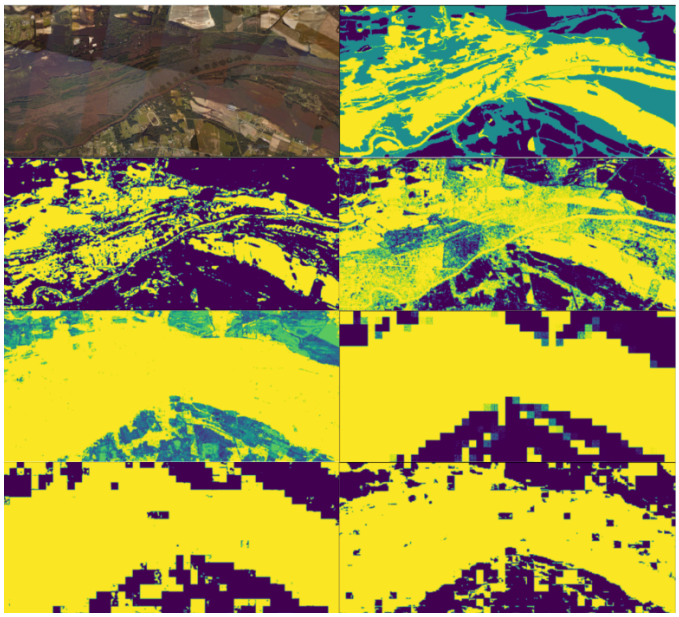

Fig. 5. Unseen satellite image, groundtruth and predictions made by each model on the **6-channel data**. From top-left to bottom-right, going row by row, we have: Unseen satellite image region, Groundtruth, U-Net, U-Net with EfficientNet backbone, LinkNet, Prithvi, U-Net with Prithvi backbone, SegFormer. Yellow: flooded, Purple: non-flooded, Blue: unknown.

In Figure 5, we see the model prediction results on an unseen test region. We see that the U-Net model overestimated the number of non-flooded pixels (marked by purple pixels). We also notice that the LinkNet model is not very confident about the non-flooded regions that are classified as non-flooded, as we see many unknown pixels in the non-flooded regions. Furthermore, we notice that the Prithvi models create segmentation masks that are square in nature, which is the cause of the ViT encoder in Prithvi that uses patches (of size 16x16) as input during training. We do notice the U-Net with EfficientNet model predicts the Unknown pixels (marked by blue pixels) quite well compared to the other models. Interestingly, the Prithvi model outperformed the other models in terms of IoU and mRec, even though it consists of square segmentation blocks.

B. Model Comparison on 7-Channel Data

However, the models' performance is quite different for the 7-channel data, i.e considering elevation data. From the results shown in Table 3, it's evident that the U-Net model achieves the highest overall segmentation performance across all measures despite having the least number of parameters

7-Channel Model	Avg. mIoU	Avg. IoU		Avg. mRec (%)	Avg. Recall (%)		Number of trainable parameters
		Flood	Non-flood		Flood	Non-flood	
U-Net	**0.89**	0.93	0.86	**92.08**	99.41	84.75	2.8M
U-Net (EfficientNet Backbone)	0.62	0.79	0.45	63.38	99.96	26.79	29M
LinkNet	0.56	0.76	0.36	76.42	99.48	53.15	29M
Prithvi	0.50	0.68	0.30	54.97	99.65	10.29	113M
U-Net (Prithvi Backbone)	0.53	0.69	0.38	61.20	99.54	22.75	115M
Segformer	-	-	-	-	-	-	-

Table 3. Models trained and tested on 7 channel satellite imagery. Channels: RGB pre-flood (3), RGB post-flood (3) and elevation.

(which speaks about its light weightness). Both the Prithvi and U-Net with Prithvi backbone performed much worse than the U-Net, getting ~40% less average mIoU and more than ~30% less average mRec. It is likely that the model architecture is not appropriately designed to handle high-dimensional data or to effectively integrate elevation information with other features as it struggles to extract useful insights from the elevation data. LinkNet does a little better, getting 76.42% average mean Recall, but still much worse than the U-Net model. Interestingly, the U-Net model with EfficientNet backbone also performed very badly compared to the first U-Net model, though it outperformed the Prithvi backbone U-Net model slightly.

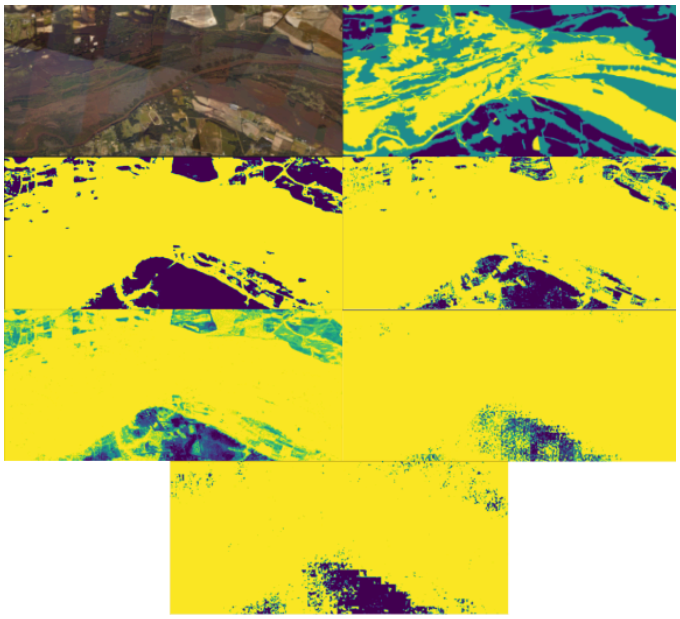

Fig. 6. Unseen satellite image, groundtruth and predictions made by each model on the **7-channel data**. From top-left to bottom-right, going row by row, we have: Unseen satellite image region, Groundtruth, U-Net, U-Net with EfficientNet backbone, LinkNet, Prithvi, U-Net with Prithvi backbone. Yellow: flooded, Purple: non-flooded, Blue: unknown.

In Figure 6, we see that both the Prithvi and U-Net with Prithvi backbone both overestimate the number of flood pixels and underestimate the number of non-flooded pixels significantly. From the data that the model was trained on, perhaps the models overestimated the effect of elevation on flood inundation, as each channel is given the same weight. Once again, we notice that the LinkNet model is not very confident about the non-flooded regions that are classified as non-flooded.

Interestingly, the difference between the average mIoU and average mRec of the U-Net that uses 7 channels as input and the Prithvi model that uses 6 channels as input is less than 1%. The U-Net model, however, clearly takes the cake, as not only does it have the better average mIoU and mRec scores, but it also has ~40 times lesser number of trainable parameters, which makes it a highly effective, light weight model for flood inundation mapping given elevation data.

IV. Discussion

This study, while insightful, encountered several limitations that must be acknowledged. A primary constraint was the limited amount of training data available, which restricted the models' ability to generalize across diverse flood scenarios. Limited data often lead to overfitting, where models perform well on training data but poorly on unseen data. Additionally, the complexity of integrating elevation data into flood prediction models presented substantial challenges. The study revealed that the U-Net model (without EfficientNet backbone) could adapt more flexibly to this integration, whereas more complex models such as Prithvi and LinkNet underperformed. This was particularly evident in the models utilizing the Prithvi architecture, which, despite showing promising results in 6-channel evaluations, struggled significantly with the 7-channel data that included elevation information. Another challenge was related to the architectural differences among the models. For instance, the patch-based training of the Vision Transformer models led to segmented outputs that were square in nature, potentially limiting the models' utility in real-world applications where flood boundaries are irregular and complex.

V. Conclusion

Our work gives a comparison of multiple state of the art segmentation models and shows that the U-Net model performs the best for satellite data with an additional elevation channel, and Prithvi performs the best without the 7th channel. The study also demonstrates the potential for hybrid models, such as the U-Net with Prithvi backbone, which blend traditional convolutional approaches with modern transformer-based methods. These hybrid models represent a promising direction for future research, potentially combining the strengths of different architectural paradigms to enhance flood prediction accuracy. The results of this study highlight

the critical role of model selection in flood inundation mapping. Future research could focus on expanding the dataset size and diversity to better train and evaluate the models under more varied environmental conditions. This would help in understanding the models' robustness and scalability across different geographic regions and flood types. Furthermore, one could also perform further hypertuning of the models and tweaks to the architectures to see if their performance could improve. Additionally, segmentation networks specifically designed to handle multi-dimensional data effectively, such as DeepLabv3 [6] and Stacked Hourglass [7], could give promising results for flood inundation mapping.

Another promising avenue is the enhancement of models' ability to integrate and interpret elevation data alongside spectral data, potentially through the development of specialized neural network layers or preprocessing techniques that better capture the relationship between elevation and flood likelihood. While this study lays a foundational understanding of various models' performance in flood inundation mapping, it also opens up numerous pathways for further enhancement and innovation in disaster management technology.

References

[1] W. Li., H. Lee., S. Wang, C. Y. Hsu, S. T. Arundel, Assessment of a new GEOAI Foundation model for flood, (n.d.). https://arxiv.org/pdf/2309.14500.pdf

[2] Qubvel. (n.d.). Qubvel/segmentation_models.Pytorch: Segmentation models with pretrained backbones. pytorch. GitHub. https://github.com/qubvel/segmentation_models.pytorch/tree/master

[3] G. Pedrero, A. Garc, A. I. Olano, J. M. Hidalgo, M. Lillo-Saavedra, M. Gonzalo, C. Caetano, C. Calder, S. Ram (2019, October 24). Convolutional neural networks for segmenting xylem vessels in stained cross-sectional images - neural computing and applications. SpringerLink. https://link.springer.com/article/10.1007/s00521-019-04546-6

[4] Jakubik, Johannes, Sujit Roy, C. E. Phillips, Paolo Fraccaro, Denys Godwin, Bianca Zadrozny, Daniela Szwarcman et al. "Foundation models for generalist geospatial artificial intelligence." arXiv preprint arXiv:2310.18660 (2023). https://arxiv.org/abs/2310.18660

[5] D. Bonafilia, B. Tellman, T. Anderson and E. Issenberg, "Sen1Floods11: a georeferenced dataset to train and test deep learning flood algorithms for Sentinel-1," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA, 2020, pp. 835-845, doi: 10.1109/CVPRW50498.2020.00113.

[6] Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017, December 5). Rethinking atrous convolution for Semantic Image segmentation. arXiv.org. https://arxiv.org/abs/1706.05587

[7] Newell, A., Yang, K., & Deng, J. (2016, July 26). Stacked Hourglass Networks for Human Pose Estimation. arXiv.org. https://arxiv.org/abs/1603.06937