# Automatic Segmentation of River Channels from Satellite Imagery

Brian Amaro
Stanford Univeristy
bamaro@stanford.edu

Vivek Vajipey
Stanford Univeristy
vvajipey@stanford.edu

Michael Hasson*
Stanford Univeristy
mhasson@stanford.edu

## Abstract

*In order to understand how river channels evolve in different climate conditions, a global mapping of river channels is required. This project produces a methodology that takes in satellite imagery of river basins to produce a segmentation map of the river channels. This study compared the results of aggregating predictions from binary classification models as well as binary segmentation models. We designed and implemented a novel pipeline to convert satellite imagery of entire basins into binary masks, showing where river channels are located. After a thorough evaluation and analysis of binary classification and binary segmentation models, we have shown promising results that these models can be used for river channel mapping. Overall, the binary segmentation method performed better than the binary classification method, both quantitatively and qualitatively.*

## 1. Introduction

How the morphology of river channels varies in response to climate and vegetation factors is a problem in the geosciences that is not well understood. As a subset of the study occurring within the Stanford Earth and Planetary Surface Processes Lab, this project will specifically build a river channel detector in order to understand how rare single threaded rivers are in unvegetated regions. Vegetation has classically been assumed to be required in order to provide the bank strength needed for channel stability [4], however, various examples of single threaded rivers can be found in arid basins [5]. Yet, the prevalence of single threaded rivers is poorly understood. A comprehensive study and geospatial mapping of such river channels would aid in discovering common patterns and characteristics of these single threaded rivers, leading to a better understanding of their formation mechanisms. In order to conduct such a large-scale survey in an efficient, timely, and reproducible manner, an automated method to map river channels is needed. As such, this project aims to develop a deep learning method that automatically segments river channels from satellite imagery.

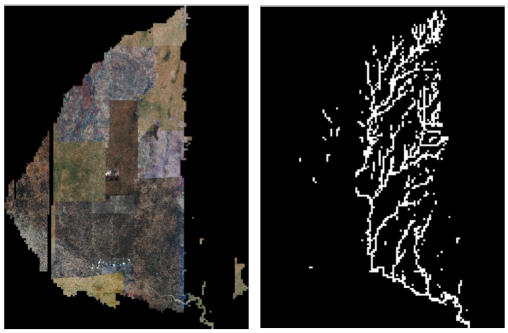

Figure 1. An example basin and corresponding binary classification mask

The input to this method is satellite imagery of river basins and the output is a segmentation mask showing where the river channels are. An example of an input-output pair is shown in Figure 1.

## 2. Related Work

Automatic extraction of river channel networks from landscapes has long been the subject of research in geomorphology. Traditionally, most methods have relied on topography to route flow down a path of steepest descent between adjacent pixels (e.g., [2]; [13]; [11]). This method is useful for outlining watersheds, but coarse pixel resolution of both elevation data and imagery limits the scale of features that can be analyzed (e.g., 30 m-resolution topographic or imagery data render channels narrower than 30 meters undetectable). Furthermore, steepest-descent algorithms tend to perform poorly in very low-sloping regions with subtle topography. Whereas these constraints do not pose problems when looking at, for example, continental-scale drainage divides, they cannot be used to resolve small-scale channels that are common in many

*Note that Michael Hasson is a non-CS231N contributor from the Stanford Earth and Planetary Surface Processes Lab.

basins. Other studies tackled the problem using water content indices such as the modified normalized difference water index (MNDWI) [16] derived from multispectral satellite imagery such as Landsat ( [6]; [10]). This method is best at detecting large, perennial bodies of water but is not viable for our study which requires the detection of rivers in arid regions that may contain water for a few days in a year. Other non-machine learning computer vision techniques such as active contour models have also been used to segment rivers based on their shapes [12].

Several machine learning approaches have been developed in recent years. Most studies appear to utilize binary semantic segmentation models, typically U-Net, that are used with synthetic-aperture radar (SAR) data [14]; [8]). The current state-of-the-art appears to be the model from Pai et al which is able to segment rivers in the Arctic, even from panchromatic (i.e. visual) satellite imagery. It cleverly uses water indices in order to generate an abundance of training data. However, this reliance on water indices, as well as the lack of geographic and climatic diversity in its training set, prevents the model from being able to predict river channels around the globe as needed in our project. Other studies attempt to expand on or develop new architectures such as a separable residual in [15] and an attention-based residual network in this study. The latter takes advantage of the separate modules so that "the shallow feature and large-scale attention module are used to locate the main position of the river, while the deep feature and small-scale attention module are responsible for the fine segmentation of the river edge".

## 3. Data

Our imagery was derived from Google Earth basemaps, which comprise processed, 25-50 cm-spatial resolution Maxar WorldView-3 and WorldView-4 imagery (Maxar Technologies/NASA CSDA; ) [3]. Watersheds were delineated by HydroSHEDS boundaries, which are derived from 90 m Shuttle Radar Topography Mission (SRTM) elevation data [7] Michael Hasson from the Stanford Earth and Planetary Surface Processes Lab created our training dataset using imagery from watersheds that encompassed the full range of dry and wet single and multi-threaded rivers from each of the five global aridity classifications (defined by [17]). He also conducted the preprocessing that involved splitting the basins into tiles. This processing (as well as later mosaicing) was done using the Geospatial Data Abstraction Library (GDAL) package. The size of the tile is made to be roughly 3 times the size of the largest river channel in the basin (and therefore varies per basin with tiles ranging from 375 x 375 to 3000 x 3000 pixels). This process produced our dataset of 59,554 tiles. These tiles were then labeled as either containing a river channel or not containing a river channel. In this labeled dataset, 8,297 tiles contain a river channel and 51,257 do not, revealing a clear class imbalance. Examples of these tiles can be seen in Figure 2.

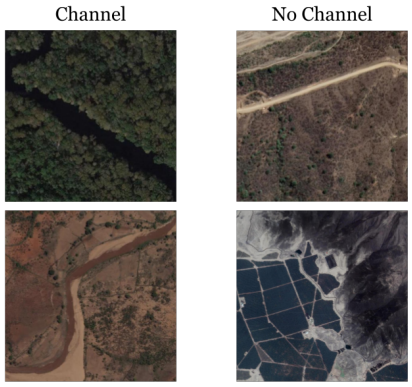


Figure 2. Comparison of basin tiles with and without channels

### 3.1. Datasets

We split our dataset into two different configurations for the sake of different experiments. The first configuration is the tilewise split, where 60% of the tiles are in the train set (35,738 tiles), 20% of the tiles are in the validation set (11,914 tiles), and 20% of the tiles are in the test set (11,902 tiles). This configuration allows us to train the model with data from all of the basins with the idea that this will aid in the model's generalizability. The second configuration is the basinwise split where all the tiles for a given basin are in the same split (train, validation, or test set) and the basins are distributed across the splits to match the 60-20-20 split as close as possible. This resulted in 36,491 tiles in the training set, 5,454 tiles in the validation set, and 16,605 tiles in the test set. This configuration allows us to see how the model generalizes to unseen basins.

## 4. Methods

We approach the problem (of river channel mapping) in two distinct methods, each producing a binary segmentation map of a given river basin. As shown in the flowchart in Figure 2, the binary classification method predicts one tile at a time while the binary segmentation method predicts on a larger "megatile" composed of 64 individual tiles, allowing for more robust predictions and for more spatial information to be considered in its prediction.

Note that due to hardware limitations and the variation in tile resolutions, the input images were resized from their

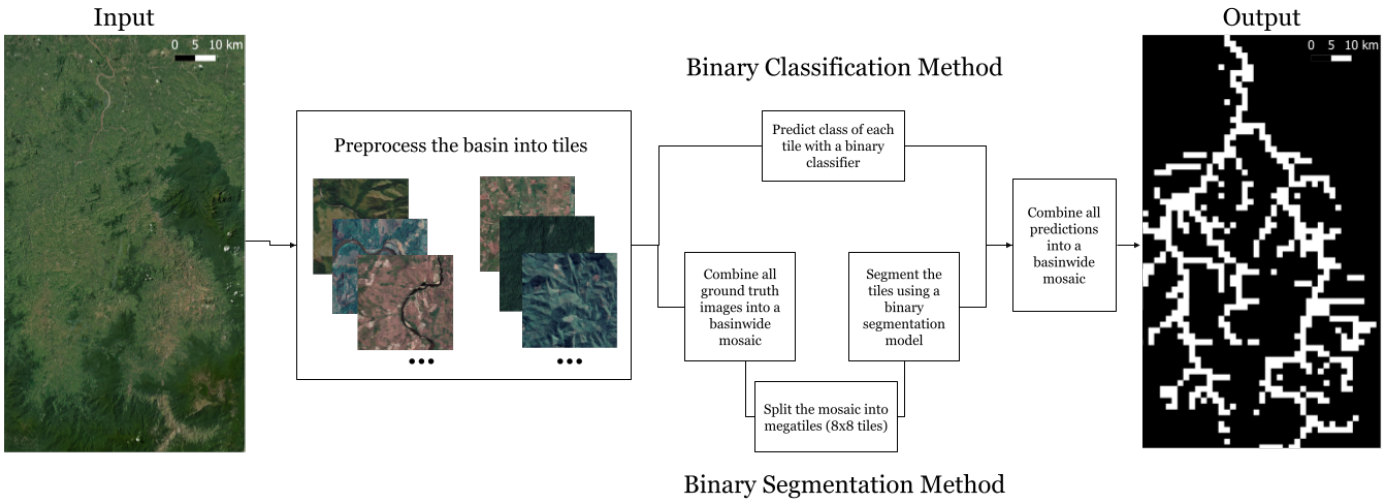


Figure 3. Flowchart depicting input to output pipeline

native resolution to 128 $\times$ 128 pixels for the binary classification method and to 64 $\times$ 64 pixels for the binary segmentation method.

## 4.1. Method 1: Binary Classification

### 4.1.1 Baselines

To establish performance benchmarks for our model, we implemented several baseline methods.

As an upper expectation of model performance, we first created a human manual classification baseline. We had the authors who did not create the dataset first review the mapping guidelines for classification (e.g. a river channel must be observably depositional with bars present) and then spend 4 minutes studying randomly selected image tiles from the training and validation sets (1000 tiles that had a river channel and 1000 tiles that did not have a river channel). Each author had to then label the same 100 images. We averaged the performances between the two authors that took place in this part of the study. Our other baselines serve as simple approaches that serve as a lower baseline of performance (and are often more interpretable methods).

Our most naive baseline is an input-independent model that simply predicts the majority class (no river channel present) for all inputs. This was included as a baseline to show how the essentially simplest model can still have a high accuracy due to the major class imbalance and is thus useful to understand what a good accuracy is in this problem.

One simple algorithmic approach for the task of binary classification is using a $k$-Nearest Neighbor Classifier (KNN). When predicting whether a tile contains a river, the classifier determines the top $k$ tiles in the training set that are nearest to the input tile based on a specified metric. The classifier then predicts the input tile to be the majority class of the $k$ nearest tiles. For the baseline testing, we tested the KNN with $k$ values of 3 and 5, using either only pixels values as the features, or using the Histogram of Oriented Gradients (HOG) as the features. The Histogram of Oriented Gradients is a feature descriptor that captures shape and texture by counting gradient orientations in an image. This method is useful in our scenario as it can detect the linear texture patterns common to river channels.

The other baseline involved training a convolutional neural network (CNN) with a simple architecture without any pretrained weights. Convolutional Neural Networks (CNNs) are ideal for image classification because they learn spatial hierarchies of features; their layered structure allows them to identify different features at various levels, from edges and corners to more complex structures, such as river channels in our case. The CNN used for this baseline consisted of sequential layers including two convolutional layers with ReLU activation, max pooling layers, a flatten layer, followed by two dense layers with ReLU activation and dropout, and a final sigmoid activation output layer for binary classification. Considering the simplicity of

the model, we expected that it would detect linear features and broad-scale textures but not the intricacies that differentiate, e.g., depositional river channels from incisional river channels, or river channels from roads.

#### 4.1.2 Main Models: Pretrained ResNets

In addition to the baselines, we explored the power of deeper networks and transfer learning by employing Residual Networks (ResNets) - specifically, ResNet-18 and ResNet-50 architectures - pre-trained on the ImageNet dataset. ResNets introduce the concept of residual learning by incorporating skip connections to allow the gradient to be directly backpropagated to earlier layers. This key architectural innovation combats the problem of vanishing gradients which becomes exacerbated in deeper networks. Consequently, ResNets can effectively learn complex patterns and hierarchies of features, a necessary condition for satellite image analysis where context and feature structure play a crucial role.

Transfer learning with ResNets pre-trained on the ImageNet dataset leverages the notion that ImageNet weights can serve as a universal feature extractor. Though ImageNet contains images of diverse objects and animals, its learned weights encapsulate a broad range of features including textures, edges, and shapes that could be beneficial for our task. In essence, these pre-trained models have already learned to discern numerous low and mid-level features, and we can repurpose these learned feature maps for our task of identifying river channels. For our task, these models were fine-tuned on our dataset. The final classification layer was replaced with a new layer suitable for binary classification, and the entire model was trained on our satellite image dataset. We employed both ResNet-18 and ResNet-50 architectures to experiment with varying model capacities. ResNet-18, being shallower, has less computational complexity and is less likely to overfit, making it suitable for a relatively smaller dataset. On the other hand, ResNet-50, having more layers, has the potential to learn more complex features, and thus, might exhibit superior performance on larger or more complex datasets. By evaluating both, we aim to find a balance between model complexity and performance.

We did not freeze ImageNet weights during ResNet training. Despite ImageNet's diverse features, its content significantly differs from our satellite imagery dataset. Thus, allowing further training of these weights lets the model learn satellite-specific features. Moreover, our substantial dataset size reduces overfitting risk that might arise when fine-tuning the entire model on a smaller dataset. Thus, by not freezing the ImageNet weights, we could use transfer learning while customizing the model for satellite image classification.

### 4.2. Binary Segmentation

The models used for binary segmentation include U-Net and FPN (Feature Pyramid Network). Both utilize several convolutional neural network layers that downsample an image in order to achieve semantic understanding and then upsample the image in order to match this understanding to the corresponding locations (i.e. pixels) in the image. Both models utilize ResNet backbones that were pre-trained on the ImageNet dataset.

## 5. Experiments

### 5.1. Hyperparameters

We experimented with various initial learning rates in the range of 1e-2 to 1e-4 for the different ResNet models trained on distinct datasets. Our observation indicated that learning rates near 1e-3 typically led to the most effective convergence. We selected the Adam optimizer due to its ability to dynamically adjust learning rates and leverage momentum, resulting in faster convergence and improved training performance. Adam calculates an exponential moving average of both the gradient and its square, then adjusts these averages to counter the zero-bias issue, thereby deriving the effective learning rate. Our selection of batch sizes was guided by hardware constraints, but we chose as large of a batch size as possible to ensure each batch represented a sizeable subset of the dataset. For binary classification, we used batch sizes of 512 and 64 for ResNet-18 and ResNet-50, respectively, and for the segmentation models, 16 for ResNet-18 and 8 for ResNet-50.

### 5.2. Primary Metrics

The primary metrics used include accuracy, precision, recall, F1-score, AUC-ROC, and AUC-PR. Additionally, IoU was used to evaluate the segmentation models. Due to the class imbalance in our datasets, accuracy alone can be misleading; thus, we incorporated metrics that provide insights into different aspects of the model's performance.

- **Precision:** Precision is calculated as the ratio of true positives to the sum of true positives and false positives. It indicates the model's ability to avoid falsely identifying a non-river tile as a river tile. High precision is crucial in our task as it ensures that the mapped areas indeed represent river channels.

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

- **Recall:** Recall is the ratio of true positives to the sum of true positives and false negatives. It signifies the

model's ability to correctly identify all river tiles. In the context of basin mapping, high recall ensures that no potential river channels are overlooked.

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

- **F1-Score:** The F1-score is the harmonic mean of precision and recall, providing a metric that incorporates both the frequency of false positives and false negatives.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **AUC-ROC:** The AUC-ROC (Area Under the Receiver Operating Characteristic Curve) is calculated by plotting the true positive rate against the false positive rate at various thresholds, and then integrating the resulting curve. A value of 1 indicates perfect classification, where the model correctly identifies all positive and negative instances. A value of 0.5 represents a model performing no better than random chance. The closer the AUC-ROC value is to 1, the better the model is at distinguishing between positive and negative classes.

- **AUC-PR:** The AUC-PR (Area Under the Precision-Recall Curve) is calculated by plotting the precision against the recall at various binary classification thresholds, and then integrating the resulting curve. A value of 1 implies that the model has perfect precision and recall. However, unlike AUC-ROC, a model with AUC-PR equal to the ratio of positive samples is performing at random. Therefore, AUC-PR can provide a more informative picture of model performance when dealing with imbalanced classes.

- **IoU:** The IoU (Intesection over Union) is defined as the ratio of overlap between ground truth and predicted river channels to the total area covered by ground truth and predictions. It ranges from 0 to 1, with 1 being a perfect prediction.

In the context of tile classification, high precision is considered more valuable than high recall: due to the inherent connectivity of positive tiles for a river, post-processing interpolation can address and correct false negatives while false positives may add undesirable noise.

### 5.3. Model Training Setup

The binary classification CNNs were all trained on a Amazon Web Services (AWS) g5.2xlarge EC2 instance using Deep Learning AMI (Ubuntu 18.04) Version 71.0, which supports both PyTorch and TensorFlow [1]. The baseline simple CNN was trained using the TensorFlow Keras library while the pre-trained ResNets were trained using PyTorch [9]. The KNN baseline testing was done using Scikit-learn's `KNeighborsClassifier` function and the HOGs were calculated with `skimage.feature.hog`. The binary segmentation models were trained on Google Colab using the Segmentation Models library, which allows popular segmentation models and encoders to be imported and trained with PyTorch Lightning.

When training the ResNets, checkpoints of the model states were saved whenever a new best (lowest) validation loss or best (highest) validation AUC-PR were achieved. In the case that the model starts overfitting (i.e. when the model significantly underperforms on the validation set compared to the training set), these model states can be restored and evaluated.

## 6. Results and Discussion

### 6.1. Binary Classification (Tilewise Dataset)

The results of the model runs using the tilewise splits are shown in Table 1. The primary evident trend is that, for a model to be useful for recognizing a river channel, it must be sufficiently complex. The simplest-case models (KNNs and Simple CNN) tend to have relatively good precision but extremely poor recall, indicating that their relatively high accuracy is dominantly derived from the model underpredicting positive identifications (the presence of a river channel). The most effective model, following most metrics, is the ResNet-50 trained to optimize the area under the precision-recall curve. Indeed, when this model was used to predict the location of river channels, the results matched well with the manually labeled images (Fig. 4).

Whereas there are slight differences in the identification, the overall channel network is still readily observable. Because some of the imagery of this watershed was included in the training dataset, the model was trained on 60% of the images that it then predicted the class of. However, 40% of the images were in the validation and test sets, such that if the model were merely overfit to "memorize" the training set, 40% of the tiles would be poorly labeled.

When trained on the "tile-wise" training dataset, the model performance metrics were significantly improved (Table 2). This is qualitatively supported through agreement between ground truth and segmentation results in a random test dataset of megatiles (Fig. 3). This is likely due to the construction of the training dataset: since imagery from each basin included in the training set fills a unique set of possible conditions that we want to predict on (i.e., only imagery from one basin containing a dry sub-humid multi-thread channel with water in it), the model performs much

| Model | Accuracy | Precision | Recall | F1-Score | AUC-ROC | AUC-PR |
|---|---|---|---|---|---|---|
| Human Baseline | 0.850 | 0.780 | 0.920 | 0.840 | N/A | N/A |
| Input Independent | 0.865 | N/A | 0 | 0 | N/A | N/A |
| KNN (k = 3) | 0.845 | 0.268 | 0.087 | 0.131 | 0.525 | 0.239 |
| KNN (k = 5) | 0.855 | 0.308 | 0.062 | 0.104 | 0.52 | 0.248 |
| KNN HOG (k = 3) | 0.867 | 0.704 | 0.024 | 0.046 | 0.511 | 0.429 |
| KNN HOG (k = 5) | 0.867 | 0.857 | 0.015 | 0.029 | 0.507 | 0.502 |
| Simple CNN | 0.89 | 0.76 | 0.32 | 0.45 | N/A | N/A |
| ResNet-18 (Loss) | 0.885 | 0.545 | 0.894 | 0.678 | 0.889 | 0.502 |
| ResNet-50 (Loss) | 0.852 | 0.475 | 0.911 | 0.624 | 0.877 | 0.445 |
| ResNet-50 (AUC-PR) | 0.932 | 0.740 | 0.767 | 0.753 | 0.863 | 0.599 |

Table 1. Test set results for binary classification models trained on tilewise dataset. ResNet-18 (Loss) and ResNet-50 (Loss) represent the model states with the lowest validation loss, which trained for 11 epochs and 16 epochs respectively. ResNet-50 (AUC-PR) achieved the highest validation AUC-PR after training for 88 epochs.

| Model | Accuracy | IoU | F1-Score | Loss | Precision | Recall |
|---|---|---|---|---|---|---|
| FPN (ResNet-18) | 0.914 | 0.523 | 0.687 | 0.471 | 0.711 | 0.664 |
| FPN (ResNet-50) | 0.917 | 0.535 | 0.697 | 0.471 | 0.727 | 0.669 |
| U-Net (ResNet-18) | 0.919 | 0.516 | 0.681 | 0.507 | 0.774 | 0.608 |
| U-Net (ResNet-50) | 0.918 | 0.531 | 0.693 | 0.499 | 0.737 | 0.655 |

Table 2. Test set results for binary segmentation models trained on tilewise dataset. Model column indicates architecture and encoder: FPN (ResNet-18) uses Feature Pyramid Network architecture with the ResNet-18 encoder.

| Model | Model Type | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| ResNet-18 (PRC) | Classification | 0.852 | 0.314 | 0.879 | 0.463 |
| ResNet-18 (Loss) | Classification | 0.767 | 0.228 | 0.920 | 0.365 |
| ResNet-50 (PRC) | Classification | 0.856 | 0.311 | 0.809 | 0.449 |
| RestNet-50 (Loss) | Classification | 0.789 | 0.234 | 0.837 | 0.365 |
| FPN (ResNet-18) | Segmentation | 0.890 | 0.093 | 0.093 | 0.093 |
| FPN (ResNet-50) | Segmentation | 0.864 | 0.093 | 0.141 | 0.112 |
| U-Net (ResNet-18) | Segmentation | 0.885 | 0.088 | 0.096 | 0.092 |
| U-Net (ResNet-50) | Segmentation | 0.875 | 0.096 | 0.126 | 0.109 |

Table 3. Test set results comparing binary classification and binary segmentation models trained on basinwise dataset.

better if it was trained on some images from each aridity index zone and river channel type.

### 6.2. Binary Segmentation (Tilewise Dataset)

The results of the binary segmentation method using the tilewisse splits are shown in Table 2. The different models do not appear to significantly vary in terms of performance. This is true across all metrics. The deeper models appear to have slightly better performance, however, with an increased IoU and and F1-score. While a rigorous, direct comparison cannot be made with the binary classification due to the data in the respective tilewise splits being different, it appears that the best segmentation model notably have a lower recall and F1-score compared to the best binary classification model.

### 6.3. Binary Classification and Binary Segmentation (Basinwise Dataset)

The basinwise imagery split for the image segmentation model leads us to only have one basin in the test dataset. However, the model never trained on any imagery from the test dataset, such that it demonstrates how we expect the model would perform on a completely new set of imagery. The results of this method are shown in Table 2. Initially, the performance metrics appear to state that this method is far inferior to the binary classification method for

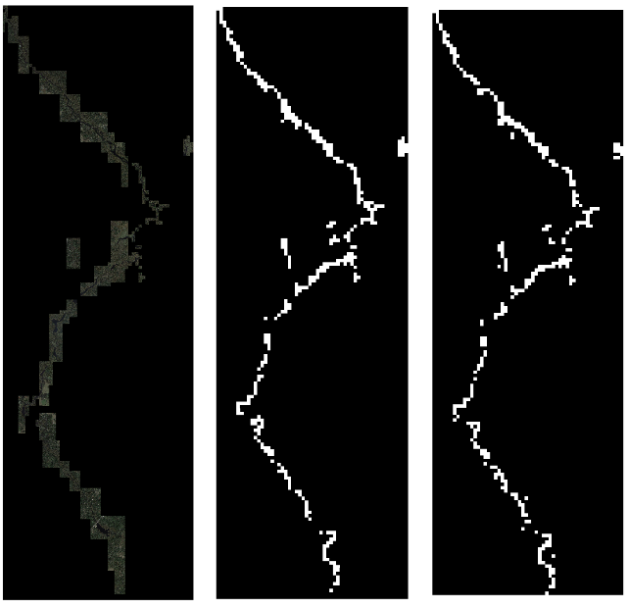

Figure 4. From left to right, the figure shows the original imagery, manually labeled ground truth data, and predictions made by the ResNet-50 model trained on the tilewise dataset to optimize the precision-recall curve. Note the slight differences between the ground truth and predictions, but general agreement. Some of the imagery in this basin was included in the training dataset; however, had the model overfit and "memorized" the training dataset, we would not expect better than 60% agreement.

river channel prediction. However, the model is at an inherent disadvantage: the tile-wise binary classification data were used to create the masks that the segmentation model was trained on. As such, the training data mask is effectively very highly pixelated. The segmentation model, however, has mapped channels as curvilinear (Fig. 5) such that pixel-wise accuracy metrics may imply that the segmentation model is performing extremely poorly despite relatively good qualitative agreement with ground truth labels. When trained on the "tile-wise" training dataset, the model performance metrics were significantly improved (Table 2 and Table 3). This is qualitatively supported through agreement between ground truth and segmentation results in a random test dataset of megatiles. This is likely due to the construction of the training dataset: since imagery from each basin included in the training set fills a unique set of possible conditions that we want to predict on (i.e., only imagery from one basin containing a dry sub-humid multithread channel with water in it), the model performs much better if it was trained on some images from each aridity index zone and river channel type

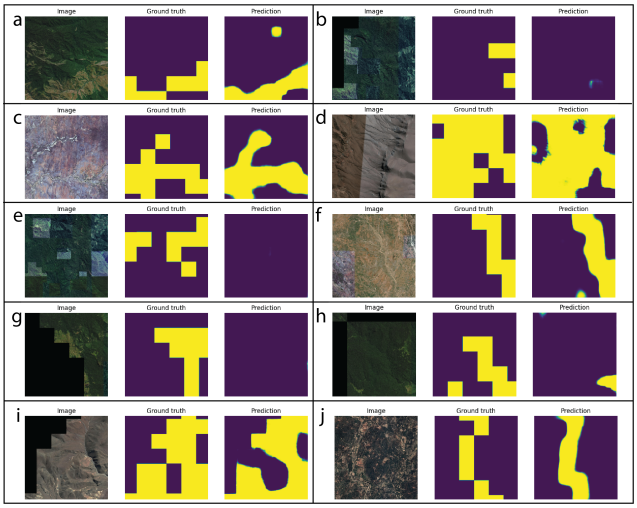


Figure 5. Examples of segmentations performed on megatiles (8x8 composites of manually labeled image tiles). Yellow indicates predicted channel location; blue indicates that there is no channel present. There is generally good agreement between ground truth and predictions, although in some instances there is strong disagreement (e.g., panels (e) and (g)).

### 6.4. Discussion

Whereas the maps created using the techniques discussed in this paper are considerably lower fidelity than handmade maps would be, we have demonstrated that deep learning-based computer vision provides a novel mapping method that can be used to extract scientifically useful information (e.g., channel length and drainage network extents). For the scientific question at hand, the goal is not to segment individual reaches at high resolution; rather, the goal is to map the lengths of channel networks that were previously unmappable through other automatic techniques. The process of model development and comparison provided valuable insights into the types of model architectures capable of recognizing river channels from panchromatic imagery. Now, with these results as a starting point, we can tune the hyperparameters of the model to further refine how the model learns and ultimately makes predictions. Additional next steps will involve post-processing the data to remove noise from incorrect predictions and then using skeletonization algorithms to improve the connectivity of the generated maps.

Table 3 shows a comparison between the classification and segmentation models on the test set. We see generally better performance by the segmentation models. This is inline with the idea that they are able to have a greater spatial knowledge in their predictions due to looking at more than one tile at a time. Additionally, something that is not taken into account in the statistics but is reflected qualitatively is that the segmentation model is able to have finer grained

predictions due to predicting at the pixel level. The predictions from the segmentation model are thus often more curvy, more closely reflecting the true shapes of rivers.

## 7. Contributions and Acknowledgements

Brian Amaro contributed to the segmentation pipeline, report creation, and figure production. Vivek Vajipey contributed to the development and training of the binary classification and segmentation models and report creation. Michael Hasson contributed to the project ideation, data collection, creation of the baseline CNN, and report creation.

## References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 5

[2] Lawrence E Band. Topographic partition of watersheds with digital elevation models. *Water resources research*, 22(1):15–24, 1986. 1

[3] Simon Cantrell, Jon Christopherson, Cody Anderson, Gregory L Stensaas, Shankar N Ramaseri Chandra, Minsu Kim, and Seonkyung Park. System characterization report on the worldview-3 imager. Technical report, US Geological Survey, 2021. 2

[4] Neil S Davies and Martin R Gibling. Paleozoic vegetation and the siluro-devonian rise of fluvial lateral accretion sets. *Geology*, 38(1):51–54, 2010. 1

[5] ME Donselaar, MC Cuevas Gozalo, and S Moyano. Avulsion processes at the terminus of low-gradient semi-arid fluvial systems: lessons from the río colorado, altiplano endorheic basin, bolivia. *Sedimentary Geology*, 283:1–14, 2013. 1

[6] Furkan Isikdogan, Alan Bovik, and Paola Passalacqua. Rivamap: An automated river analysis and mapping engine. *Remote Sensing of Environment*, 202:88–97, 2017. 2

[7] Bernhard Lehner, Kristine Verdin, and Andy Jarvis. New global hydrography derived from spaceborne elevation data. *Eos, Transactions American Geophysical Union*, 89(10):93–94, 2008. 2

[8] Manohara MM Pai, Vaibhav Mehrotra, Shreyas Aiyar, Ujjwal Verma, and Radhika M Pai. Automatic segmentation of river and land in sar images: A deep learning approach. In *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 15–20. IEEE, 2019. 2

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[10] DP Roy, MA Wulder, and TR Loveland. Wce, rg allen, mc anderson, d. *Helder, JR Irons, DM Johnson, R. Kennedy, TA Scambos, CB Schaaf, JR Schott, Y. Sheng, EF Vermote, AS Belward, R. Bindschadler, WB Cohen, F. Gao, JD Hipple, P. Hostert, J. Huntington, CO Justice, A. Kilic, V. Kovalskyy, ZP Lee, L. Lymburner, JG Masek, J. McCorkel, Y. Shuai, R. Trezza, J. Vogelmann, RH Wynne, and Z. Zhu*, pages 154–172, 2014. 2

[11] Harish Sangireddy, Colin P Stark, Anna Kladzyk, and Paola Passalacqua. Geonet: An open source software for the automatic and objective extraction of channel heads, channel network, and channel morphology from high resolution topography data. *Environmental Modelling & Software*, 83:58–73, 2016. 1

[12] Eitan Shelef and George E Hilley. Impact of flow routing on catchment area calculations, slope estimates, and numerical simulations of landscape development. *Journal of Geophysical Research: Earth Surface*, 118(4):2105–2123, 2013. 2

[13] David G Tarboton. A new method for the determination of flow directions and upslope areas in grid digital elevation models. *Water resources research*, 33(2):309–319, 1997. 1

[14] Ujjwal Verma, Arjun Chauhan, Manohara Pai MM, and Radhika Pai. Deeprivwidth: Deep learning based semantic segmentation approach for river identification and width measurement in sar images of coastal karnataka. *Computers & Geosciences*, 154:104805, 2021. 2

[15] Liguo Weng, Yiming Xu, Min Xia, Yonghong Zhang, Jia Liu, and Yiqing Xu. Water areas segmentation from remote sensing images using a separable residual segnet network. *ISPRS international journal of geo-information*, 9(4):256, 2020. 2

[16] Hanqiu Xu. Modification of normalised difference water index (ndwi) to enhance open water features in remotely sensed imagery. *International journal of remote sensing*, 27(14):3025–3033, 2006. 2

[17] Robert J Zomer, Jianchu Xu, and Antonio Trabucco. Version 3 of the global aridity index and potential evapotranspiration database. *Scientific Data*, 9(1):409, 2022. 2