An Elevation-Guided Annotation Tool for Flood Extent Mapping on Earth Imagery (Demo Paper)

Saugat Adhikari*, Da Yan*, Mirza Tanzim Sami*, Jalal Khalil*, Lyuheng Yuan*, Bhadhan Roy Joy*, Zhe Jiang+, Arpan Man Sainju#

*University of Alabama at Birmingham, Birmingham, AL, United States
+University of Florida, Gainesville, FL, United States
#Middle Tennessee State University, Murfreesboro, TN, United States
{saugat,yanda,mtsami,jalalk,lyuan,joyroy}@uab.edu
zhe.jiang@ufl.edu
arpan.sainju@mtsu.edu

ABSTRACT

Accurate and timely mapping of flood extent plays a crucial role in disaster management such as damage assessment and relief activities. In recent years, high-resolution optical imagery becomes increasingly available with the wide deployment of satellites and drones. However, analyzing such imagery data to extract flood extent poses unique challenges due to noises such as obstacles (e.g., tree canopies, clouds). In this paper, we propose an elevation-guided annotation tool for flood extent mapping, which allows annotators to provide the flooded/dry labels for just a few pixels to cover a large area where the labels of most other pixels are automatically inferred. The physical rule we use here to guide the automatic label inference is that if a location is flooded (resp. dry), then its adjacent locations with a lower (resp. higher) elevation must also be flooded (resp. dry). In this way, annotators just need to label the pixels that they are confident with, and the true labels of many ambiguous pixels such as tree-canopy ones can be automatically inferred. We demonstrate the usage of our annotation tool using high-resolution aerial imagery from National Oceanic and Atmospheric Administration (NOAA) National Geodetic Survey (NGS) together with the corresponding Digital Elevation Model (DEM) data. The annotated data can be used to train machine learning models for flood extent mapping, and we train U-Net models to infer the flood map for an unseen region and achieve a high accuracy. Our annotation tool is open-sourced at https://github.com/SaugatAdhikari/Flood-Annotation-Tool.

CCS CONCEPTS

• **Information systems** → **Geographic information systems**; **Data mining**; • **Computing methodologies** → **Machine learning**; • **Applied computing** → **Earth and atmospheric sciences**.

KEYWORDS

flood mapping, Earth imagery, annotation, U-Net, DEM

SIGSPATIAL '22, November 1–4, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9529-8/22/11.
https://doi.org/10.1145/3557915.3560962

ACM Reference Format:
Saugat Adhikari*, Da Yan*, Mirza Tanzim Sami*, Jalal Khalil*, Lyuheng Yuan*, Bhadhan Roy Joy*,, Zhe Jiang+, Arpan Man Sainju#. 2022. An Elevation-Guided Annotation Tool for Flood Extent Mapping on Earth Imagery (Demo Paper). In *The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3557915.3560962

1 INTRODUCTION

Flood extent mapping plays a crucial role in disaster management. For example, during Hurricane Harvey floods in 2017, first responders needed to know where flood water was to plan rescue efforts to residents in vulnerable communities, to understand the extent of damage to critical infrastructures (e.g. chemical plants and oil refineries), and to evaluate the impact on road networks and transportation. Another application is national water forecasting [4].

Given earth imagery with spectral features (e.g., RGB values) on a terrain surface (defined by an elevation map over a 2D grid), our goal is to conduct large-scale image segmentation so that each pixel (i.e., location) is classified as either "flooded" or "dry". The challenge, however, is with data annotation, which could be very laborious since a large number of pixels need to be manually labeled for model training. Moreover, pixels covered by clouds and tree canopies are ambiguous but the cost of their wrong annotation could be very high, especially when used to train physics-guided graphical models for flood extent mapping, such as Hidden Markov tree [4, 8] where a high-elevation pixel that is mislabeled as "flooded" could cause many surrounding dry pixels to be inferred as flooded.

In this demo paper, we propose a novel elevation-guided annotation tool which allows annotators to provide the flooded/dry labels for just a few pixels to cover a large area where the labels of most other pixels are automatically inferred. By iteratively labeling pixels that are not yet covered, an annotator can quickly assign labels to cover the majority pixels in the entire imagery. Since the automatic label inference is based on the physical constraint that *if a location is flooded (resp. dry), then its adjacent location with a lower (resp. higher) elevation must also be flooded (resp. dry)*, our annotation approach avoids wrong labels that may corrupt later inference steps. This approach also addresses the difficulty of annotating ambiguous pixels. For example, a tree-canopy pixel with an elevation lower (resp. higher) than its adjacent pixel that is clearly flooded (resp. dry) must also be flooded (resp. dry).

We remark that this work focuses on demonstrating our semi-automated elevation-guided annotation approach and on how it

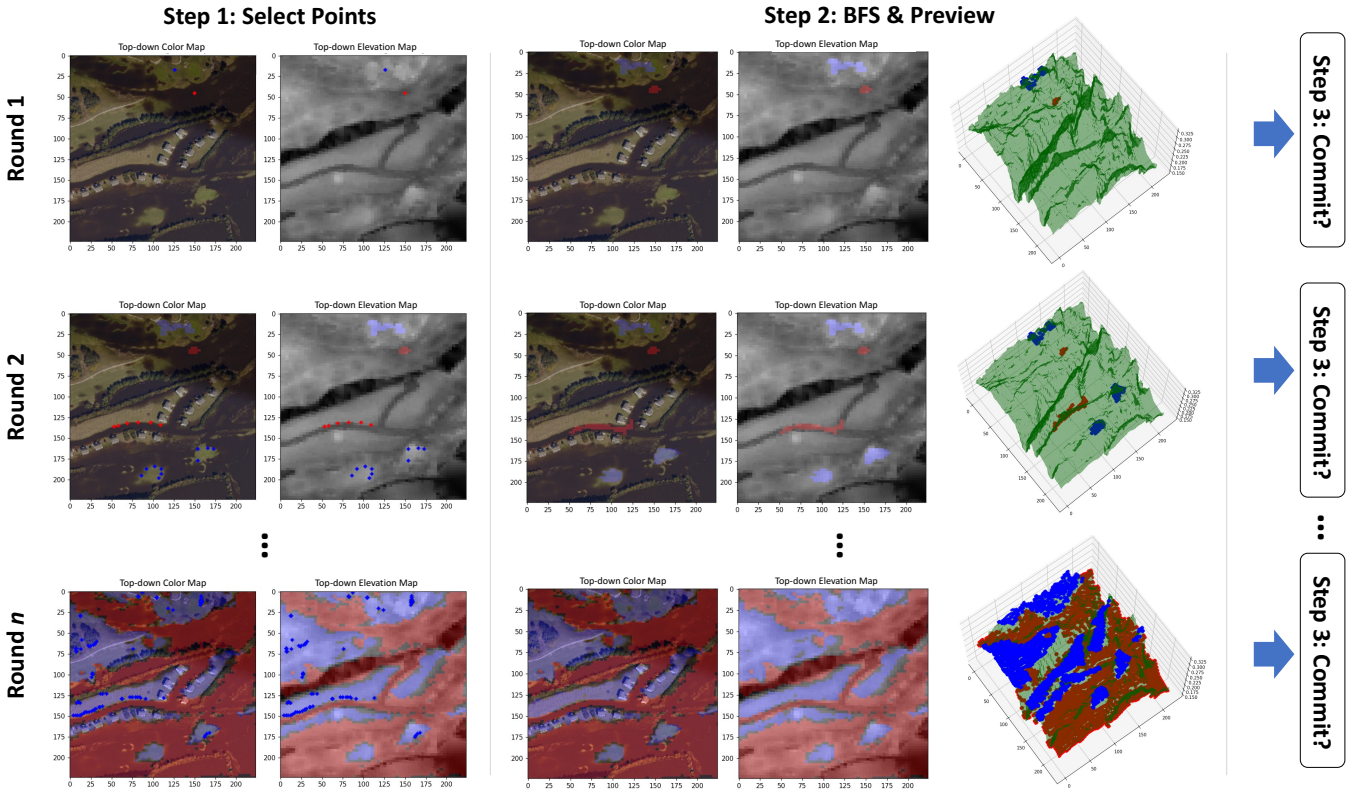

Figure 1: Illustration of the Annotation Process

facilitates the accurate annotation of ground-truth flood maps. The goal is to obtain error-free ground-truth flood maps to help us develop new physics-guided machine learning models, which would otherwise be misled by annotation errors/inaccuracies (e.g., on the boundary of the conventional polygon annotations). While this work uses the generic segmentation model U-Net [5] to illustrate our annotating-training-inference machine learning pipeline, our ultimate goal is to facilitate the development of physics-guided machine learning models that are even more sensitive to annotation errors/inaccuracies than U-Net, but once trained properly, are much more accurate than U-Net for flood extent mapping.

The rest of this demo paper is organized as follows. Section 2 describes our annotation tool, followed by Section 3 which reports the performance of U-Net models trained with our annotated data to infer the flood map for an unseen region. Finally, Section 4 describes our demonstration plan, and Section 5 discusses the room to improve our annotation tool and prediction models.

2 ELEVATION-GUIDED ANNOTATION

A common machine-learning approach for flood extent mapping is to divide a large high-resolution image into smaller patches, each of which fits as the input to a convolutional neural network (CNN) such as U-Net [5], which is learned with the elevation (or, depth) as an additional feature channel [2, 7].

For annotation and U-Net training purpose, we partition large high-resolution imagery into smaller patches of size 224×224 each. The patches can then be distributed to crowdworkers for annotation by crowdsourcing, for which we develop a convenient web-based annotation tool as illustrated in Figure 1, where annotators mark some flooded (resp. dry) pixels with red (resp. blue) points in each round, until most of the pixels have been labeled.

In Figure 1 Round 1, we first annotate a flooded (resp. dry) location p with a red (resp. blue) point in Step 1; let us denote its elevation by $h(p)$. Step 2 then infers surrounding pixels with elevations $\leq h(p)$ (resp. $> h(p)$) also as flooded (resp. dry) by running a breadth-first search (BFS) from p which stops where elevations begin to surpass $h(p)$ (resp. drop below $h(p)$). This gives the region covered in red (resp. blue), which is also visualized in a 3D plot that can be rotated from different angles to get a better intuition of the annotation result. After the result preview in Step 2, an annotator may commit the annotation in Step 3 if he/she is satisfied with the result. Round 2 illustrates the same process except that users may click multiple points in Step 1 at each time for higher efficiency, so that Step 2 will conduct BFS from each of the clicked points to infer more labels. Note that this is what annotators usually do in practice, since it is quick to click those obviously dry/flooded pixels that have not been labeled (i.e., covered by the red or blue mask) yet, so that Step 2 can infer labels for a large number of pixels to minimize the number of rounds required to cover most part of the input image with labels, as illustrated in the last round in Figure 1.

Note that it is not necessary to cover every pixel since U-Net uses a pixel-wise cross-entropy loss function, for which we can

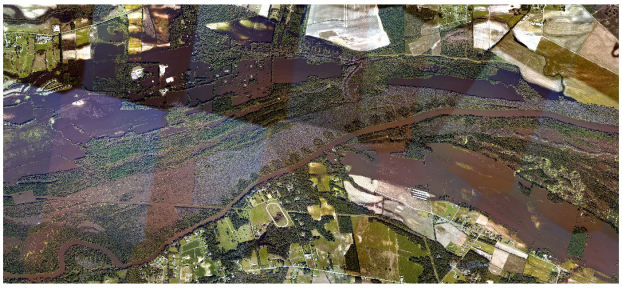

Figure 2: Training Region in Flooding Time (NOAA)

Figure 3: Training Region in Normal Time (Google Earth)

simply compute with those pixels that have labels. We do require annotators to put more points on the boundary between flooded and dry regions, since they are the most informative in model learning.

To facilitate the labeling of large continuous regions of flooded (resp. dry) pixels, we also allow annotators to use polygon annotations so that all pixels contained by a polygon are considered as being labeled as flooded (resp. dry) in Step 1. Moreover, we actually show 4 images in Step 1: (1) the original image, (2) the original elevation map, (3) the masked image, and (4) the masked elevation map (even though Figure 1 only plots the latter two), since sometimes it is easier to judge whether a location is flooded or dry from the original image, while the masked image can help identify which pixels are still not labeled yet so that an annotator only needs to click among those locations. Our interactive tool allows an annotator to move the mouse on any of the four images, and the current mouse location will be synchronously shown on all four images so that one may see if the location is flooded/dry on Image (1) and if it is not labeled yet on Image (3) simultaneously for better judgement. Previously selected points in Step 1 of the current round will also be displayed on all the four images, so that one may choose to click pixels in regions that have less point selections.

3 ELEVATION-AWARE FLOOD MAPPING

Once the training data are labeled, we then use them to train a U-Net [5] model to provide initial flood predictions. Besides the disaster-time image (see Figure 2), we also obtain its corresponding normal-time image from Google Earth (see Figure 3). We align the two images using Georeferencer in QGIS, and rotate them so that they are axis-aligned for ease of patch cutting. Note that the normal-time image is effective in helping identify flooded regions even visually, such as the lower right part of Figure 2.

The input to U-Net is a patch of size 224×224 with RGB image channels plus the corresponding elevation channel from DEM data. We consider two kinds of input tensors to U-Net: (1) a $224 \times 224 \times 4$ tensor where each pixel has RGB values in the flooding time plus an elevation value, and (2) a $224 \times 224 \times 7$ tensor where each pixel has RGB values in the flooding time, RGB values in the normal time, plus an elevation value.

Our experiments show that directly using absolute elevation as a pixel feature delivers a poor prediction performance, since the average elevation of the training region (e.g., upstream of a river) could be much larger than that of the target region where we want to predict flood map (e.g., downstream of a river), leading to many dry locations in the downstream of a river being predicted as flooded. Therefore, given a region R, we always first obtain its highest (resp. lowest) elevation $\max(R)$ (resp. $\min(R)$), and conduct min-max normalization that maps an elevation value h to $h' = \frac{h-\min(R)}{\max(R)-\min(R)}$; h' is then used for the elevation channel. In this way, systematic bias in elevation between the training region and the test region is resolved.

Our U-Net model takes the input tensor of shape $224 \times 224 \times 7$ (or $224 \times 224 \times 4$ if normal-time RGB values are not used). The encoder has 6 convolutional blocks that use 3×3 kernels. Each block has 2 convolution layers, a batch normalization layer, a ReLU activation layer and a 2×2 max-pooling layer. The number of convolution filters in the blocks are 32, 64, 128, 256, 512, and 1024, respectively. The bottleneck tensor has shape $7 \times 7 \times 1024$ which is then decoded back using a symmetric sequence of upsampling blocks. The U-Net outputs the flood probability map: for each pixel of the input image, the corresponding location in the output map gives the probability that this pixel is flooded. The final flood map can be extracted by binarizing the flood probability map using a probability threshold.

We find that U-Net predictions on the patch borders are not of a high quality, likely due to zero padding of convolutions. So the test region is cut into patches of size 192×192, but each patch is expanded by 16 pixels along every border to create a 224×224 patch to input into U-Net. The prediction then cuts the borders to obtain the predicted patch of size 192×192, to be assembled together into the flood probability map of the entire test region.

4 DEMONSTRATION PLAN

For demonstration purpose, we will use high-resolution aerial imagery from National Oceanic and Atmospheric Administration (NOAA) National Geodetic Survey[1] during Hurricane Mathew in North Carolina (NC) in 2016. The digital elevation map (DEM) data are from the University of North Carolina Libraries[2]. We resampled all the images into a resolution of 2 m $\times$ 2 m. The training region is Greenville, NC, and the test region is Grifton, NC.

We have provided our web-based annotation tool at https://flood-annotation.herokuapp.com/ (please leave the password empty to log in) to let the audience (1) try the annotating of aerial-image patches from the training and test regions, (2) obtain the flood probability map predicted by U-Net for aerial-image patches from the test region, (3) visualize the entire assembled flood probability map of the test region, (4) binarize the flood probability map using a slide bar of probability threshold to visualize the extracted flood extent as well as to see the reported performance metrics for different threshold values.

[1] https://geodesy.noaa.gov/storm_archive/storms/matthew/
[2] https: //www.lib.ncsu.edu/gis/elevation

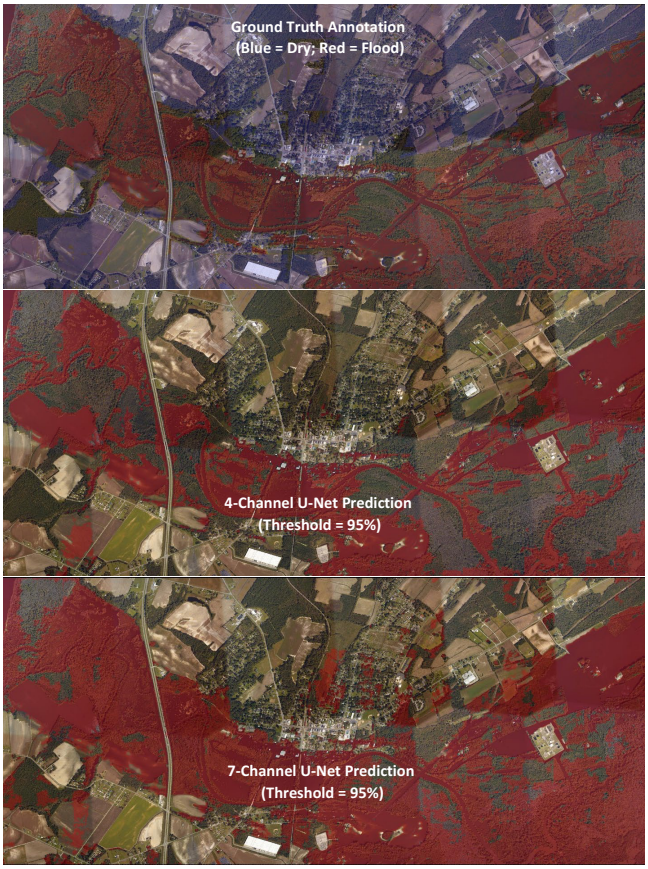

Figure 4: Ground-Truth Annotations and Predictions

Table 1: Performance Comparison of Different U-Net Models

Model	Accuracy	Precision	Recall	F-score
4-Channel (Flood = Positive)	86%	95%	68%	80%
7-Channel (Flood = Positive)	92%	86%	92%	89%
4-Channel (Dry = Positive)	86%	83%	98%	90%
7-Channel (Dry = Positive)	92%	95%	91%	93%

Figure 4 shows (1) the ground-truth annotations of the test region obtained using our annotation tool (note that most pixels are labeled), (2) U-Net predictions binarized by probability threshold 95% (tuned to give the highest pixel-level accuracy), where we use a 4-channel U-Net with the RGB values of only the flooding time imagery, as well as a 7-channel U-Net that also uses the RGB values of the normal-time imagery from Google Earth. Visually, the 7-channel U-Net predicts more dry pixels to be flooded, but it better covers the ground-truth flooding areas than the 4-channel U-Net.

Table 1 reports the performance metrics of both models, when we consider "flooded" as the positive class, as well as when we consider "dry" as the positive class. We can see that the 7-channel U-Net gives a higher accuracy on the labeled test pixels, even though the precision of flood pixels is lower (86%) than that of the 4-channel U-Net (95%) which is because the 7-channel U-Net predicts many dry pixels as flooded; in contrast, the 4-channel U-Net has a much lower recall of flood pixels (68%) than that of the 7-channel U-Net (92%), leading to a lower F-score. These numbers show that the overall winner is the 7-channel U-Net, i.e., normal-time image helps.

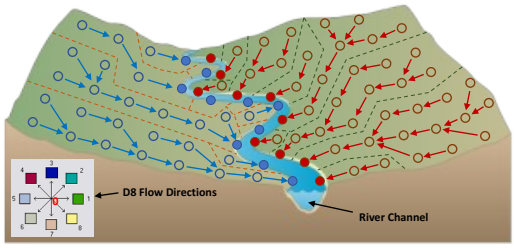

Figure 5: Flow-Direction Forest

5 DISCUSSION

Note that some dry regions are predicted as flooded by U-Nets in Figure 4. Since there is no indication that convolution kernel is effective in capturing and utilizing the complex topological structures of a terrain surface [3], physics-guided models such as HMT [8] could be a better model choice, either to be used on its own or to refine U-Net model outputs. Different from U-Net that operates on individual patches, the HMT approach regards the entire terrain region as trees for graphical model inference (see Figure 5 for an example where flow-direction trees are constructed with roots in the river channel, which can be constructed using "D8 Flow Directions" [1] that is supported by TauDEM [6]), so building our annotation tool to take an entire terrain region rather than individual patches would be beneficial, which also allows annotators to see a bigger picture to give more accurate annotations. However, displaying and rotating the entire 3D terrain for annotation is computationally expensive and parallel GPU solutions would be essential (e.g., using WebGPU technology), which will be our future work.

ACKNOWLEDGMENTS

This work was supported by ARDEF 1ARDEF21 03 from ADECA, NSF OAC-2106461, OAC-2152085, IIS-2147908 and IIS-2207072.

REFERENCES

[1] Jurgen Garbrecht and Lawrence W Martz. 1997. The assignment of drainage direction over flat surfaces in raster digital elevation models. *Journal of hydrology* 193, 1-4 (1997), 204–213.
[2] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. 2016. FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture. In *ACCV (Lecture Notes in Computer Science)*, Vol. 10111. Springer, 213–228.
[3] Wenchong He, Arpan Man Sainju, Zhe Jiang, Da Yan, and Yang Zhou. 2022. Earth Imagery Segmentation on Terrain Surface with Limited Training Labels: A Semi-supervised Approach based on Physics-Guided Graph Co-Training. *ACM Trans. Intell. Syst. Technol.* 13, 2 (2022), 26:1–26:22.
[4] Zhe Jiang and Arpan Man Sainju. 2021. A hidden Markov tree model for flood extent mapping in heavily vegetated areas based on high resolution aerial imagery and DEM: A case study on hurricane matthew floods. *International Journal of Remote Sensing* 42, 3 (2021), 1160–1179.
[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI (Lecture Notes in Computer Science)*, Vol. 9351. Springer, 234–241.
[6] David G Tarboton. 2005. Terrain analysis using digital elevation models (TauDEM). *Utah State University, Logan* 3012 (2005), 2018.
[7] Jinghua Wang, Zhenhua Wang, Dacheng Tao, Simon See, and Gang Wang. 2016. Learning Common and Specific Features for RGB-D Semantic Segmentation with Deconvolutional Networks. In *ECCV (Lecture Notes in Computer Science)*, Vol. 9909. Springer, 664–679.
[8] Miao Xie, Zhe Jiang, and Arpan Man Sainju. 2018. Geographical Hidden Markov Tree for Flood Extent Mapping. In *KDD*. ACM, 2545–2554.