End-term Report: River Network Detection Project

Srinivasan M (IMT2021058)   Siddharth Kothari (IMT2021019)
Sankalp Kothari (IMT2021028)

August 8, 2024

1 Accomplishments before the Midterm

1.1 Data Acquisition

- Selected appropriate Sentinel-1 data for the Amazon River Basin. Accessed relevant Sentinel-1 scenes from platforms like the Sentinels Scientific Data Hub and Copernicus web browser.
- Written a code which when given the wkt coordinates of the location, and the range of dates, it can retrieve Sentinel-1 GRD products from the Alaska Satellite Facility's website.
- The code can be used to download sentinel-1 GRD products for any location and range of dates at will, and can be used to get images for the dataset. The products downloaded can be processed using Python codes or the ESA Snap tool.

1.2 Data Pre-Processing

- Application of Orbit File to the Sentinel-1 GRD products. Conducted sensor calibration to correct for instrumental limitations.
- Reduced speckle noise through appropriate filtering techniques. Applied geometric corrections including terrain correction and geometric resampling.
- Wrote a code to get various subsets from a single product, which can be used for data augmentation, along with a code to automate the previous processes, which can also be done using the ESA Snap Tool.
- Also completed setup of the environment required to run the code.

1.3 Preliminary models and analysis

- Started implementing algorithms for river course delineation and change detection. The first one that we implemented was K-Means Clustering, which gives acceptable results.
- Began looking into existing benchmark datasets such as S1-S2 Water Dataset, and Sen1Floods11 Datasets.

1.4 Tools and Libraries

- Employed Python libraries for image processing (OpenCV, NumPy, SciPy, GeoPandas, Matplotlib), and automated the process of applying the aforementioned preprocessing techniques in Python.
- Utilized SNAP software for pre-processing (downloadable from ESA).

1.5 Dataset Mask Generation

1. Started work on generation of water masks for various sentinel 1 images, and also began to look into existing datasets with water masks.
2. Currently, the water masks are generated by using shapefiles on the net, which are then clipped to suit our purpose, from which we take the intersection.
3. We are looking into better shapefiles which can capture the width of the river as well, which is not captured by shapefiles currently. Only the direction of flow is captured, which renders inefficient masks.

1.6 Readings

We also looked into few of the research papers, to get to know more about the current research and progress that is happening in this domain. Most of the papers that we looked into consisted of using SAR images and doing further segmentation analysis on water bodies. Few of them also included on specialzed methods to do computation and Image processing (eg. New and improved methods of doing convolution in Neural Networks). Few of the notable research papers that we had gone through are given below:

1. Dynamic Snake Convolution [10]
2. Deep learning based semantic segmentation approach for river identification and width measurement in SAR images of Coastal Karnataka [11]
3. Water-Body Segmentation for SAR Images: Past, Current, and Future [12]
4. A Review on Multiscale-Deep-Learning Applications [13]

2 Accomplishments post the Midterm

2.1 Data Pre-Processing

- Modified the subsetting code to be able to do two kinds of subsetting - Based on longitude and latitude and based on the image size, to generate multiple images from the same image, with the same dimensions, capturing different parts of the rivers.
- This completely eliminates the need to use the SNAP tool, and its usage is only limited to providing the necessary tools to work with the Snappy library in Python.

2.2 Shapefile Acquisition

- Looked into multiple shapefiles corresponding to rivers from multiple sources (such as HydroSHEDS [14], and MeritHYDRO [15]). We found that none of these datasets were able to effectively capture river widths. Hence we discarded them.
- Finally, we found the Diva-GIS [16] shapefile which gave acceptable results. However we observed a number of discrepancies as well, which we shall highlight in section 3.
- We used the shapefiles from Diva-GIS from 5 countries which captured the entirety of the Amazon region - Brazil, Guyana, Venezuela, Peru and Bolivia.

2.3 Tools and Libraries

- Employed Python libraries for image processing (OpenCV, NumPy, SciPy, GeoPandas, Matplotlib), and automated the process of applying the aforementioned preprocessing techniques in Python using snappy library.
- We used QGIS to be able to work effectively with the shapefiles and to carry out operations such as merging shapefiles, visualising geogrpahic regions etc.

2.4 Survey on Existing Datasets

We reviewed the Sen1Floods11 and S1-S2 Water Datasets, however they did not seem to be of use. Hence we did not look further into them. We still intend to revisit s1-s2 water dataset to review the images once again.

2.5 Dataset Mask Generation

1. This combines all the stuff we have done till now. We first downloaded 8-9 products using our downloading code from the ASF facility.
2. We then performed bulk preprocessing on each of the products. We then used the subsetting code to be able to generate images of size 512x512 size. Using this, we were able to generate around 1263 images.
3. We then ran the code to clip the shapefile and intersect with our tiff files to generate the ground truth. The shapefile used was the Diva-GIS shapefile.

2.6 Unet Model

The architecture for the Residual Unet is provided below, and the architecture can be inferred from the same. The Dense Unet has the same architecture, but with dense blocks instead of residual blocks.

```
def residual_unet(input_shape):
    inputs = Input(input_shape)

    # Encoder
    conv1 = residual_conv_block(inputs, 64)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)
    conv2 = residual_conv_block(pool1, 128)
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
    conv3 = residual_conv_block(pool2, 256)
    pool3 = MaxPooling2D(pool_size=(2, 2))(conv3)
    conv4 = residual_conv_block(pool3, 512)
    pool4 = MaxPooling2D(pool_size=(2, 2))(conv4)

    # Bottleneck
    conv5 = Conv2D(1024, 3, activation='relu', kernel_initializer='he_normal', padding='same')
                                               (pool4)
    conv5 = Conv2D(1024, (3, 3), kernel_initializer='he_normal', padding='same')(conv5)
    drop5 = Dropout(0.5)(conv5)

    # Decoder
    up6 = Conv2DTranspose(512, (2, 2), strides=(2, 2), padding='same')(drop5)
    up6 = Concatenate()([up6, conv4])
    conv6 = residual_conv_block(up6, 512)
    up7 = Conv2DTranspose(256, (2, 2), strides=(2, 2), padding='same')(conv6)
    up7 = Concatenate()([up7, conv3])
    conv7 = residual_conv_block(up7, 256)
    up8 = Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same')(conv7)
    up8 = Concatenate()([up8, conv2])
    conv8 = residual_conv_block(up8, 128)
    up9 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same')(conv8)
    up9 = Concatenate()([up9, conv1])
    conv9 = residual_conv_block(up9, 64)

    # Output
    outputs = Conv2D(1, 1, activation='sigmoid')(conv9)

    model = Model(inputs=inputs, outputs=outputs)
    return model
```

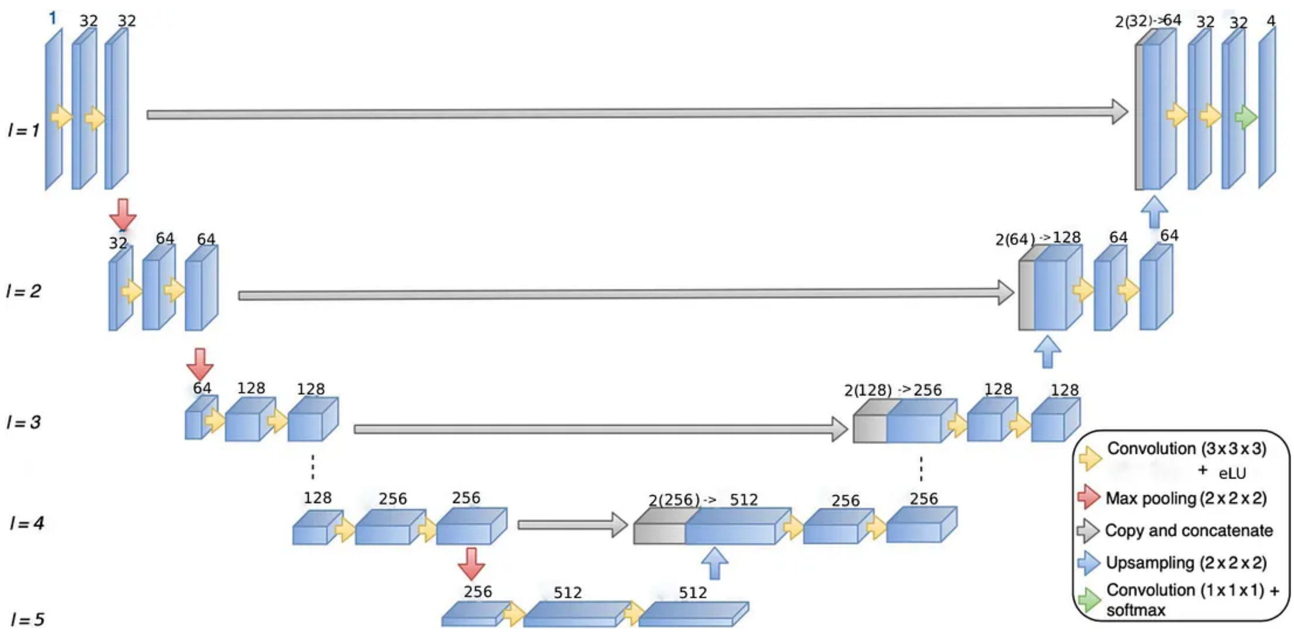

Figure 1: Unet Model

3 Results

We observed the following after training the models, and testing them to see their results. We have attached three images in support of our inferences (Figures 2, 3, 4).

1. At a higher resolution, we found that the Diva-GIS shapefile that we used for generating gorund truths is not that accurate, and a number of images are observed where the images are not related to the ground truths at all. This brings in the need to have better ground truths.

2. Despite the wrong ground truth, we observe that when solely compared to the actual image, and what we can "see" as the actual river structure, we can see that the UNet is able to learn the structure of the river properly.

3. Regardless, since the model uses corrupted ground truths, the accuracy is 76%, and we feel the model can do better if better ground truth is available.

4. We thus intend to frame the task as an adversarial learning problem, where the performance of the model with the corrupt ground truths is compared and contrasted with the performance of the model once retrained with the correct ground truths.

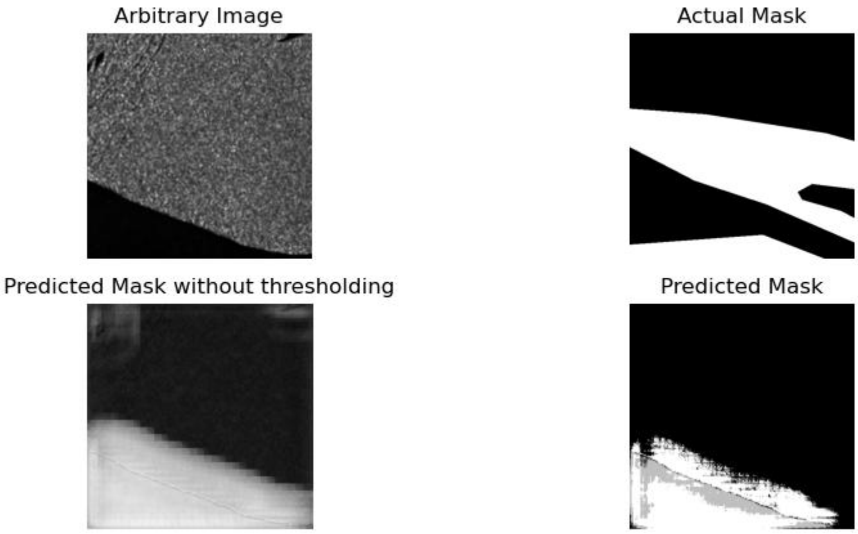

Figure 2: Output 1 of Residual UNet

4 Next Steps

1. We have a model trained on the corrupt ground truths, we now intend to first explore methods to generate the correct ground truths. Once that is done, we intend to retrain the model, and compare and contrast the performances of the models in both cases. Thus the project can be looked at from the lens of adversarial learning.

2. To generate correct ground truths, we intend to explore the use of Sentinel 2 images, and the NDWI (Normalized Difference Water Index), to be able to get correct ground truths. This approach has been taken previously by Wieland et al [5].

3. We are yet to look into data augmentation techniques, which can be used to get even more images from the current raster of images. This is something we intend to explore in the forthcoming months.

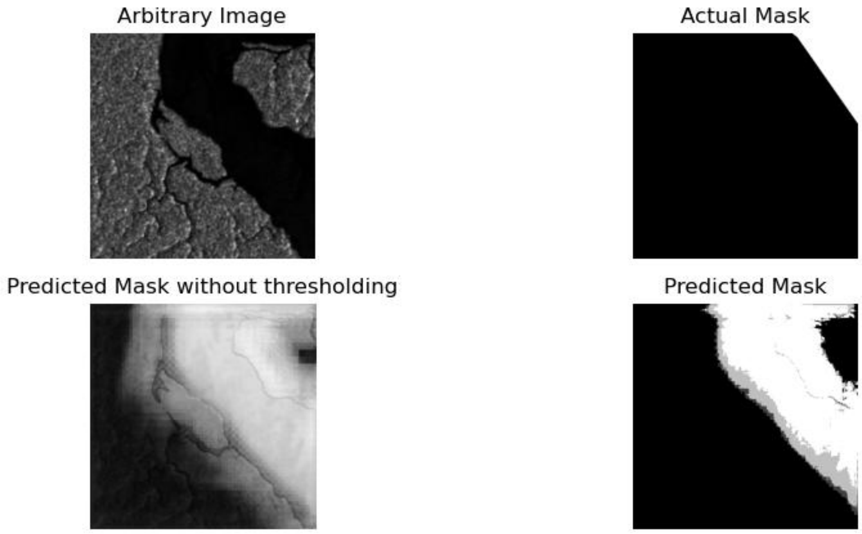

Figure 3: Output 2 of Residual UNet

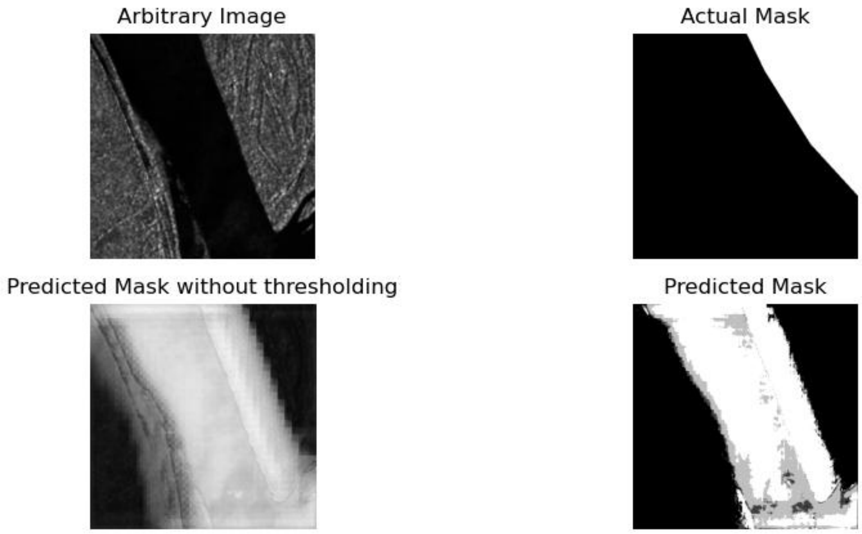

Figure 4: Output 3 of Residual UNet

5 Links to Code and Dataset

The codes for the project are hosted at the following URL - https://github.com/SSS-192858/River_Extraction_PE.git

The link to the dataset - https://iiitbac-my.sharepoint.com/:f:/r/personal/sankalp_kothari_iiitb_ac_in/Documents/Satellite%20project%20-%20Dataset/Final_Dataset?csf=1&web=1&e=p398Tt

References

[1] Obida, C. B., Blackburn, G. A., Whyatt, J. D., and Semple, K. T., "River network delineation from Sentinel-1 SAR data", *International Journal of Applied Earth Observation and Geoinformation*, vol. 83, 2019. doi:10.1016/j.jag.2019.101910.

[2] Code to bulk download data from Alaska Satellite Facility - https://www.linkedin.com/pulse/bulk-download-sentinel-1-data-aditya-sharma/

[3] Performing preprocessing on SAR images using Python library Snappy - https://step.esa.int/docs/tutorials/Performing%20SAR%20processing%20in%20Python%20using%20snappy.pdf

[4] Snappy installation guide - https://gitlab.com/JohnCrabs/installation_esa_snappy

[5] M. Wieland et al., "S1S2-Water: A Global Dataset for Semantic Segmentation of Water Bodies From Sentinel- 1 and Sentinel-2 Satellite Images," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 17, pp. 1084-1099, 2024, doi: 10.1109/JSTARS.2023.3333969.

[6] 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.

[7] Copernicus Browser Data Download Website - https://dataspace.copernicus.eu/browser

[8] Shapely Documentation - https://shapely.readthedocs.io/en/stable/manual.html

[9] KMeans Clustering implementation on Sentinel 1 and 2 - https://github.com/acgeospatial/Satellite_Imagery_Python/blob/master/Clustering_KMeans-Sentinel2.ipynb

[10] Y. Qi, Y. He, X. Qi, Y. Zhang and G. Yang, "Dynamic Snake Convolution based on Topological Geometric Constraints for Tubular Structure Segmentation," 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2023, pp. 6047-6056, doi: 10.1109/ICCV51070.2023.00558.

[11] Ujjwal Verma, Arjun Chauhan, Manohara Pai M.M., Radhika Pai, DeepRivWidth : Deep learning based semantic segmentation approach for river identification and width measurement in SAR images of Coastal Karnataka, Computers & Geosciences, Volume 154, 2021, 104805, ISSN 0098-3004, https://doi.org/10.1016/j.cageo.2021.104805.

[12] Guo, Z.; Wu, L.; Huang, Y.; Guo, Z.; Zhao, J.; Li, N. Water-Body Segmentation for SAR Images: Past, Current, and Future. Remote Sens. 2022, 14, 1752. https://doi.org/10.3390/rs14071752

[13] Wang, D.; Cao, W.; Zhang, F.; Li, Z.; Xu, S.; Wu, X. A Review of Deep Learning in Multiscale Agricultural Sensing. Remote Sens. 2022, 14, 559. https://doi.org/10.3390/rs14030559

[14] L. Warmedinger et al., "The New Hydrographic Hydrosheds Database Derived from the Tandem-X Dem," IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium, Pasadena, CA, USA, 2023, pp. 1485-1488, doi: 10.1109/IGARSS52108.2023.10282244.

[15] Yamazaki, Dai & Ikeshima, Daiki & Sosa, Jeison & Bates, Paul & Allen, George & Pavelsky, Tamlin. (2019). MERIT Hydro: A High-Resolution Global Hydrography Map Based on Latest Topography Dataset. Water Resources Research. 55. 10.1029/2019WR024873.

[16] Hijmans, Robert J., Luigi Guarino, A. Jarvis, R. O'brien, P. Mathur, C. Bussink, M. Cruz, I. Barrantes, and E. Rojas. "DIVA-GIS, version 5.2." (2005): 2.