

Ni Putu Karisma Dewi<sup>1</sup>, Putu Hendra Suputra<sup>2</sup>, A.A. Gede Yudhi Paramartha<sup>3</sup>,  
Luh Joni Erawati Dewi<sup>4</sup>, Pariwate Varnakovida<sup>5</sup>, Kadek Yota Ernanda Aryanto<sup>6</sup>

## River Area Segmentation Using Sentinel-1 SAR Imagery with Deep-Learning Approach


**Abstract:** River segmentation is important in delivering essential information for environmental analytics such as water management, flood/disaster management, observations of climate change, or human activities. Advances in remote-sensing technology have provided more complex features that limit the traditional approaches' effectiveness. This work uses deep-learning-based models to enhance river extractions from satellite imagery. With Resnet-50 as the backbone network, CNN U-Net and DeepLabv3+ were utilized to perform the river segmentation of the Sentinel-1 C-Band synthetic aperture radar (SAR) imagery. The SAR data was selected due to its capability to capture surface details regardless of weather conditions, with VV+VH band polarizations being employed to improve water surface reflectivity. A total of 1080 images were utilized to train and test the models. The models' performance was measured using the Dice coefficient. The CNN U-Net architecture achieved an accuracy of 0.94, while DeepLabv3+ attained an accuracy of 0.92. Although DeepLabv3+ showed more stability during the training and performed better on wider rivers, CNN U-Net excelled at identifying narrow rivers. In conclusion, a river-segmentation model was conducted using Sentinel-1 C-Band SAR data, with CNN U-Net outperforming DeepLabv3+; this enabled detailed river mapping for irrigation- and flood-monitoring applications – particularly in cloud-prone tropical regions.

**Keywords:** river, segmentation, satellite imagery, remote sensing, deep learning, CNN U-Net, DeepLabv3+


Received: December 5, 2024; accepted: June 16, 2025


© 2025 Author(s). This is an open-access publication, which can be used, distributed, and reproduced in any medium according to the Creative Commons CC-BY 4.0 License


<sup>1</sup> Ganesha University of Education, Faculty of Engineering and Vocational, Computer Science, Singaraja, Indonesia, email: karisma.dewi.3@undiksha.ac.id

<sup>2</sup> Ganesha University of Education, Faculty of Engineering and Vocational, Software Engineering Technology, Singaraja, Indonesia, email: hendra.suputra@undiksha.ac.id,  <https://orcid.org/0000-0003-1521-9307>

<sup>3</sup> Ganesha University of Education, Faculty of Engineering and Vocational, Computer Science, Singaraja, Indonesia, email: yudhi.paramartha@undiksha.ac.id,  <https://orcid.org/0000-0003-2640-3433>

<sup>4</sup> Ganesha University of Education, Faculty of Engineering and Vocational, Software Engineering Technology, Singaraja, Indonesia, email: joni.erawati@undiksha.ac.id,  <https://orcid.org/0000-0003-3041-731X>

<sup>5</sup> King Mongkut's University of Technology Thonburi, Faculty of Science, Department of Mathematics, Bangkok, Thailand, email: pariwate@gmail.com,  <https://orcid.org/0000-0003-3425-3447>

<sup>6</sup> Ganesha University of Education, Faculty of Engineering and Vocational, Computer Science, Singaraja, Indonesia, email: yota.ernanda@undiksha.ac.id (corresponding author),  <https://orcid.org/0000-0002-0368-890X>

## 1. Introduction

The accurate extraction of river information from remote-sensing images has always been a significant area of research for future planning, including irrigation, river monitoring, flood mapping, and comprehensive watershed management [1–4]. One important task in obtaining river information is extracting the river areas within the images. Traditional river-segmentation methods for remote-sensing images primarily include morphology, thresholding [5], edge detection [6], clustering [7], filtering [8] and band ratio as well as some machine-learning methods such as support vector machine [9], random forest [10], and maximum likelihood [11]. However, conducting a manual segmentation of river areas is time-consuming and labor-intensive – even on a small scale; this makes it increasingly impractical for large-scale global assessments.

Thresholding-based segmentation and surface features such as ponds, lakes, and seas can exhibit values that closely represent or even match a river, thus introducing significant noise and reducing the segmentation's accuracy in isolating the river [1]. Similarly, morphological operations like opening and closing can alter the river's geometry, leading to such inconsistencies as narrowing the river or distorting its actual shape [12]. With the continuous advancements in remote-sensing technology, the complexity of surface features in imagery has additionally increased, making non-river interference more prominent; this causes the more challenging extraction of river information due to features that are more complex and harder to distinguish than their surrounding noises.

Therefore, the advancement of satellite remote-sensing technology could make river mapping easier while significantly reducing one's time constraints. More-advanced methods such as deep learning are necessary to extract river areas more effectively and accurately. Deep-learning architectures are advanced machine-learning techniques that enhance computational performance and accuracy by increasing a network's number of layers or the overall depth [13]. In recent years, the use of deep learning in satellite remote-sensing data has been widely applied in various fields [14–16]; semantic segmentation [17–20] is one of the deep-learning methods that are widely used for processing remote-sensing images [21–24]. Semantic segmentation is computer vision that involves classifying each pixel of an image into a predefined category. The CNN U-Net architecture works by its skip connections; these link the encoder and decoder layers at their corresponding levels, which helps to retain high-resolution features and recover spatial information that may be lost during the down-sampling. The DeepLabv3+ architecture leverages atrous convolutions, which enable control over the receptive field without reducing the spatial resolutions of feature maps (thus, effectively capturing multiscale contextual information) [25, 26].

Furthermore, satellite imagery is vital in delivering consistent, comprehensive, and detailed data on the Earth's surface to allow for the monitoring, analysis, and

management of environmental conditions across vast areas. The SAR satellite is an active remote-sensing technology that utilizes microwave radar signals to capture images of the Earth. It emits radio waves toward the surface and records the echoes that return [27]. One of the primary advantages of SAR is its ability to obtain high-resolution images – even through cloud cover [28]. SAR sensors can also operate both during the daylight hours and at night regardless of weather conditions [9, 11, 29], making it well-suited for obtaining imagery for river segmentation.

This paper uses SAR imagery from Copernicus, which is Sentinel 1 Level-1 C-Band Ground Range Detected (GRD) Interferometric Wide Swath (IW) – a semantic segmentation approach for detecting river areas. The models that were used for this task were CNN U-Net and DeepLabv3+, which were trained on the data set to identify the river regions. The effectiveness of the models was assessed and compared based on their Dice coefficient performances.

## 2. Related Works

Research that is focused on segmentation using remote-sensing data (especially river segmentation) is relatively scarce for several reasons. One significant reason is the need for multiple processing steps to isolate only the river while excluding other water bodies such as lakes, ponds, and wetlands. This complexity increases the difficulty of accurately identifying and classifying river features.

Previous studies have used semi-automatic methods such as thresholding, morphology, and edge detection for river detection. For instance, Zhu et al. [5] utilized the Otsu thresholding algorithm combined with morphological features to extract river channels in SAR imagery. The method began with gray threshold-based image segmentation to remove any background noise, followed by a novel morphological model for river channel identification. After the rough extraction, the gray threshold segmentation was reapplied to refine the results, and a morphological filter was used for correction. Yang et al. [8] detected river networks in the Yukon Basin and the Greenland Ice Sheet by combining Gabor filtering and path opening in Landsat 8 remote-sensing imagery. The utilization of Gabor filtering was to make the rivers' shapes stand out from their surroundings, while path opening extended the lengths of the rivers and removes unwanted details. However, these semi-automatic methods had certain limitations, such as requiring multiple steps to achieve accurate river-area extraction. Additionally, these steps needed to be repeated each time when classifying new rivers, thus making the process time-consuming and less efficient.

The deep-learning approach is seen as an effective solution for overcoming the shortcomings of the semi-automatic methods, as it can be applied to river-data segmentation without the need for repetitive preprocessing each time a new river area needs to be segmented. Verma et al. [30] employed CNN U-Net and

DeepLabv3+ for semantic segmentation in coastal areas. This research had two primary objectives: developing a river-segmentation model, and measuring river width by determining the river skeleton. The data set that was used in the study was comprised of Sentinel-1 satellite imagery with VH-band polarization. The original  $2048 \times 3072$ -pixel images were tiled into  $256 \times 256$ -pixel segments and manually hand-labeled as rivers and non-rivers. The CNN U-Net and DeepLabv3+ architectures achieved the same mean Intersection over Union (mIoU) metric (96%); however, DeepLabv3+ outperformed CNN U-Net by 1% in the F1-score, achieving 98% as compared to CNN U-Net's 97%. The river data set for this study only covered rivers that had large surfaces; a river that was close to a populated area was not contained in the data set.

Pai et al. [2] conducted semantic river segmentation using satellite SAR imagery. The architectures that were employed were a CNN U-Net model that was built from scratch and a pre-trained CNN U-Net model with weights that were learned from the ISBI 2015 Cell Tracking data set. The study yielded highly accurate results, with the Vanilla CNN U-Net and Transfer CNN U-Net each achieving a precision of 0.99 and a mIoU of 0.95.

Cai et al. [31] utilized the Gaofen Image data set (GID) and the Remote-Sensing Image Block Segmentation data set (BDCI) in order to employ river segmentation using the CNN U-Net architecture enhanced with VGG16 and ResNet 34 to improve the detections of river edges with greater detail. The experimental results indicated that the mIoU of the ResNet34 CNN U-Net network on the GID-river data set reached 93.6%, while the mean Pixel Accuracy (mPA) of the VGG16 CNN U-Net network on the BDCI river data set reached 82.1%.

Chen et al. [32] developed the ASA-DRNet architecture – an improved version of the DeepLabv3+ framework. This architecture was designed to detect oil-spill pollution in the ocean more accurately using SAR imagery. The backbone network combined an axial self-attention module with ResNet-18 and an atrous spatial pyramid pooling (ASPP) optimized to enhance the network's capacity. ASA-DRNet achieved an mIoU accuracy of 0.6447, thus showing a significant improvement when compared to the CNN U-Net architecture (which had an accuracy of 0.5925).

Prior river-segmentation research (especially when using traditional methods) has often been hampered by time-consuming manual processes and misclassifications. While deep learning offers improved efficiency, existing studies still show varied performance and lack a clear consensus on optimal architectures for detailed river extraction. This work aims to enhance river extraction from Sentinel-1 C-Band SAR imagery using CNN U-Net and DeepLabv3+. Our innovation lies in a comprehensive comparative analysis of these models for detailed SAR-based river segmentation that specifically highlights their differential strengths for narrow versus wide rivers – a crucial contribution for applications in cloud-prone regions.

3. Materials and Methods

The research methodology is organized into four main stages: data collection, data preprocessing, model training, and performance evaluation (as illustrated in Figure 1).

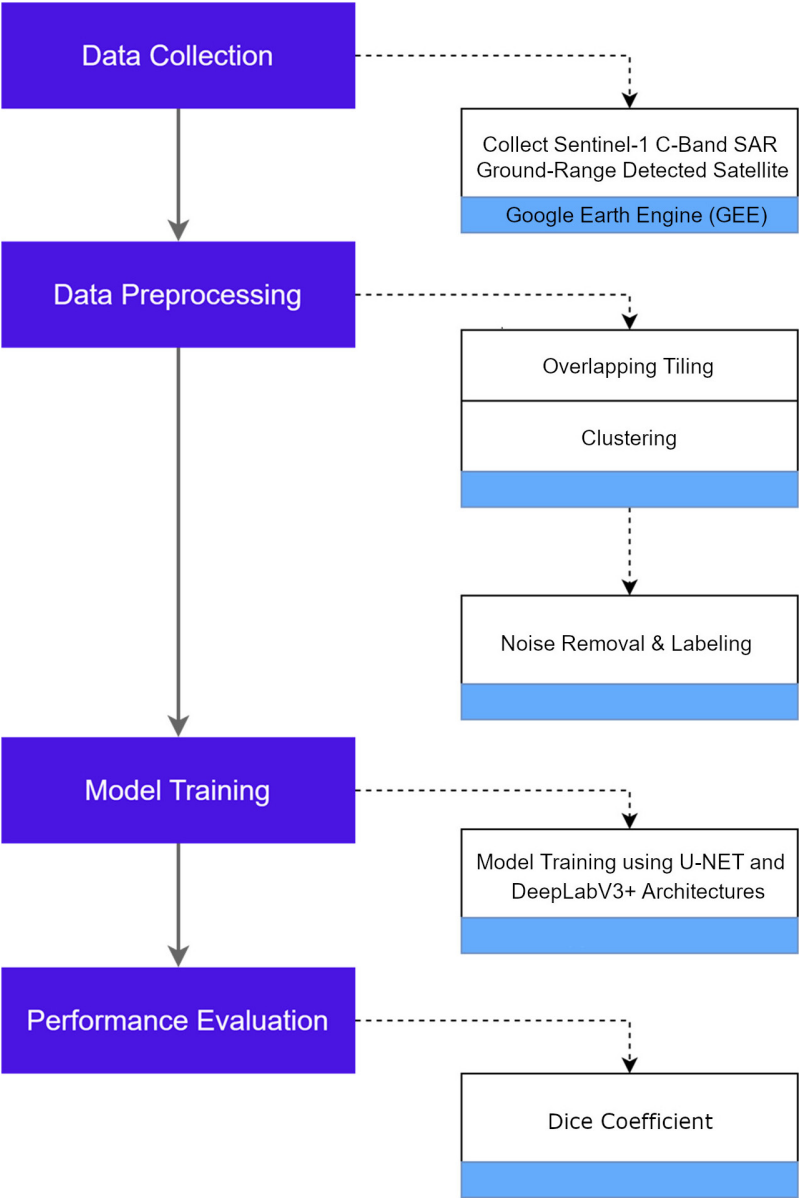


Fig. 1. Research methodology workflow

### 3.1. Data Collection

This research utilized SAR data from the Sentinel-1 Level 1 C-Band SAR GRD IW mode that was acquired from the Google Earth Engine (GEE) platform. The collected data was partitioned into training, validation, and test sets from the outset to ensure that the test data remained entirely unseen by the models during their development and training phases, thereby enabling an unbiased assessment of the generalizability.

Sentinel-1 GRD products were chosen because they are Level-1 processed; this means that the data is detected, multi-looked, and georeferenced using the WGS84 Earth ellipsoid (which reduces noise and speckle). SAR atellites use vertical (V) and horizontal (H) polarization to transmit and receive signals. The Sentinel-1 GRD IW data provides single- and dual-polarization options. Single polarization includes VV (vertical transmit and receive) or HH (horizontal transmit and receive), while dual polarization includes VV+VH (vertical transmit, vertical and horizontal receive) or HH+HV (horizontal transmit, horizontal and vertical receive). This study focuses on VH polarization, as it correlates more strongly with reference water masks than VV polarization and is more effective for detecting surface water [33].

For this study, a total of 30 Sentinel-1 Level-1 C-Band SAR GRD satellite images were extracted using GEE, with TIFF file sizes varying depending on the river coverage (from a minimum of  $792 \times 1308$  pixels to a maximum of  $10,458 \times 9556$  pixels). The extracted bands included VV, VH, and a calculation of the VV+VH polarization (VV+VH polarization was chosen because it can map water surfaces more clearly). The polygon creation for mapping the area and dimensions of the region of interest was part of the data-extraction process. The differing widths and heights of the pixels were due to limitations in GEE's precise polygon definition; these necessitated manual approximations. The selected areas covered locations along the Mekong River, capturing diverse river segments (including its tributary network).

### 3.2. Data Preprocessing

The data preprocessing encompassed: (1) overlapping and tiling, (2) clustering, and (3) noise removal and labeling.

#### Overlapping and Tiling

The extracted raw satellite data was tiled to cut the data into smaller resolutions; this aimed to increase the data set and provide different highlights for each bit of the data. An overlap tiling process was applied to the tiling process from the 30 images that had been extracted, resulting in  $2048 \times 2048$ -pixel tiles with an overlap of half the desired resolution size in order to increase the data set, provide different highlights, and minimize data loss. This process yielded 181 images after excluding those tiles that contained only river or non-river sections in order to avoid bias.



### Clustering: Pre-Labeling Process

Each tile was processed using the  $k$ -means clustering algorithm ( $k = 4$ ) to streamline the pre-labeling of the river and non-river areas, thus minimizing manual effort and highlighting the major rivers. The clustering output was a binary image where the river areas were preserved as the foreground. One example of the clustering process is shown in Figure 2; the first-row images stored the river area information in only one cluster, and the second-row images showed that the river information was stored in the two different clusters that needed to be combined/merged.

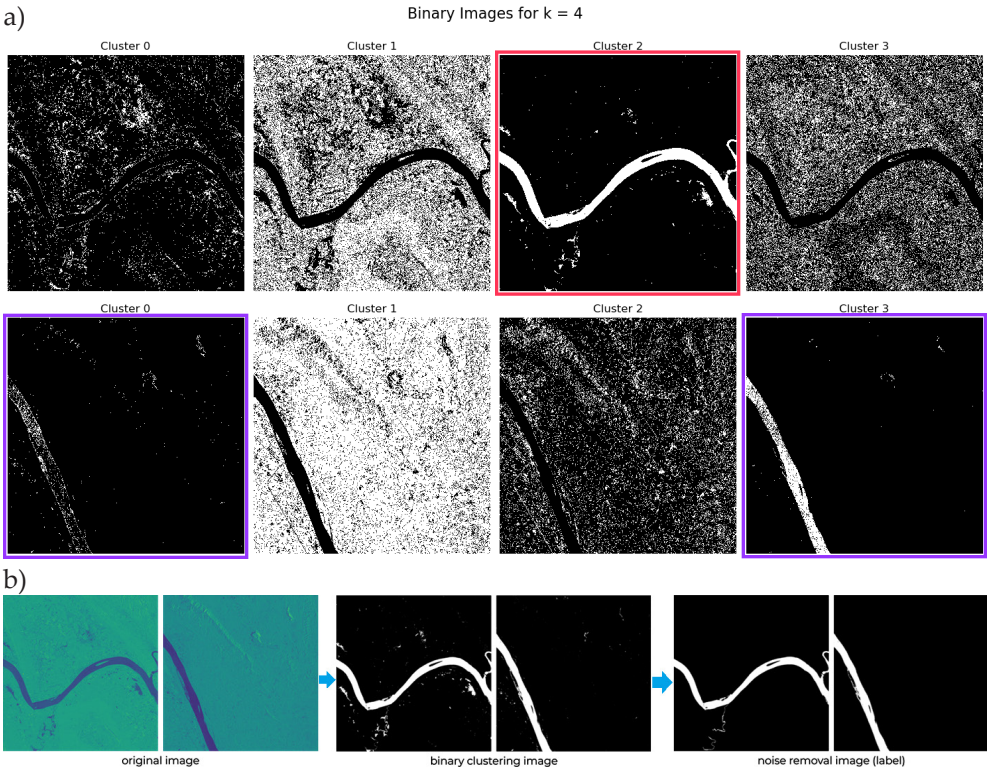


Fig. 2. Representative images from  $k$ -means clustering process (a); step-by-step process for mask data creation (b)

### Noise Removal and Labeling

Despite applying clustering, non-river areas with similar characteristics (e.g., ponds, lakes) were still clustered as river areas. These “noise” elements were manually removed using image-processing applications by reclassifying non-river foreground pixels as background. The step-by-step processes in creating the label data are shown in the third-row images of Figure 2. Manual labeling was also performed for unclear or faint river features, and those smaller rivers that were missed

by the clustering process were manually added by overlaying the clustering results on the original data. If multiple clusters contained river data, they were merged into a single binary image (white for rivers, and black for non-rivers). The binary clustering data was then converted from TIFF to the PNG format.

**Data Augmentation**

After the extracted data underwent preprocessing, data augmentation was applied in order to increase the quantity and variety of the data set. This step was crucial, as any changes in the river areas were typically insignificant over short periods. Therefore, artificial augmentation was used to enhance the data for modeling purposes. Data augmentation was applied to the training data through horizontal flipping, vertical flipping, zoom out, zoom in, and 45-degree-counterclockwise rotation. After augmentation, the final data set was comprised of 1080 images, including the original images and their corresponding masks. The data set was carefully divided into training, validation, and test sets, with the test set being kept completely separate and unused during the model’s development. This separation helped evaluate how well the model performed on new data, including different river shapes and environmental conditions that were not seen during the training. The train-to-test ratio was set at 9:1, with 90% of the data (972 entries) being used for the training. The remaining 10% (108 entries) was split evenly into 50% for the validation and 50% for the testing.

**3.3. Model Training**

The 972 training samples were used to train two deep-learning architectures: CNN U-Net, and DeepLabv3+. Both models received original and labeled data as inputs in the PNG format. Due to hardware limitations, the batch size was set to 8. After hyperparameter tuning, a learning rate of 1e-4 was chosen. The training was conducted for 150 epochs; however, optimal results were typically reached around the 100<sup>th</sup> epoch, thus leading to earlier stopping via the EarlyStopping callback. The Adam optimizer was used for the model optimization. Both architectures utilized four callbacks: ModelCheckpoint, ReduceLROnPlateau, CSVLogger, and EarlyStopping (as shown in Table 1).

**Table 1.** Hyperparameter list that was used in this work

Hyperparameters	Values
Batch size	8
Learning rate	1e-4
Epochs	150
Callbacks	ModelCheckpoint, ReduceLROnPlateau, CSVLogger, EarlyStopping
Activation function (CNN U-Net)	ReLU
Activation function (DeepLabv3+)	ReLU
Optimizer	Adam
Backbone network (DeepLabv3+)	ResNet50



ModelCheckpoint saved the best-performing model based on validation metrics, ReduceLROnPlateau lowered the learning rate if the validation loss stagnated, CSVLogger recorded the training and validation metrics, and EarlyStopping prevented overfitting by halting the training when the validation loss ceased to improve.

### CNN U-Net Architecture for Semantic Segmentation

The CNN U-Net architecture is a popular semantic segmentation architecture that was initially proposed for biomedical image segmentation by Ronneberger et al. [34]. As is shown in Figure 3, CNN U-Net consists of three paths: a contracting path (encoder), a bridge layer, and an expansive path (decoder). The contracting path uses repeated  $3 \times 3$  convolutions with rectified linear unit (ReLU) activation and  $2 \times 2$  max pooling for downsampling, thus doubling the feature channels at each step. The bridge layer connects the paths, while the expansive path involves up-sampling,  $2 \times 2$  up-convolution, concatenation with cropped feature maps from the contracting path, and two  $3 \times 3$  convolutions with ReLU.

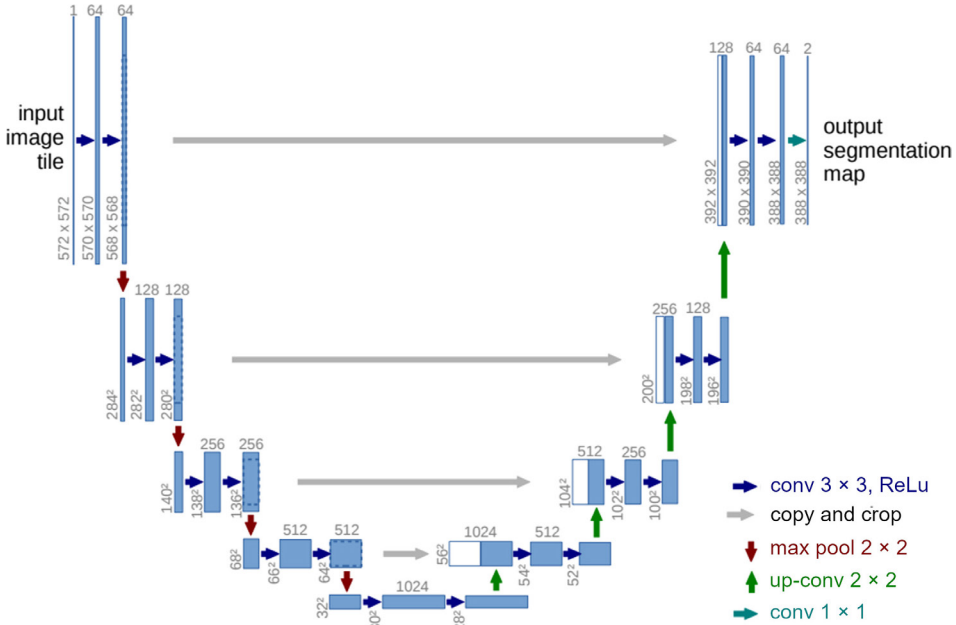


Fig. 3. CNN U-Net architecture (adapted from [34])

Cropping compensates for any border pixel loss during convolution. The final layer uses a  $1 \times 1$  convolution for class mapping. The network has 23 convolutional layers with copy and crop operations for enhancing localization and retaining high-level semantic information [2, 34]. This study used the original CNN U-Net with added BatchNormalization layers. The images were input as three-channel tensor data, and the labels were input as single-channel data; both were normalized to  $[0, 1]$ .

### DeepLabv3+ Architecture for Semantic Segmentation

The DeepLabv3+ model is an advanced version of a typical fully convolutional network (FCN) that excels in semantic segmentation by leveraging contextual information [3]. As the latest iteration in the DeepLab series, the DeepLabv3+ architecture incorporates an ASPP module based on spatial pyramid pooling (SPP) from DeepLabv3. The model uses parallel atrous convolutions at various rates to capture contextual features at multiple scales [35].

Additionally, it employs an encoder-decoder structure. The encoder captures rich contextual features using atrous convolutions and ASPP, while the decoder refines predictions at higher resolutions, thus improving the boundary accuracy and detail [3]. The decoder up-samples coarse feature maps and merges them with the higher-resolution features from the earlier layers. DeepLabv3+ offers advantages like improved boundary refinement and the better handling of small objects. This research used DeepLabv3+ with ResNet50 as a backbone, specifically utilizing the output of the conv4\_block\_6\_out layer for its high-level semantic information (shown in Figure 4). Similar to CNN U-Net, the images were three-channel tensor data, and the labels were single-channel; both were normalized to [0, 1].

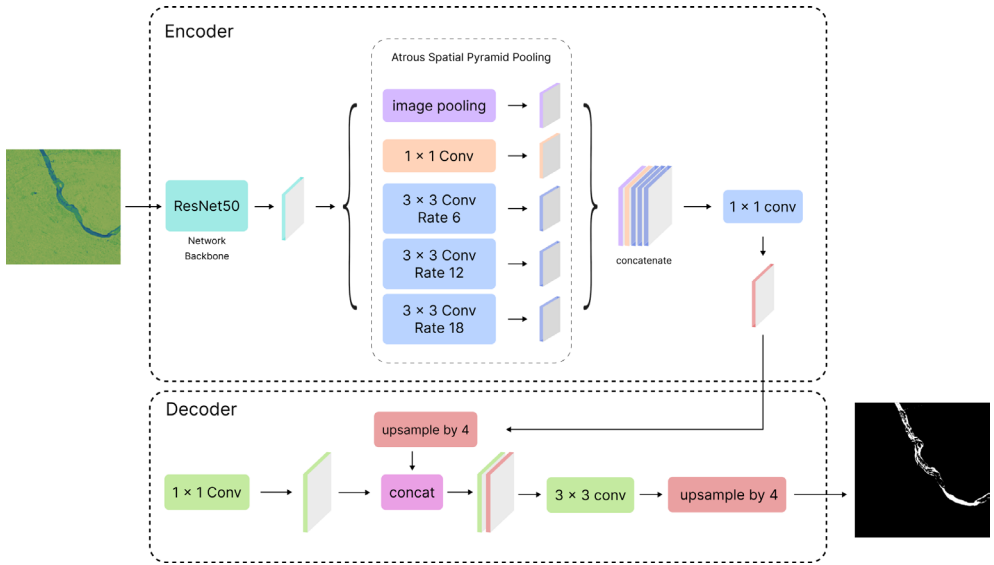


Fig. 4. Proposed DeepLabv3+ architecture using ResNet-50 as network backbone

### 3.4. Performance Evaluation

The data set that was trained with the U-Net and DeepLabv3+ architectures was evaluated using several metrics to assess the performance; these included the Dice coefficient, F1-score, Jaccard Index, recall, and precision. The Dice coefficient is

a primary metric that is used to assess the similarity between two sets ( $Y$  and  $Y$  prediction), with values that range from 0 to 1 [36]; a value of 1 indicates perfect similarity, while a value of 0 signifies no overlap between sets. The Dice coefficient is particularly important for evaluating image-segmentation performance, as it quantifies the spatial overlap between the ground truth and the predicted segmentation output [35].

The Dice coefficient is calculated as follows:

$$\text{Dice Coefficient} = 2 \frac{Y \cap Y_{pred}}{|Y| + |Y_{pred}|} \quad (1)$$

Conversely, the Dice loss is calculated as follows:

$$\text{Dice Loss} = 1 - \text{Dice Coefficient} \quad (2)$$

This metric is important for image segmentation, as it quantifies any spatial overlap between the ground truth and the predicted segmentation outputs. The Dice coefficient is particularly advantageous for imbalanced data sets and is highly sensitive to overlapping between predicted and ground-truth masks. Segmentation masks are treated as binary pixel sets. A high Dice coefficient indicates effective segmentation performance, while a low value reflects poor performance. The Dice loss was used to measure the error percentage (defined as the  $1 - \text{Dice coefficient}$ ). Along with the F1-score, Jaccard, recall, and precision, the Dice coefficient was used to assess the model's generalizability across diverse river characteristics in the unseen test set.

In addition to the Dice coefficient, the models' performance was also evaluated using the F1-score, Jaccard index, recall, and precision. The F1-score provides a balance between precision and recall, thus offering a single score that represents the harmonic mean of the two. The Jaccard index measures the similarity and diversity of the sample sets; these are calculated as the area of the intersection divided by the area of the union of the predicted and ground-truth-segmentation masks. While recall measures the proportion of the actual positive pixels that are correctly identified by the model, precision measures the proportion of correctly identified positive pixels out of all of the pixels that the model predicted as being positive.

## 4. Results and Discussion

### 4.1. Comparison Analysis

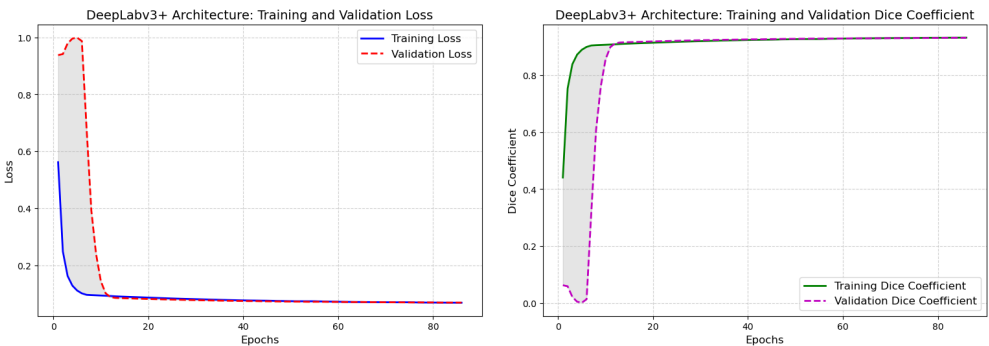
The analyses of selected test images provided insight into how the CNN U-Net and DeepLabv3+ models performed on various river shapes and environmental conditions from the unseen test set (complementing the scores in Table 2). The differing performance trajectories and subsequent test results were largely attributed to the fundamental architectural designs of CNN U-Net and DeepLabv3+. CNN U-Net excels at capturing fine details and precise localizations due to its skip connections, while DeepLabv3+ is designed for robust contextual understandings through atrous convolutions.

**Table 2.** Dice coefficient matrix of CNN U-Net and DeepLabv3+ architecture

Architecture	Dice coefficient	Dice loss	Validation Dice coefficient	Validation Dice loss	F1-score	Jaccard	Recall	Precision
CNN U-Net	0.94	0.05	0.93	0.06	0.91	0.87	0.91	0.93
DeepLabv3+	0.92	0.07	0.92	0.07	0.83	0.78	0.83	0.87

The DeepLabv3+ architecture is well-known for its ability to effectively map global information; it leverages atrous convolutions at multiple dilation rates within its ASPP module to effectively capture features across different spatial scales. This multi-scale context modeling is advantageous for interpreting larger structures and overall scene layouts. While it excels at capturing broad semantic information, however, it may be less precise in delineating fine boundaries or narrow features – particularly when compared to CNN U-Net’s direct feature concatenation through skip connections.

The DeepLabv3+ architecture’s training graph demonstrated a smooth progression with no significant drops in accuracy or sudden spikes in loss throughout the training process; this is depicted in Figure 5.



**Fig. 5.** Training history of DeepLabv3+ architecture

The DeepLabv3+ architecture is known for its effective global-information mapping; it leverages atrous convolutions at multiple dilation rates within its ASPP module to effectively capture features across different spatial scales. This multi-scale context modeling is advantageous for interpreting larger structures and overall scene layouts. While it excels at capturing broad semantic information, however, it may be less precise in delineating fine boundaries or narrow features when compared to CNN U-Net’s direct feature concatenation through skip connections. During the testing, the DeepLabv3+ model achieved a Dice coefficient accuracy of 0.928 and a Dice loss of 0.071. Training concluded at the 86<sup>th</sup> epoch out of 100. The validation

metrics mirrored the training results, with a validation Dice coefficient of 0.928 and a validation loss of 0.071.

In contrast, CNN U-Net’s accuracy gradually increased over time as the number of epochs increased; this is shown in Figure 6. The CNN U-Net architecture is renowned for its ability to perform effective segmentation – even with limited data sets. Its unique skip connections directly link encoder and decoder layers at their corresponding levels. These connections are crucial for retaining high-resolution features and recovering spatial information that may be lost during down-sampling, thus enabling highly accurate segmentation maps and the capturing of intricate details. This architectural advantage is particularly beneficial for identifying narrow structures like rivers. The CNN U-Net model was trained for 100 epochs with a batch size of 8; it utilized an early stopping callback. The model achieved a Dice coefficient accuracy of 0.944 and a Dice loss of 0.053, with the training halting at the 64<sup>th</sup> epoch. The validation Dice coefficient for the CNN U-Net architecture reached 0.939, with a validation loss of 0.071. The performance-evaluation metrics for both architectures are summarized in Table 2.

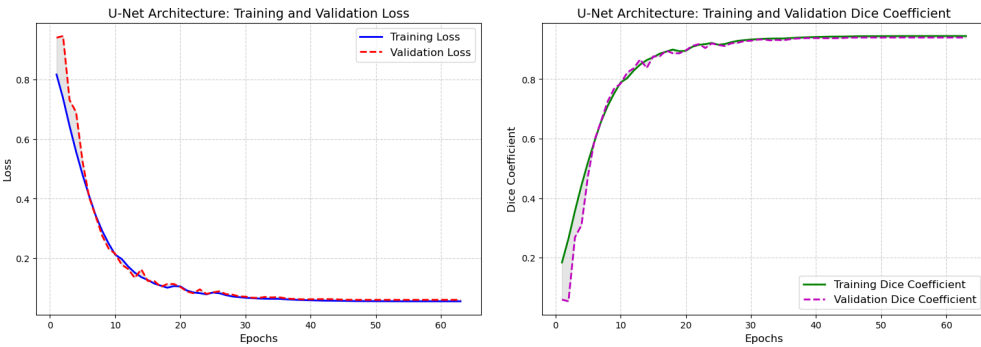
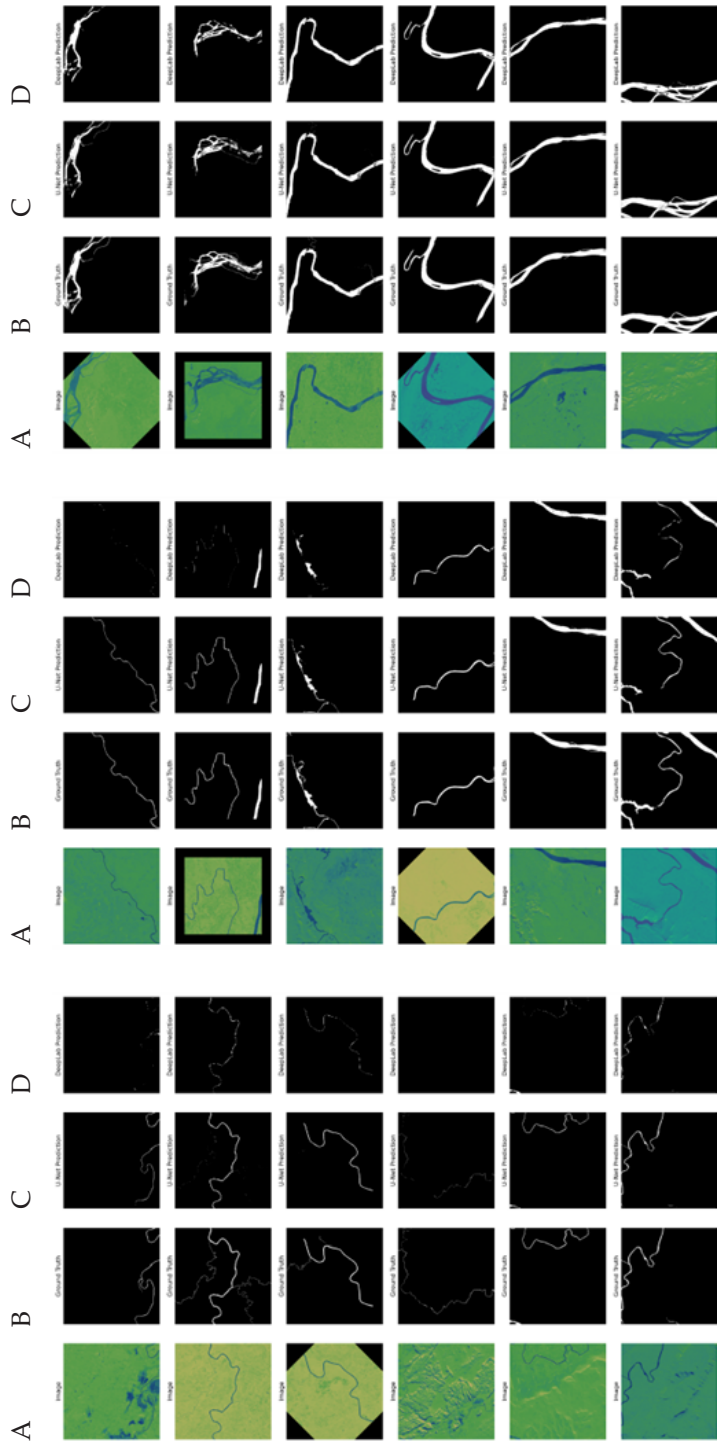


Fig. 6. Training history of CNN U-Net architecture

CNN U-Net’s architectural design (particularly, its skip connections) allows it to effectively preserve spatial information and capture the intricate details of narrow features, thus leading to its superior performance in identifying such elements. This ability to capture fine details and downscale models effectively is a known advantage in deep-learning applications (as evidenced by studies on enhanced urban-flood modeling using similar principles) [37].

CNN U-Net also performed better in predicting larger rivers when compared to DeepLabv3+, as it effectively recognized the characteristics of the rivers and labeled them accurately to reflect their actual formations. To illustrate these insights into its generalizability, specific case studies from the unseen test set are examined below; these provide qualitative and quantitative breakdowns of the model’s behavior across different river complexities. A visualization of the testing results using both models is shown in Figure 7.



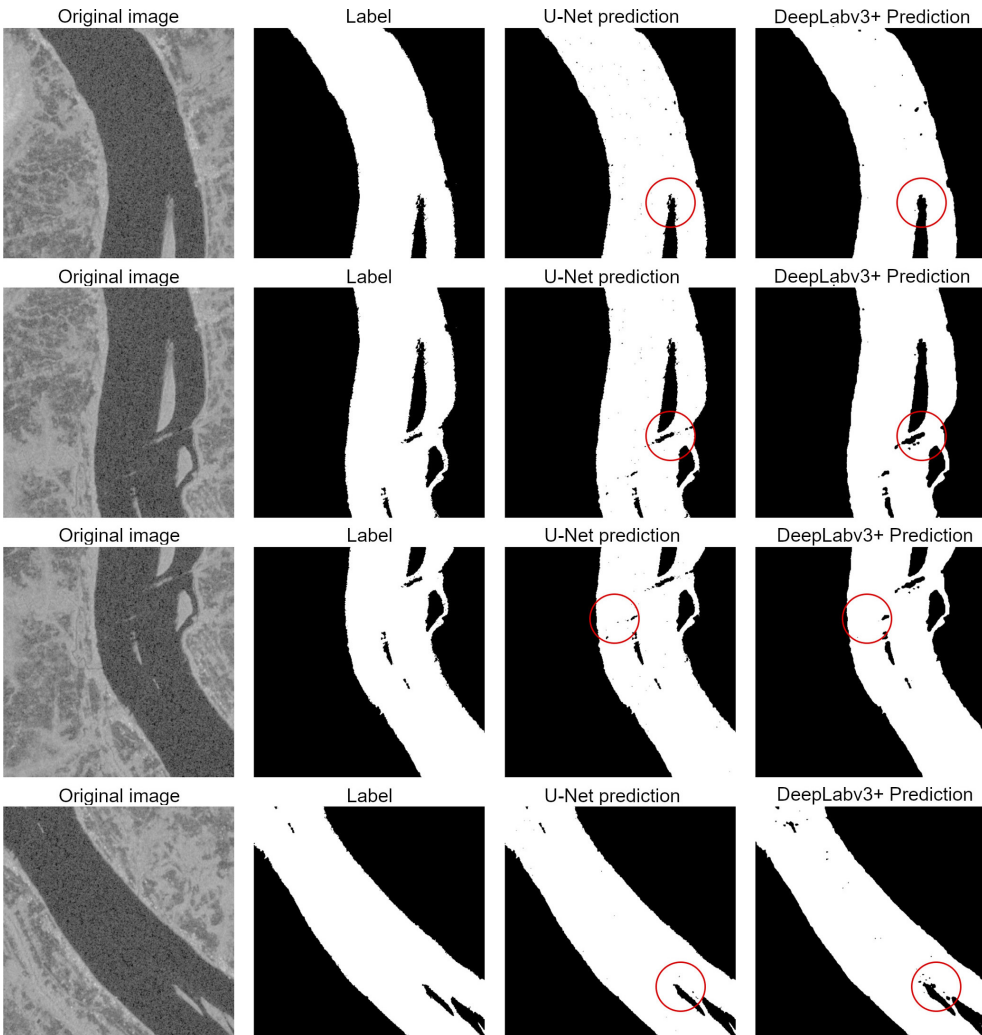
**Fig. 7.** Testing results of both CNN U-Net and DeepLabv3+ models:  
A – original images, B – ground truth/label image, C – CNN U-Net prediction, D – DeepLabv3+ prediction



## 4.2. Generalizability Insights from Diverse Test Cases

### Case Study: Very Large River Areas

This case study examined the models' generalizabilities when segmenting very large river areas (as visualized in Figure 8).



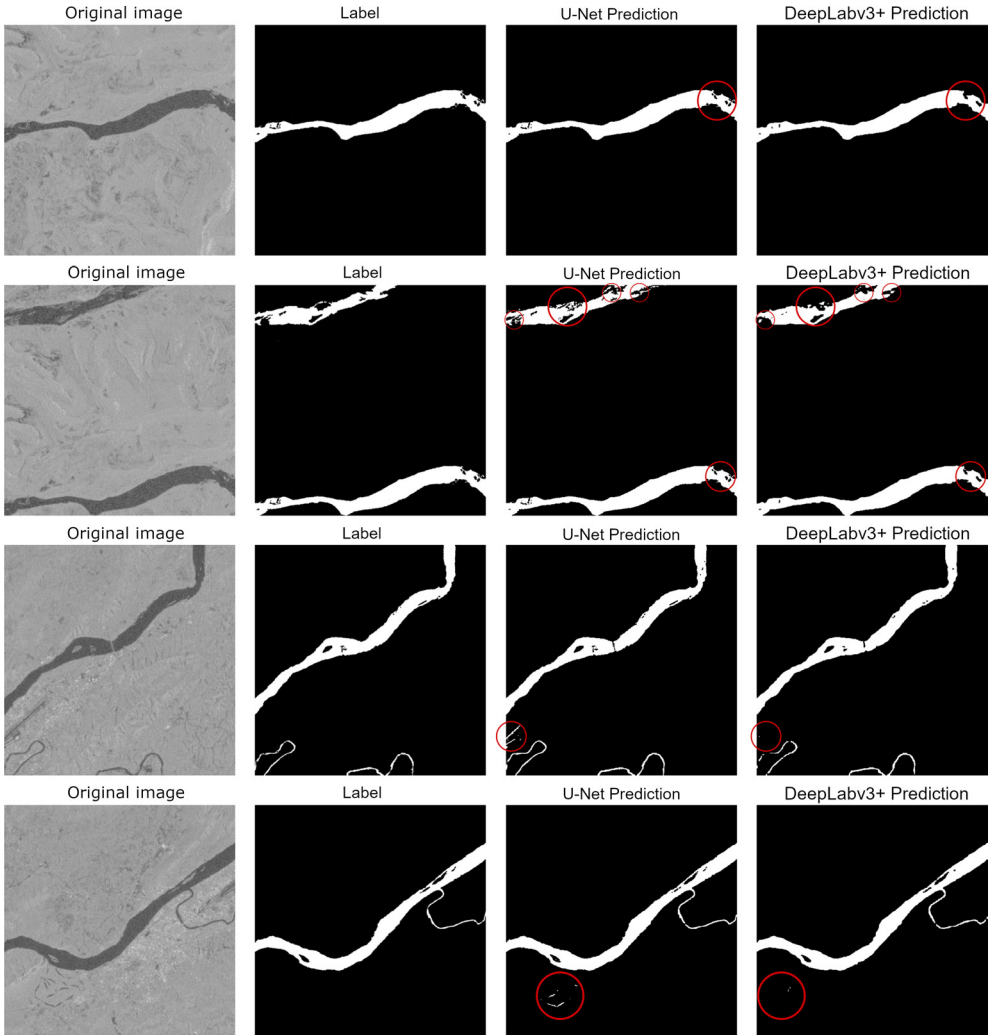
**Fig. 8.** Performance visualizations of CNN U-Net and DeepLabv3+ on very large river areas

In those images with very large river areas, both CNN U-Net and DeepLabv3+ effectively segmented river and non-river boundaries (closely matching the labeled images); this is visualized in Figure 8. The average recall scores were high, with CNN U-Net at around 0.99 and DeepLabv3+ between 0.97 and 0.99. CNN U-Net

captured fine details that were similar to the labeled images but introduced more noise (such as black or non-river pixels) within the river areas; these were absent in the labeled images. While DeepLabv3+ exhibited some noise in its predictions, this was less pronounced as compared to CNN U-Net. DeepLabv3+ also produced smoother and cleaner predictions, generally capturing small details with a more generalized approach, resulting in clearer and more-defined shapes.

**Case Study: Medium-Scale River Areas**

This case study evaluated the models’ generalizabilities with medium-scale river areas (illustrated in Figure 9).

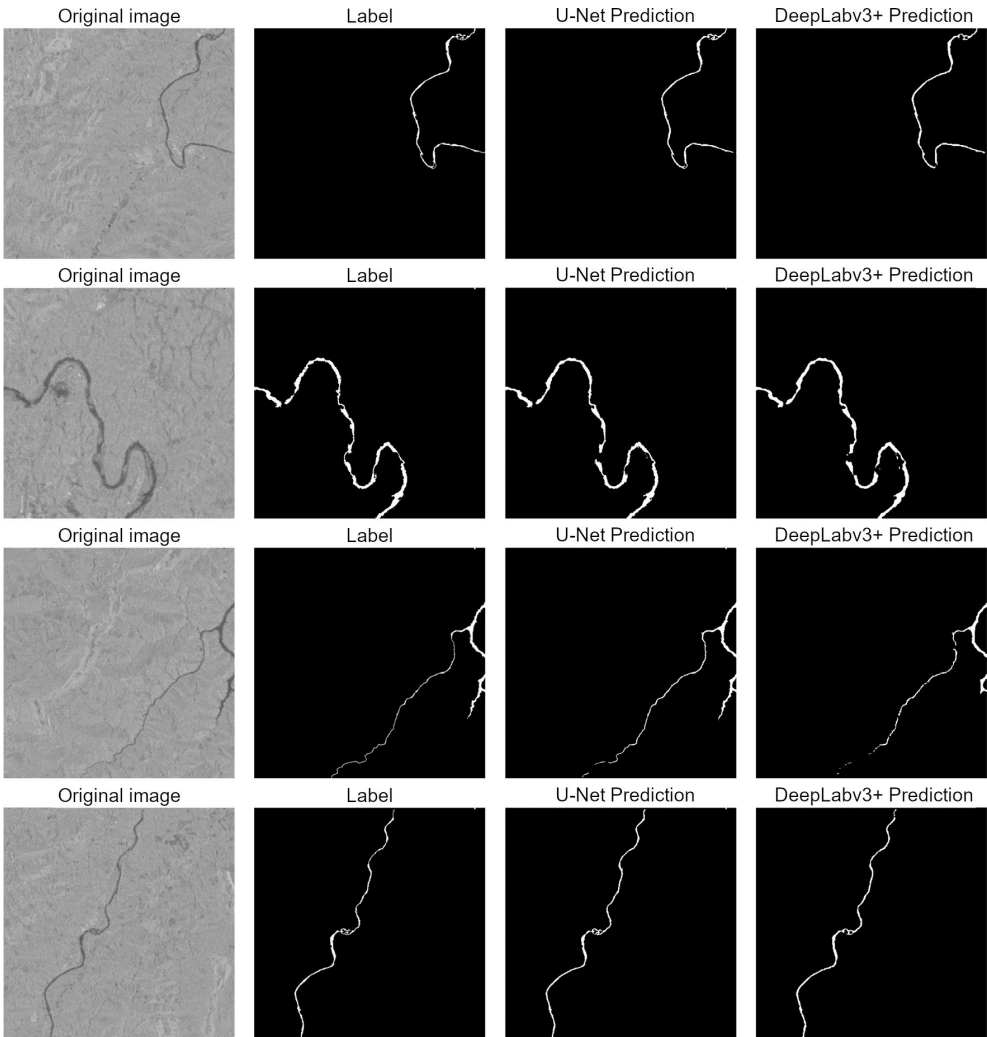


**Fig. 9.** Performance visualizations of CNN U-Net and DeepLabv3+ on medium-scale rivers

When evaluating medium-scale river areas (Fig. 9), CNN U-Net tended to map these areas as rivers but included some non-river regions (black pixels) with shallow waters or sediment. The DeepLabv3+ model, however, produced more-solid predictions by avoiding the misclassification of low-water areas as rivers. DeepLabv3+ demonstrated a better ability to exclude these areas from its river predictions, whereas CNN U-Net tended to include them.

**Case Study: Small River Areas**

This case study assessed the models' generalizabilities to very small river areas (depicted in Figure 10).

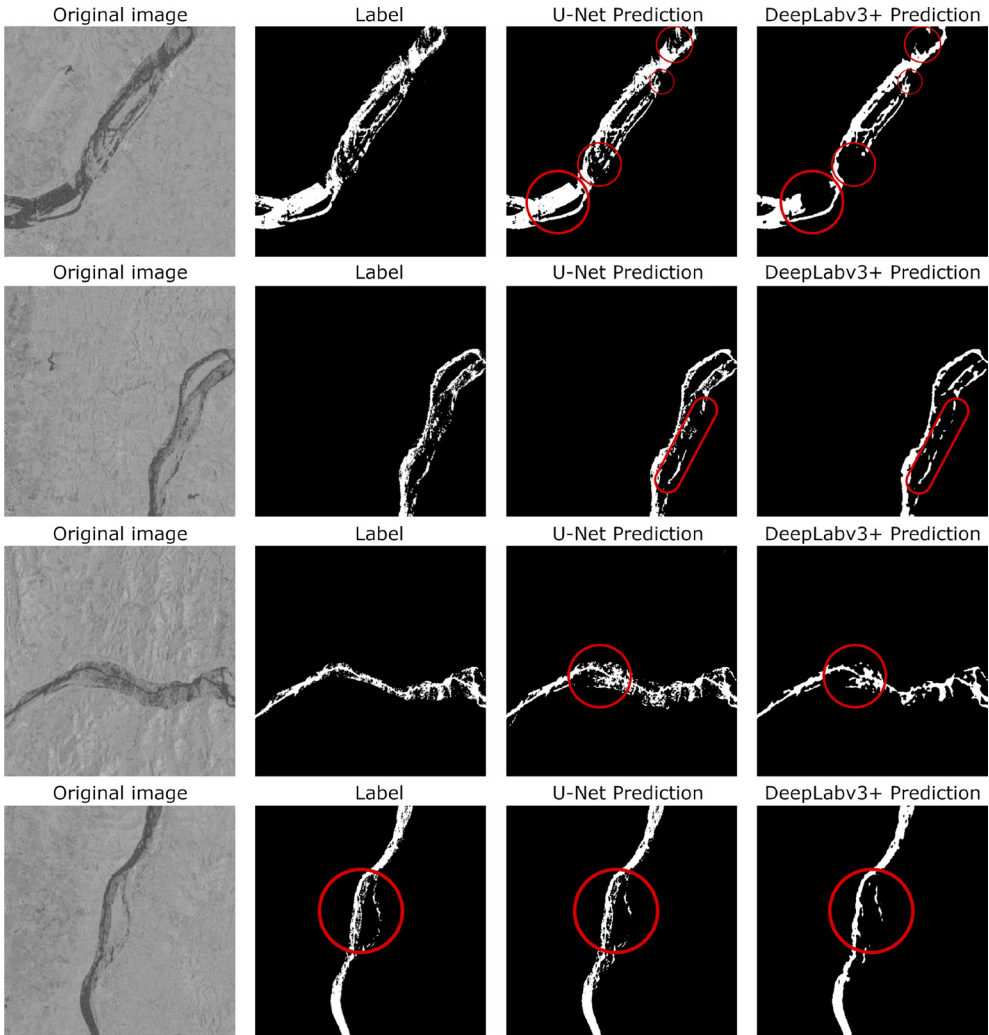


**Fig. 10.** Performance visualizations of CNN U-Net and DeepLabv3+ on small-scale rivers

**Case Study: Complex River Areas  
(Braided and Branched Rivers)**

This section evaluates the models’ generalizabilities to complex river areas – specifically, braided and branched river systems.

In the testing data, a river with a highly complex area was included; it featured a main river with many narrow tributaries and surrounding sediments or sands (as observed in Google Satellite and Sentinel-1 satellite imagery). The testing results for these complex braided river images differed between the CNN U-Net and DeepLabv3+ architectures; this is visualized in Figure 11.

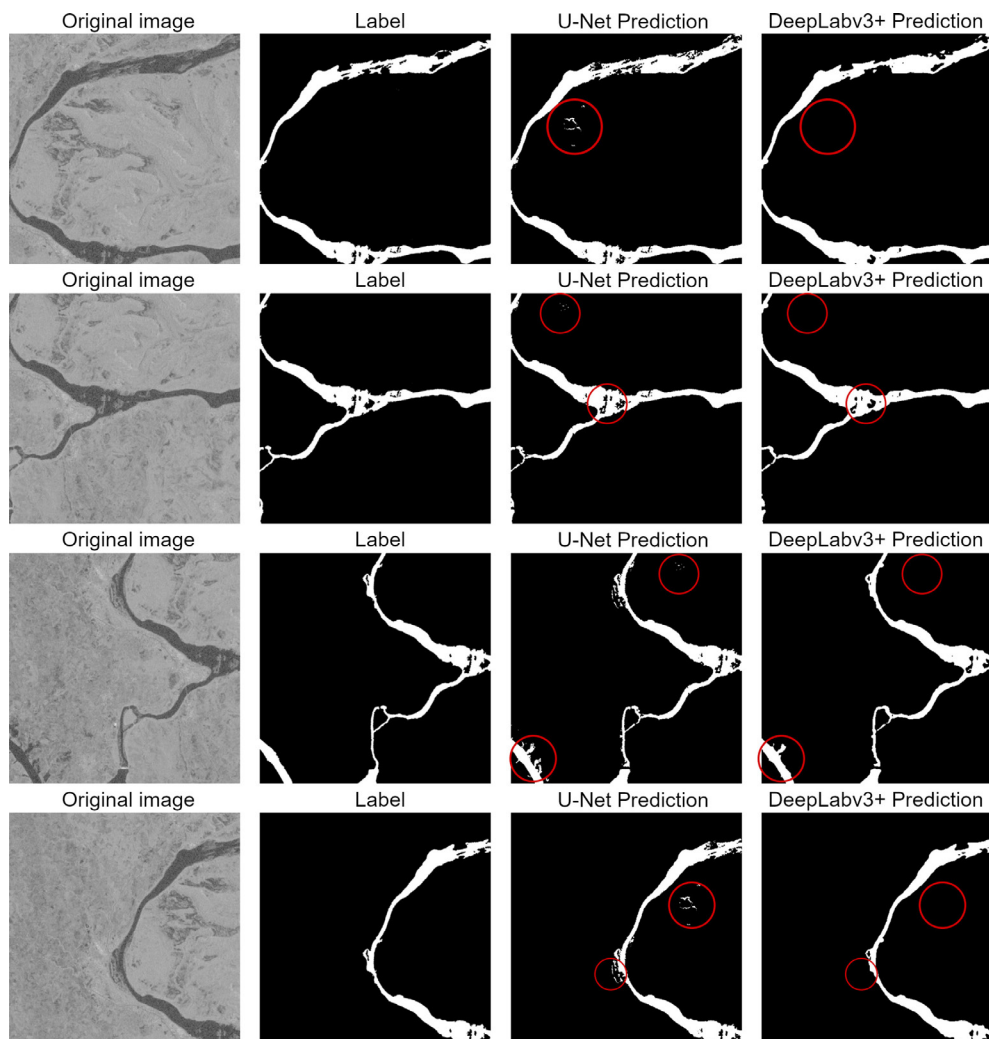


**Fig. 11.** Performance visualizations of CNN U-Net and DeepLabv3+ on braided rivers

CNN U-Net tended to predict the finer branches of the river – even when the grayscale intensity was not very dark in the original image. In contrast, DeepLabv3+ predicted the darker areas; these could be defined as parts of the main river. Based on these testing results, CNN U-Net's predictions were better as compared to DeepLabv3+'s. For this type of complex data, CNN U-Net's performance was significantly superior to DeepLabv3+'s, as its sensitivity enabled it to predict even highly complex river patterns. Because DeepLabv3+ tends to focus on global context, it conversely struggled to predict the finer branches or narrower tributaries of the river. Despite DeepLabv3+'s drawback of being less effective in detecting small/narrow rivers alongside larger rivers, it was still capable of mapping the main rivers and their tributaries (albeit, not as accurately as CNN U-Net). CNN U-Net was highly precise in predicting water segments that remained connected to the main rivers, making it easier to predict the details of smaller and narrower rivers.

The testing data also included images of rivers with unusual shapes and that featured unique branches and curves. When comparing how well the predicted river shapes matched the labeled data, DeepLabv3+ tended to perform better. Its predictions focused only on the river areas without including other objects that were not parts of the rivers. On the other hand, CNN U-Net sometimes misclassified non-river water bodies as rivers; that being said, CNN U-Net performed better in accurately capturing non-river areas within the rivers (black gaps or holes in the river areas). These gaps aligned more closely with the labeled data when compared to DeepLabv3+. The performance discrepancy between the CNN U-Net and DeepLabv3+ architectures on complex branched river images is shown in Figure 12.

Both CNN U-Net and DeepLabv3+ were equally capable of predicting very small rivers effectively (as depicted in Figure 10). In some instances, a discontinuity might have appeared in the lower part of a river (where a few pixels were predicted as being non-river); this was considered to be reasonable given that the river areas at the very bottoms consisted of only 1–2 pixels. Aside from these minor instances, the images with small river areas generally appeared very well-segmented with no major discontinuities. Both architectures successfully completed the training and yielded satisfactory results where they could distinguish river areas from similar features like ponds and lakes. Overall, the architectural differences between CNN U-Net's precise localization via skip connections and DeepLabv3+'s global contextual understanding through atrous convolutions directly explained their observed strengths and weaknesses in river segmentation. The model that was trained using the CNN U-Net architecture provided more-detailed predictions when compared to the DeepLabv3+ model. CNN U-Net was better at detecting small rivers in the images (though not as clearly as the ground truth; this was due to some river pixels being misclassified as being non-river). The results showed that CNN U-Net excelled in segmenting fine details (including narrow river segments) but demonstrated limitations in distinguishing non-river water areas from the rivers. The DeepLabv3+ model was less accurate in predicting small rivers, but it performed just as well as CNN U-Net for larger rivers.



**Fig. 12.** Performance visualizations of CNN U-Net and DeepLabv3+ on branched rivers

## 5. Conclusion

This research focused on developing a semantic segmentation model for rivers using SAR satellite data from Sentinel-1 C-Band Ground-Range Detected IW; this aimed to provide detailed river mapping for applications such as irrigation and flood monitoring – particularly in cloud-prone tropical regions.

Two deep-learning architectures (CNN U-Net and DeepLabv3+) were chosen for model training, which was performed on a data set that was augmented



to 1024 images. The testing results showed that CNN U-Net outperformed DeepLabv3+ in classifying rivers, with a Dice coefficient accuracy of 0.94 as compared to 0.92 for DeepLabv3+.

A detailed analysis revealed their distinct strengths and weaknesses. CNN U-Net excelled in segmenting fine details (including narrow river segments) and provided more-detailed predictions. Conversely, DeepLabv3+ demonstrated greater stability during the training and performed better on wider rivers (offering smoother and cleaner segmentations). While CNN U-Net sometimes struggled with distinguishing non-river water bodies from rivers, DeepLabv3+ showed better generalizations in such cases by avoiding over-segmentation. Conversely, DeepLabv3+ occasionally missed some small river details due to its more global segmentation approach.

Case studies from the unseen test set showed that the models could handle a range of river shapes and environmental conditions, thus supporting their abilities to generalize. Future work should focus on testing the models with entirely new data from different regions and time periods; this would offer a stronger assessment of how well the models can perform across global river systems and environmental conditions that were not covered in the current data set.

## **Funding**

The research was supported by a grant from the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia with Contract Number 397/UN48.16/LT/2024.

## **CRediT Author Contribution**

N. P. K. D.: conceptualization, investigation, methodology, software, visualization, writing – original draft, writing – review & editing.

P. H. S.: conceptualization, funding acquisition, supervision, writing – review & editing.

A. A. G. Y. P.: conceptualization, funding acquisition, supervision, writing – review & editing.

L. J. E. D.: conceptualization, funding acquisition, supervision, writing – review & editing.

P. V.: conceptualization, methodology, project administration, supervision, writing – review & editing.

K. Y. E. A.: conceptualization, funding acquisition, investigation, methodology, project administration, supervision, writing – original draft, writing – review & editing.

## **Declaration of Competing Interests**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work that was reported in this paper.

### Data Availability

The public data in this article was downloaded from Google Earth Engine (<https://earthengine.google.com/>). Requests for access to the data sets that were generated or analyzed in this research will be considered upon inquiries to the corresponding author.

### Use of Generative AI and AI-Assisted Technologies

No generative AI or AI-assisted technologies were employed in the preparation of this manuscript.

### Acknowledgements

The authors would like to thank the Data Science Research Group of Universitas Pendidikan Ganesha (Undiksha) Indonesia, KMUTT Geospatial Engineering and Innovation Center (KGEO) Thailand, and the Ganetics Student Research Group of Computer Science Undiksha for supporting this study.

### References

- [1] Fan Z., Hou J., Zang Q., Chen Y., Yan F.: *River segmentation of remote sensing images based on composite attention network*. Complexity, vol. 2022, 2022. <https://doi.org/10.1155/2022/7750281>.
- [2] Pai M.M.M., Mehrotra V., Aiyar S., Verma U., Pai R.M.: *Automatic segmentation of river and land in SAR images: a deep learning approach*, [in:] *Proceedings of the IEEE 2nd International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 2019, pp. pp. 15–20. <https://doi.org/10.1109/AIKE.2019.00011>.
- [3] Li Z., Wang R., Zhang W., Hu F., Meng L.: *Multiscale features supported DeepLabv3+ optimization scheme for accurate water semantic segmentation*. IEEE Access, vol. 7, 2019, pp. 155787–155804. <https://doi.org/10.1109/ACCESS.2019.2949635>.
- [4] Ismanto R.D., Fitriana H.L., Manalu J., Purboyo A.A., Prasasti I.: *Development of flood-hazard-mapping model using random forest and frequency ratio in Sumedang Regency, West Java, Indonesia*. Geomatics and Environmental Engineering, vol. 17(6), 2023, pp. 129–157. <https://doi.org/10.7494/geom.2023.17.6.129>.
- [5] Zhu H., Li C., Zhang L., Shen J.: *River channel extraction from SAR images by combining gray and morphological features*. Circuits, Systems and Signal Processing, vol. 34(7), 2015, pp. 2271–2286. <https://doi.org/10.1007/s00034-014-9922-2>.
- [6] Vignesh T., Thyagarajan K.K.: *Water bodies identification from multispectral images using Gabor filter, FCM and Canny edge detection methods*, [in:] *Proceedings of the 2017 International Conference on Information Communication and Embedded Systems (ICICES)*, IEEE, 2017, pp. 1–5. <https://doi.org/10.1109/ICICES.2017.8070767>.

- 
- [7] Liu Z., Li F., Li N., Wang R., Zhang H.: *A novel region-merging approach for coastline extraction from Sentinel-1A IW mode SAR imagery*. IEEE Geoscience and Remote Sensing Letters, vol. 13(3), 2016, pp. 324–328. <https://doi.org/10.1109/LGRS.2015.2510745>.
  - [8] Yang K., Li M., Liu Y., Cheng L., Huang Q., Chen Y.: *River detection in remotely sensed imagery using Gabor filtering and path opening*. Remote Sensing, vol. 7(7), 2015, pp. 8779–8802. <https://doi.org/10.3390/rs70708779>.
  - [9] Ciecholewski M.: *River channel segmentation in polarimetric SAR images: watershed transform combined with average contrast maximisation*. Expert Systems with Applications, vol. 82, 2017, pp. 196–215. <https://doi.org/10.1016/j.eswa.2017.04.018>.
  - [10] Ko B., Kim H., Nam J.: *Classification of potential water bodies using Landsat 8 OLI and a combination of two boosted random forest classifiers*. Sensors, vol. 15(6), 2015, pp. 13763–13777. <https://doi.org/10.3390/s150613763>.
  - [11] Goumehei E., Tolpekin V., Stein A., Yan W.: *Surface water body detection in polarimetric sar data using contextual complex Wishart classification*. Water Resources Research, vol. 55(8), 2019, pp. 7047–7059. <https://doi.org/10.1029/2019WR025192>.
  - [12] Wei Z., Jia K., Liu P., Jia X., Xie Y., Jiang Z.: *Large-scale river mapping using contrastive learning and multi-source satellite imagery*. Remote Sensing, vol. 13(15), 2021, 2893. <https://doi.org/10.3390/rs13152893>.
  - [13] Neupane B., Horanont T., Aryal J.: *Deep learning-based semantic segmentation of urban features in satellite images: A review and meta-analysis*. Remote Sensing, vol. 13(4), 2021, 808. <https://doi.org/10.3390/rs13040808>.
  - [14] Tian X., de Bruin S., Simoes R., Isik M. S., Minarik R., Ho Y.-F., Şahin M., Herold M., Consoli D., Hengl T.: *Spatiotemporal prediction of soil organic carbon density for Europe (2000–2022) in 3D+T based on Landsat-based spectral indices time-series*. Research Square, 2024. <https://doi.org/10.21203/rs.3.rs-5128244/v1>.
  - [15] Singh G., Dahiya N., Sood V., Singh S., Sharma A.: *ENVINet5 deep learning change detection framework for the estimation of agriculture variations during 2012–2023 with Landsat series data*. Environmental Monitoring and Assessment, vol. 196(3), 2024, 233. <https://doi.org/10.1007/s10661-024-12394-8>.
  - [16] Ait El Asri S., Negabi I., El Adib S., Raissouni N.: *Enhancing building extraction from remote sensing images through UNet and transfer learning*. International Journal of Computers and Applications, vol. 45(5), 2023, pp. 413–419. <https://doi.org/10.1080/1206212X.2023.2219117>.
  - [17] Hao S., Zhou Y., Guo Y.: *A brief survey on semantic segmentation with deep learning*. Neurocomputing, vol. 406, 2020, pp. 302–321. <https://doi.org/10.1016/j.neucom.2019.11.118>.
  - [18] Yu H., Yang Z., Tan L., Wang Y., Sun W., Sun M., Tang Y.: *Methods and datasets on semantic segmentation: A review*. Neurocomputing, vol. 304, 2018, pp. 82–103. <https://doi.org/10.1016/j.neucom.2018.03.037>.

- [19] Wang P., Chen P., Yuan Y., Liu D., Huang Z., Hou X., Cottrell G.: *Understanding convolution for semantic segmentation*, [in:] *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2018, pp. 1451–1460. <https://doi.org/10.1109/WACV.2018.00163>.
- [20] Guo Y., Liu Y., Georgiou T., Lew M.S.: *A review of semantic segmentation using deep neural networks*. *International Journal of Multimedia Information Retrieval*, vol. 7(2), 2018, pp. 87–93. <https://doi.org/10.1007/s13735-017-0141-z>.
- [21] Khan S.D., Alarabi L., Basalamah S.: *Deep hybrid network for land cover semantic segmentation in high-spatial resolution satellite images*. *Information*, vol. 12(6), 2021, p. 230. <https://doi.org/10.3390/info12060230>.
- [22] Wurm M., Stark T., Zhu X.X., Weigand M., Taubenböck H.: *Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks*. *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, 2019, pp. 59–69. <https://doi.org/10.1016/j.isprsjprs.2019.02.006>.
- [23] Barthakur M., Sarma K.K.: *Semantic segmentation using K-means clustering and deep learning in satellite image*, [in:] *2019 2nd International Conference on Innovations in Electronics, Signal Processing and Communication (IESPC)*, IEEE, 2019, pp. 192–196. <https://doi.org/10.1109/IESPC.2019.8902391>.
- [24] Hordiiuk D., Oliinyk I., Hnatushenko V., Maksymov K.: *Semantic segmentation for ships detection from satellite imagery*, [in:] *2019 IEEE 39th International Conference on Electronics and Nanotechnology (ELNANO)*, IEEE, 2019, pp. 454–457. <https://doi.org/10.1109/ELNANO.2019.8783822>.
- [25] Askevold R., Vågen M.: *Automated mapping and change detection of rivers and inland water bodies by semantic segmentation of SAR imagery using deep learning*. NTNU, Trondheim 2022 [M.Sc. thesis]. <https://hdl.handle.net/11250/3020996>.
- [26] Zou Q., Yu J., Fang H., Qin J., Zhang J., Liu S.: *Group-based atrous convolution stereo matching network*. *Wireless Communications and Mobile Computing*, vol. 2021(1), 2021, 7386280. <https://doi.org/10.1155/2021/7386280>.
- [27] Carreño Conde F., De Mata Muñoz M.: *Flood monitoring based on the study of Sentinel-1 SAR images: The Ebro River case study*. *Water (Basel)*, vol. 11(12), 2019, 2454. <https://doi.org/10.3390/w11122454>.
- [28] Belba P., Kucaj S., Thanas J.: *Monitoring of water bodies and non-vegetated areas in Selenica – Albania with SAR and optical images*. *Geomatics and Environmental Engineering*, vol. 16(3), 2022, pp. 5–25. <https://doi.org/10.7494/geom.2022.16.3.5>.
- [29] Pappas O.A., Anantrasirichai N., Achim A.M., Adams B.A.: *River planform extraction from high-resolution SAR images via generalized gamma distribution superpixel classification*. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59(5), 2021, pp. 3942–3955. <https://doi.org/10.1109/TGRS.2020.3011209>.

- 
- [30] Verma U., Chauhan A., Pai M.M.M., Pai R.: *DeepRivWidth: Deep learning based semantic segmentation approach for river identification and width measurement in SAR images of Coastal Karnataka*. Computers & Geosciences, vol. 154, 2021, 104805. <https://doi.org/10.1016/j.cageo.2021.104805>.
- [31] Cai Q., Wan R., Li H., Wang C., Chang H.: *Remote sensing image river segmentation method based on U-Net*, [in:] 2022 IEEE 8th International Conference on Cloud Computing and Intelligent Systems (CCIS), IEEE, 2022, pp. 215–220. <https://doi.org/10.1109/CCIS57298.2022.10016397>.
- [32] Chen S., Wei X., Zheng W.: *ASA-DRNet: An improved DeepLabv3+ framework for SAR image segmentation*. Electronics (Switzerland), vol. 12(6), 2023, 1300. <https://doi.org/10.3390/electronics12061300>.
- [33] Pham-Duc B., Prigent C., Aires F.: *Surface water monitoring within Cambodia and the Vietnamese Mekong Delta over a year, with Sentinel-1 SAR observations*. Water (Basel), vol. 9(6), 2017, 366. <https://doi.org/10.3390/w9060366>.
- [34] Ronneberger O., Fischer P., Brox T.: *U-Net: convolutional networks for biomedical image segmentation*, [in:] Navab N., Hornegger J., Wells W., Frangi A. (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. MICCAI 2015, Lecture Notes in Computer Science, vol. 9351, Springer, Cham 2015, pp. 234–241. [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28).
- [35] Kolhar S., Jagtap J.: *Convolutional neural network based encoder-decoder architectures for semantic segmentation of plants*. Ecological Informatics, vol. 64, 2021, 101373. <https://doi.org/10.1016/j.ecoinf.2021.101373>.
- [36] Jadon S.: *A survey of loss functions for semantic segmentation*, [in:] 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2020), IEEE, 2020, pp. 1–7, <https://doi.org/10.1109/CIBCB48159.2020.9277638>.
- [37] Zandsalimi Z., Barbosa S.A., Alemazkooor N., Goodall J.L., Shafiee-Jood M.: *Deep learning-based downscaling of global digital elevation models for enhanced urban flood modeling*. Journal of Hydrology (Amsterdam), vol. 653, 2025, 132687. <https://doi.org/10.1016/j.jhydrol.2025.132687>.